2020

Reference Governors: From Theory to Practice

Joycer Osorio
University of Vermont

Reference Governors: From Theory to Practice

A Dissertation Presented

by

Joycer Osorio

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
Specializing in Electrical Engineering

August, 2020

Defense Date: July 23th, 2020
Dissertation Examination Committee:

Hamid Ossareh, Ph.D. Advisor
Taras Lakoba, Ph.D. Chairperson
Mads Almassalkhi, Ph.D.
Luis Duffaut Espinosa, Ph.D.
Mario Santillo, Ph.D. Ford Motor Company
Cynthia J. Forehand, Ph.D. Dean of Graduate College

Abstract

Control systems that are subject to constraints due to physical limitations, hardware protection, or safety considerations have led to challenging control problems that have piqued the interest of control practitioners and theoreticians for many decades. In general, the design of constraint management schemes must meet several stringent requirements, for example: low computational burden, performance, recovery mechanisms from infeasibility conditions, robustness, and formulation simplicity. These requirements have been particularly difficult to meet for the following three classes of systems: stochastic systems, linear systems driven by unmodeled disturbances, and nonlinear systems. Hence, in this work, we develop three constraint management schemes, based on Reference Governor (RG), for these classes of systems. The first scheme, which is referred to as Stochastic RG, leverages the ideas of chance constraints to construct a Stochastic Robustly Invariant Maximal Output Admissible set (SR-MAS) in order to enforce constraints on stochastic systems. The second scheme, which is called Recovery RG (RRG), addresses the problem of recovery from infeasibility conditions by implementing a disturbance observer to update the MAS, and hence recover from constraint violations due to unmodeled disturbances. The third method addresses the problem of constraint satisfaction on nonlinear systems by decomposing the design of the constraint management strategy into two parts: enforcement at steady-state, and during transient. The former is achieved by using the forward and inverse steady-state characterization of the nonlinear system. The latter is achieved by implementing an RG-based approach, which employs a novel Robust Output Admissible Set (ROAS) that is computed using data obtained from the nonlinear system. Added to this, this dissertation includes a detailed literature review of existing constraint management schemes to compare and highlight advantages and disadvantages between them. Finally, all this study is supported by a systematic analysis, as well as numerical and experimental validation of the closed-loop systems performance on vehicle roll-over avoidance, turbocharged engine control, and inverted pendulum control problems.

To my parents,
Silvia and Luis,
and to my brother and sister,
Luis and Urpy
with love.

Acknowledgements

First and foremost, I would like thank my advisor, Prof. Hamid Ossareh, for his guidance and tutoring during all my doctoral studies, for allowing me to get into the control research community, and for granting me the opportunity to do both theoretical and applied research. From a more personal level, I am deeply thankful to him for believing in me and spending the time every week for the past four years to allow me to learn from his critical thinking and wisdom.

I thank my committee for spending the time to guide me in this dissertation and providing me helpful feedback about my research. I especially thank Prof. Mads Almassalkhi and Prof. Luis Duffaut Espinosa for helping to put the first stepping-stone that allowed me to get into UVM. Also, Dr. Mario Santillo for working closely with me and providing mentorship during the development of the Ford-UVM project and the internships at Ford Motor Company.

I also acknowledge the members of the Advanced and Applied Optimal Systems and Control at Ford Research & Advanced Engineering, specially Dr. Julia Buckland, Dr. Mrdjan Jankovic, Mr. Chenliu Stephen Lu, Dr. Amey Karnik, Mr. John Polonchan, Dr. Ashley Wiese, and Dr. Baitao Xiao; Ford Motor Company for funding my research and supporting my work; my colleagues and good friends at UVM, specially Sarnaduti Brahma, Collin Freiheit, Adil Khurram, Aidan Laracy, Yudan Liu, Nawaf Nazir, and Molly Rose. I want to thank my family and friends for their support and love during all this journey, without them I would not be able to make this achievement possible.

Last but not least, I heartfully thank my girlfriend, Melissa Chima, for being next to me, for all her support and love, thanks.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In control system applications, it is a common practice to design controllers that handle tracking, disturbance rejection, and closed-loop stability. However, it happens often that constraint enforcement is not rigorously considered in the control design process. Nevertheless, as dynamic systems become downsized and performance is maximized to its physical boundaries, it is exceedingly important for the control systems to be cognizant of the constraints. Constrained control schemes are designed with the intention to preserve the integrity of hardware components and user's safety, while keeping the desirable closed-loop transient performance. To address the growing necessity of these schemes, control practitioners and theoreticians have explored techniques that use steady-state clips on control commands to enforce constraints, methods that address both tracking and constraint management, and schemes that leverage set theoretic approaches to modify the control reference under certain circumstances to enforce constraints. The latter is the case of the reference governor, which is an add-on mechanism that modifies the reference signal to closed-loop systems only if it predicts constraint violations, otherwise, the reference is not changed. This dissertation presents the work that has been developed around RG to extend its applicability and to overcome its shortcomings.

1.1 Motivation

The work developed in this dissertation was motivated by the necessity to study a novel model-based control strategy with real-time capabilities for constrained systems in automotive applications, specifically turbocharged engines. Nevertheless, the schemes and theory presented in this dissertation are not only applicable to these

applications.

Automotive systems are becoming increasingly complex due to ever-stringent fuel economy, emissions, and performance requirements. To meet these requirements, most systems are pushed to operate closer to their physical constraints, making control of these systems a challenging task. For instance, in aggressively downsized turbocharged engines, the engine is forced to operate close to the turbocharger hardware limits, such as the turbocharger speed limit and the compressor outlet temperature limit. If these limits are violated, the hardware integrity may be compromised. Conventional hardware-protection strategies impose limits on the desired boost pressure that are determined based on the steady-state relationships between pressure, temperature and turbocharger speed. Although effective, this static approach does not take into account system dynamics. Therefore, offsets often must be included in the static limits to avoid constraint violation during transients. These offsets reduce maximum achievable boost pressure and therefore engine torque, potentially impacting driveability and performance

The existing control literature offers various advanced state-of-the-art constraint-management approaches. One common approach is founded on Lyapunov-based barrier functions, wherein a barrier function is defined and the system is controlled away from this barrier. A disadvantage of this approach is that it may be difficult to analyze and synthesize. An alternative approach is Model Predictive Control (MPC), wherein a model of the system is used to compute an optimal-control law to steer the system away from the constraints. MPC is particularly attractive because it can perform both tracking and constraint management. However, in the automotive sector, it can be challenging to replace legacy closed-loop engine controllers unless the al-

ternatives simultaneously minimize controller design effort, computational overhead, and calibration effort. Also, MPC may not be amenable to real-time implementation due to large computational overhead.

Another constraint-management approach is the Reference Governor (RG), which uses a model of the system to modify the reference command to the closed-loop system when constraint violation is predicted. In contrast to barrier or model predictive approaches, RG is simple to develop and analyze and has a low computational complexity. Furthermore, RG is an add-on mechanism to existing control strategies and can be designed independently of the tracking controller. Despite the above attractive features, RG has a few shortcomings. First, many practical systems are affected by stochastic noise. In the context of RG, these disturbances have traditionally been treated by introducing conservative margins by considering the worst case disturbance realizations and hence leading to conservative solutions. Second, if the constraints are violated (for example, due to larger than expected or unmodeled disturbances), the RG algorithm may get "stuck", i.e., permanently stay in infeasibility condition. Currently, there is no theoretical development that would steer the system back from constraint violation. Third, the RG cannot efficiently and effectively handle system nonlinearities, which are inherent to almost all real systems.

The reference governor is the main focus of this dissertation, and the contributions, presented here, address the problems mentioned above. By contributing to the RG literature, this dissertation provides practical tools and a theoretical framework for constraint management of a broad class of dynamical systems.

1.2 Problem Statement

For the sake of clarity, the problem statement for each one of the constraint management schemes presented in this work is explained separately.

1.2.1 Constraint Enforcement in Stochastic Linear Systems

Predictive control and constraint management have become important topics in the past few decades. One of the commonly used methods, which has shown an increasing acceptance by the industry, is Model Predictive Control (MPC) [5]. In recent years, in addition to deterministic approaches, several probabilistic approaches to MPC have also been developed. Examples are stochastic MPC for controlling average number of constraint violations [6], probabilistic MPC [7], and scenario generation MPC [8]. However, these probabilistic approaches may be numerically expensive and not amenable to real-time implementation for systems with fast dynamics controlled by slow processors. Furthermore, properties like stability and recursive feasibility are still difficult to ascertain.

A computationally attractive alternative to MPC is the Reference Governor (RG) [9]. The RG, initially proposed for linear systems in continuous-time [10] and then extended to, and now mostly used in, discrete-time [11], is an add-on scheme for enforcing pointwise-in-time state and control constraints by modifying, whenever required, the reference to a well-designed stable closed-loop system. A block diagram of RG is shown in Fig. 1.1. To compute $v(t)$, RG employs the so-called maximal output

admissible set (MAS) [12], defined as the set of all the initial inputs and states that ensure constraint satisfaction for all times. To compute $v(t)$, the RG solves a simple linear program based on MAS at every timestep.

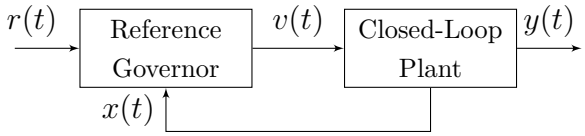

Figure 1.1: Reference governor block diagram, where $y(t)$ is the constrained output, $r(t)$ is the reference, $v(t)$ is the governed reference, and $x(t)$ is the system state (measured or estimated).

In the literature of RG, several works have been proposed in order to tackle uncertainties on model parameters [13, 14] and uncertainties from exogenous disturbances [11, 15–17]. Unknown disturbances are typically assumed to belong to specified compact sets. The MAS is then designed based on the worst case scenario of unrealized disturbances, which leads to a conservative margin that guarantees constraints satisfaction. However, the worst case scenario is highly unlikely to happen in practice, and hence exceedingly conservative control signals (i.e., $v(t)$ in Fig. 1.1) are computed based on this approach. The introduction of probabilistic constraints, also known as chance constraints, alleviates this conservatism as will be explained in more detail in Chapter 3.

Chance constraints have been explored by stochastic MPC [18–20] with some application examples, such as water control [21], power systems [22], and temperature control [23]. Also, works that explore the application of chance constraints to stochastic MAS have been presented in [24,25]. For RG applications, however, chance constraints have received little attention. The work presented in [26] studies the implementation of chance constraints for RG application, by introducing confidence

ellipsoid sets that guarantee probabilistic invariance of a stochastic MAS.

1.2.1.1 Stochastic RG: Problem Statement

The first problem tackled in this dissertation is chance-constraint-based constraint management of linear systems affected by exogenous stochastic disturbances using the RG framework. In order to have a practical solution, the scheme must have similar solves-times as standard RG.

As discussed previously, this is an open problem for stochastic systems, and there is still a lack of solutions within the context of RG that can address it. Chapter 3 tackles this problem and proposes a computationally attractive solution that is finitely determined and positively invariant.

1.2.2 RG for Linear Systems and Recovery From Infeasibility Conditions

The problem of keeping constrained systems under feasible operational conditions, while guaranteeing closed-loop tracking performance, has been studied under various schemes in control theory. One path is to solve the tracking and constraint management problems simultaneously under the Model Predictive Control (MPC) framework [27–31]. Even though this technique has been widely explored in the literature, it tends to be computationally demanding, limiting its applicability, especially for systems with fast dynamics and/or high dimensionality. In addition, theoretical guarantees such as stability are difficult to obtain in practice. Another path, followed by practitioners who want to design the tracking and constraint management por-

tions modularly, is to implement an add-on constraint management mechanism to a closed-loop system with a legacy tracking controller. Examples of the latter are: anti-windup compensation [32], Lyapunov controllers with barrier functions [33], as well as the Command Governor (CG) [9], and the Reference Governor (RG) [34].

For systems that are affected by unmeasured exogenous disturbances, an RG may suffer from feasibility problems that can produce constraint violation. In such situations, there may be no feasible $v(t)$ (see Fig. 1.1) to recover the output from constraint violation.

In the literature, this recovery problem has received little attention. Relevant works, such as the one presented in [3], show different RG and CG solutions to tackle the infeasibility problem to compute the control signal $v(t)$. The most closely-related solution to the recovery problem is the command contraction introduced in [3], which is based on contracting the reference in order to avoid constraint violations. However, undesirable operating modes are induced by plant/model mismatch due to nonlinearities.

Another related publication that studies infeasibility issues for the RG is found in [13], where a robust nonlinear RG is proposed for a fuel-cell system. The authors evaluate the case of parametric uncertainties that may push an RG to infeasibility. They overcome the infeasibility problem by redefining the constraint set based on the nominal plant model and sensitivity functions that consider parametric uncertainties. This solution requires extra computation of the sensitivity functions obtained from the linearization of the plant. A similar work is presented in [35], where a fast RG is proposed. The latter implements a step disturbance observer and a nonlinear/linear compensator. Also, a reduced-order load governor is studied in order to

minimize complexity. The results obtained in [35] show that constraint violations were minimized under plant/model mismatch due to nonlinearities.

1.2.2.1 Recovery RG: Problem Statement

The second problem tackled in this dissertation is the infeasibility conditions that may lead to constraints violations, which arise in standard RG when the constrained linear system is affected by unmodeled unknown disturbances. Within the RG framework, the solution must be capable of finding a solution that recovers the constrained output from constraint violations.

Infeasibility conditions in RG can produce undesirable closed-loop responses that can lead to unsafe operating conditions. Therefore, proposing a solution that can address this problem and still maintains the formulation simplicity and properties of RG is the main goal of the scheme presented in Chapter 4.

1.2.3 Constraint Enforcement in Nonlinear Systems

Constraint management of nonlinear systems has been a challenging problem for control applications, since it needs to handle a compound of non-trivial requirements, such as: formulation complexity, closed-loop stability, real-time implementability, and robustness, to mention a few. Consider, for example, the problem of constraint handling in turbocharged gasoline engines. In this problem, constraints are imposed on the gas temperature at the compressor outlet, as well as the rotational speed of the turbocharger [9, 36, 37]. Imposing these constraints is, however, a challenging

task due to the highly nonlinear nature of the engine. In general, existing constraint management schemes are either conservative (for example, due to the use of static clips on the control input), computationally demanding (for example, due to the use of a Nonlinear Model Predictive Control, i.e., NMPC [38]), or require Lyapunov or Barrier functions that may be difficult to obtain in practice [33, 39, 40].

An approach to handle both constraints and tracking for nonlinear systems is NMPC. However, NMPC may not be feasible for most practical applications due to its complexity and computational burden. An alternative approach to NMPC is to use a set of linear MPC controllers scheduled in real-time based on the operating conditions (e.g., see [41] for an engine control application). However, note that since MPC is not an add-on scheme, an approach like [41] may require the redesign of inner-loop, legacy controllers. On the other hand, there are approaches that handle tracking and constraint management separately. The former is handled by a primary controller (e.g. PID, LQR, etc.), and the latter is tackled by the Reference Governor (RG) [9, 34].

Recently, the standard RG formulation has been extended to handle constraint management of nonlinear systems. Some of these schemes still rely on linear prediction models, for example the work in [3, 35, 42–45]. However, these may not guarantee recursive feasibility [35,42]; demand more significant computational burden depending on the system order, number of inputs, and computational power available for real-time implementation [3, 43]; may lead to a conservative steady-state response [44]; or may not enforce the constraints for all times [45].

Other references have explored the use of RG schemes without linear prediction models, and rely on Lyapunov-based methods [46, 47]. The most recent development

for this type of schemes is Explicit RG (ERG) [48–52], which manipulates the input to a closed loop system continuously so the states always belongs to a safe invariant set. For these Lyapunov-based schemes, the output admissible set is characterized by a positively invariant ellipsoid, which may result in conservative solutions depending on the application. Furthermore, obtaining a global Lyapunov function may be difficult in practice.

1.2.3.1 RG for Nonlinear Systems: Problem Statement

The third problem that is tackled in this dissertation is constraint management of nonlinear system using RG-based formulation. The solution must handle plant model mismatch uncertainty and enforce the constraints. Also, it must be computationally attractive, without introducing excessive steady-state margins, nor extremely conservative solutions during transients.

As discussed previously, current schemes in the literature propose solutions that are based on Lyapunov approaches, which may be overly conservative; and other methods use linear models in their formulation but do not ensure constraint enforcement for all times. Thus, this is still an open problem that is tackled in Chapter 5.

1.3 Original Contributions and Dissertation Outline

This dissertation contributes with the theoretical development of new RG-based schemes, and practical and numerical examples of their implementation. Most of

the content presented in this dissertation has been published or submitted to scientific journals [53] or conference proceedings [45, 54]. Related developments that are not included in this dissertation have been submitted to journals [55] and conferences [56, 57]. The high level contributions of this work are:

- This work contributes to the field of set-theoretic constraint management with focus a on discrete-time dynamical systems. Specifically, we develop novel RG schemes that can handle the problems mentioned in Section 1.2. By contributing to the literature with novel schemes in the RG framework, this dissertation provides practical tools and a theoretical framework for constraint management of a wider class of dynamical systems.

- Even though RG has been in the literature for at least three decades, more work needs to be done in order to increase its industry acceptance. This work contributes on this avenue by proposing efficient algorithms and theoretical guarantees that can lead RG to have a major influence in the industrial scene.

The individual contributions per chapter are below.

Chapter 2 reviews, in more detail, different constraint management schemes that have been developed in the literature. Also, this chapter discusses Model Predictive Control (MPC) to highlight main differences with reference governors schemes. The contributions of Chapter 2 are:

- Provide a comprehensive literature review of different constraint management techniques.

- A technical explanation of RG and the different extensions of it.

- A brief introduction to MPC and the main differences with respect to RG.

Chapter 3 [54] presents a stochastic RG that leverages a Stochastic Robustly invariant MAS (SR-MAS). In order to construct a SR-MAS, we extend the earlier ideas in the literature to Lyapunov stable systems with output constraints. It is shown that the SR-MAS is less conservative than the deterministic approach. The main contributions of Chapter 3 are:

- Analysis of the structural properties of MAS;
- Development of a SR-MAS for Lyapunov stable systems with constrained outputs.
- Development of an inner approximation of the SR-MAS, that can be computed in finite time, and an algorithm to perform this computation.
- Analysis of the effects of feedthrough between constrained output and control input and/or disturbance on the SR-MAS.
- Formulation of a stochastic RG that offers a less conservative response compared to standard RG theory.

Chapter 4 [45] presents a novel RG formulation, named as Recovery RG (RRG) applied to linear systems affected by exogenous disturbances, specifically those large enough to cause constraint violation. This formulation is based on a set-theoretic approach which includes external disturbances estimation within a new RG. The idea is to update the MAS and compute a constraint-admissible input that recovers the system from constraint violation at steady-state. The main contributions of Chapter 4 are:

- Formulation of the RRG scheme that can recover from constraint violation when unknown exogenous disturbances affect a system;
- Analysis of the recursive feasibility of the RRG;
- Theoretical guarantees of recovery from constraint violation by applying the RRG scheme;
- Comparison between the standard RG and the RRG applied to a turbocharged gasoline engine model.

Chapter 5 [53] presents a scheme to enforce constraints on nonlinear systems. The scheme decomposes the design of the constraint management strategy into two parts: enforcement at steady-state, and during transient. The former is achieved by using the forward and inverse steady-state characterization of the nonlinear system. The latter is achieved by implementing an RG-based approach, which employs a novel Robust Output Admissible Set (ROAS), which is obtained using data from the nonlinear system. The main contributions of this chapter are:

- A novel RG scheme for constraint management of nonlinear dynamical systems, which maintains desirable properties of standard RG such as recursive feasibility, formulation simplicity, and closed-loop stability. The inclusion of the forward and inverse steady state map for constraint management, as well as the ROAS for transient satisfaction, are also novel contributions.
- Theory, as well as methods and algorithms, to construct and tune the ROAS based on experimental data and linear systems theory.

- Extension of the theory from single-output to multi-output nonlinear systems, as well as systems whose steady-state characterization may vary slowly with time, which arise in practical applications.
- Validation of the scheme using simulations and experimentation on a turbocharged engine.

Chapter 6 shows two practical simulation examples of nonlinear systems and the implementation of TR-RG to them. The first system is an example of a vehicle dynamics control application, where the idea is to enforce rollover stability by implementing TR-RG. The second example is about an inverted pendulum, where TR-RG is implemented to enforce constraints on the rate of change of the arm that maintains the pendulum in the upright position. The main contributions of this chapter are:

- Study the versatility of TR-RG when a system does not satisfy all the assumptions for the implementation of TR-RG.
- Test the robustness of TR-RG for the cases when there are multiple outputs.

Chapter 7 presents the concluding remarks and observations of the work presented in this dissertation. This chapter also shows the future works and possible extensions of the theory presented here.

1.4 Statement of Impact

The development of new constraint management techniques for stochastic, linear, and nonlinear systems has relevant implications on both theoretical and practical

fronts. The former is seen for instance, in the stochastic RG, where the concept of chance constraint is used to build a SR-MAS, which is positively invariant and finitely determined. By developing this, we are extending the applicability of the theory of RG to constraint management of stochastic systems. Also, by introducing the new concept of ROAS, which can be constructed based on data, we are incorporating a new form to robustify a constraint management technique based on linear system theory that can be applied to nonlinear systems. On the practical aspect, by proposing these new schemes that tackle issues like conservativeness of solutions, infeasibility conditions, and robustness against plant/model mismatch; and providing algorithms for their implementation, we are offering new tools to robustly design stochastic, linear, and nonlinear closed-loop systems with constraint awareness capabilities and their use in an industrial environment.

Throughout the work developed for this dissertation, we were able to share with the control community, through journals and international conferences, our technical contributions in the field of constraint management and reference governors. This dissemination of knowledge has the intention of producing a significant impact on the community interested in constraint awareness and set-theoretic approaches. Also, as all new theories, we are on the shoulders giants, hence we are contributing to build knowledge on top of the well-built-foundations established by great minds in the field of constrained control.

1.5 Notation

The following notations are used throughout this dissertation. $\mathbb{Z}_+$ denotes the set of all non-negative integers. The set $\mathbb{Z}_t$ denotes all non-negative integers up to t, where t is a positive integer. Let $V, U \subset \mathbb{R}^n$. Then, $V \sim U := \{z \in \mathbb{R}^n : z + u \in V, \forall u \in U\}$ is the Pontryagin-subtraction (P-subtraction). The interested reader can find more details about the P-subtraction in [16]. The identity matrix is denoted by I. The ball with radius r centered at c is denoted as $\mathcal{B}_r(c) = \{z : \|z - c\| \leq r\}$. The interior of the set V is denoted as $\text{int}(V)$. The zero matrix is denoted by $0_{n\times n}$. The variable $t \in \mathbb{Z}_+$ is the discrete time. The inequality $Ax \leq b$ is component wise. Throughout the dissertation, x is used to denote a system state, output is denoted by y, u is used for the control input command, and w is used to denote disturbances. Capital letters are used to denote matrices, and lower case are for scalars and vectors.

Chapter 2

Review of Constraint Management Schemes

2.1 Literature Review

Control engineers need to gather and analyze a significant amount of data (e.g., input and output signals) in order to deliver a control law capable of satisfying stability, tracking performance and disturbance rejection properties. This may sometimes be a difficult task to accomplish. However, thanks to the advances in control theory, more reliable control laws can be designed for a broad range of practical applications nowadays. In addition to the properties mentioned before, constraint management is quite important for all real applications, i.e., to have control laws that can guarantee performance without violating physical and safety constraints under different operating conditions. Since the majority of control systems are exposed to constraints and performance is stretched to its physical limitations, it becomes highly important to develop mechanisms that enforce constraints while preserving desirable characteristics of the transient response.

This chapter gathers some of the most recent works related to constraint management schemes. The main focus is around Reference Governor. However, other strategies are analyzed in order to understand strengths and weaknesses.

2.1.1 Reference Governors

Reference governor was first proposed by Kapasouries in 1988 [58]. Since then, several governor techniques have been proposed. The list includes and is not limited to: scalar and vector reference governors, command governors, extended command governors, decoupled reference governor, recovery reference governor, stochastic reference governor, incremental reference governors, feedforward reference governors, network

reference governors, reduced order reference governors, explicit reference governor, generalized reference governor, distributed reference governors, parameter governors, and virtual state governors. Even though there are significant variations in terms of the formulation and implementation of these techniques, the main idea is to protect the system from any possible constraint violation and try to preserve, whenever possible, the response of the closed loop system designed by conventional control techniques. One form to categorize the different governor schemes is by dividing them as governors methods for linear and nonlinear systems.

2.1.1.1 Governor Schemes for Linear Systems

2.1.1.1.1 Review of Maximal Output Admissible Sets

This section reviews the theory presented in [11,12]. Consider the closed-loop discrete-time linear time-invariant (LTI) system given by:

$$\begin{aligned} x(t+1) &= Ax(t) + Bv(t) \\ y(t) &= Cx(t) + Dv(t), \end{aligned} \tag{2.1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $v(t) \in \mathbb{R}^m$ is the input, and $y(t) \in \mathbb{R}^p$ is the constrained output vector. Over the latter, the following constraint is imposed: $y(t) \in Y$, $\forall t \in \mathbb{Z}_+$, where $Y \subset \mathbb{R}^p$ is a specified compact polytope with the origin in its interior. Similar to [11, 12, 15] the following assumptions are made:

A. 2.1.1. *System (2.1) reflects the combined closed-loop dynamics of the plant with a stabilizing controller. Consequently, the system is asymptotically stable (i.e.* $|\lambda_i(A)| < 1, i = 1, ..., n$*). Furthermore, it is assumed that the pair* (C, A) *is observable.*

The common feature of most reference governor schemes proposed in the literature is that they compute at each time instant a command $v(t)$ such that, if it is constantly applied from the time instant t onward, the ensuing output will always satisfy the constraints. This idea is used to create what is known as the Maximal Output Admissible Set.

2.1.1.1.1.1 Maximal Output Admissible Sets (MAS)

The MAS, denoted by O_∞, is defined as the set of all initial states and inputs, such that the output constraints are satisfied for all future times. To characterize the MAS for system (2.1), the following assumption is made:

A. 2.1.2. *In the construction of the MAS for (2.1), the control signal $v(t)$ is assumed to be constant for all times, i.e., $v(t+1) = v(t), \forall t \in \mathbb{Z}_+$.*

Using A.2.1.2, $y(t)$ is expressed as a function of the initial state, x_0, and the constant input, $v(t) = v_0$:

$$y(t) = CA^t x_0 + C(I - A)^{-1}(I - A^t)Bv_0 + Dv_0. \tag{2.2}$$

Based on (2.2), the MAS can be characterized as:

$$O_\infty := \{(x_0, v_0) \in \mathbb{R}^{n+m} : y(t) \in Y, \forall t \in \mathbb{Z}_+\} \tag{2.3}$$

where $y(t)$ is given in (2.2). The set O_∞ defined in (2.3) is characterized by an infinite number of halfspace intersections, which is impossible to compute in finite time. However, a close inner approximation of O_∞, which can be computed in finite time, is readily available. To show this, the following set of steady-state admissible

inputs is introduced:

$$\bar{V} := \{v_0 \in \mathbb{R}^m : H_0 v_0 \in Y_{ss}\}. \tag{2.4}$$

where $H_0 = C(I - A)^{-1}B + D$ is the DC gain of system (2.1) from $v(t)$ to $y(t)$, and $Y_{ss} := (1 - \epsilon)Y$ for some $0 < \epsilon \ll 1$. Using $\bar{V}$, an inner approximations to O_∞ can be found, this is:

$$\bar{O}_\infty := \{(x_0, v_0) \in \mathbb{R}^{n+m} : H_{0_v} v_0 \in Y_{ss}, y(t) \in Y, \forall t \in \mathbb{Z}_+\}. \tag{2.5}$$

Definition 2.1.1. *The set $\bar{O}_\infty$ is finitely determined if there exists $t^* \in \mathbb{Z}_+$ such that $\bar{O}_{t+1} = \bar{O}_t, \forall t \geq t^*$. This also implies that $\bar{O}_\infty = \bar{O}_{t^*}$ [12]. Where $\bar{O}_{t^*}$ is given by:*

$$\bar{O}_{t^*} := \{(x_0, v_0) \in \mathbb{R}^{n+m} : H_0 v_0 \in Y_{ss}, y(j) \in Y, j = 1, \ldots, t^*\}.$$

For system (2.1) satisfying assumption A.2.1.1 and with the pair (C, A) observable. Then, $\bar{O}_\infty$ is finitely determined [59].

It was proved in [60] that properties of O_∞ are inherited from properties of Y. If Y is convex, closed, symmetric, and $0 \in \text{int}(Y)$ then O_∞ is convex, closed, symmetric, and $0 \in \text{int}(O_\infty)$. Also, it was presented in [60] that if A is Schur, $[A, C]$ observable, and Y compact, then the maximal output admissible set O_∞ is finitely determined. This last result has contributed enormously to the implementation of the reference governor, since it allows a finite computation of the maximal output admissible set (2.5).

For the discussion that follows, the constraint set Y is assumed to be a convex polytope, defined as :

$$Y := \{Sy \leq s\} \tag{2.6}$$

Based on (2.6) we re-define $\bar{O}_\infty$ as:

$$\bar{O}_\infty = \{(v, x) : H_x x + H_v v \leq h\}, \tag{2.7}$$

where the first rows of H_x, H_v, and h describe the steady state characteristics of the response (with the ϵ margin) and the remaining rows are given by SCA^t, $S(C(I - A^t)(I - A)^{-1}B + D)$, and s, respectively, where s is the constraint or constraints that need to be satisfied. With some abuse of notation, in the rest of the document O_∞ will be used to refer to (2.5) or (2.7).

Before getting into the details of the different reference governors, we first explain how a MAS is computed when a LTI is affected by disturbances.

2.1.1.1.1.2 MAS for Systems Affected by Additive Disturbances

Consider the discrete-time LTI system given by:

$$\begin{aligned} x(t+1) &= Ax(t) + B_v v(t) + B_w w(t) \\ y(t) &= Cx(t) + D_v v(t) + D_w w(t) \end{aligned} \tag{2.8}$$

where in addition to the conditions of (2.1), the disturbance input satisfies $w(t) \in W$, where $W \subset \mathbb{R}^d$ is also a compact polytope with the origin in its interior. System (2.8) satisfies assumptions A.2.1.1, and for the construction of the MAS A.2.1.2 is satisfied. Beside, the following assumption is imposed:

A. 2.1.3. *The disturbances $w(0), w(1), \ldots$ are assumed to be independent and identically distributed (i.i.d) random vectors with a probability density function $f_w : W \mapsto$*

$\mathbb{R}_+$ *and* $\int_W f_w(w)dw = 1$.

In order to define the MAS for system (2.8), we write $y(t)$ as a function of the initial state, x_0, and the constant input, $v(t) = v_0$, which is possible thanks to assumption (A.2.1.2):

$$\begin{aligned} y(t) = CA^t x_0 + C(I-A)^{-1}(I-A^t)B_v v_0 + D_v v_0 \\ + C\sum_{j=0}^{t-1} A^{t-j-1}B_w w(j) + D_w w(t) \end{aligned} \tag{2.9}$$

We now define the sets Y_t using the following recursion:

$$\begin{aligned} Y_0 &= Y \sim D_w W \\ Y_{t+1} &= Y_t \sim CA^t B_w W \end{aligned} \tag{2.10}$$

P-subtraction allows us to rewrite the requirement $y(t) \in Y, \forall\{w(j)\} \in W, j = 0, \ldots, t$ as:

$$CA^t x_0 + (C(I-A)^{-1}(I-A^t)B_v + D_v)v_0 \in Y_t$$

Now, consider the set O_t given below, which defines the set of all initial states and inputs such that the output constraints are satisfied from time 0 to time t:

$$O_t := \bigcap_{i=0}^{t} P_i \tag{2.11}$$

where P_i are defined by:

$$\begin{aligned} P_0 &:= \{(x_0, v_0) \in \mathbb{R}^{n+m}, Cx_0 + D_v v_0 \in Y_0\}, \\ P_i &:= \{(x_0, v_0) \in \mathbb{R}^{n+m}, \\ &\qquad CA^i x_0 + (C(I-A)^{-1}(I-A^i)B_v + D_v)v_0 \in Y_i\} \end{aligned} \tag{2.12}$$

The MAS, denoted by O_∞, is the set of all safe initial conditions and inputs, such that for any given disturbance, the output constraints are satisfied for all times. The set O_∞ is defined as the limit of O_t:

$$O_\infty := \bigcap_{i=0}^{\infty} P_i \tag{2.13}$$

Since Y and W are polytopic sets (i.e., defined by an intersection of halfspaces), each O_t is also polytopic. Thus, the set O_∞ defined in (2.13) requires an infinite number of halfspace intersections, similar to (2.3). Hence, a close inner approximation of the set O_∞ is used. To show this, introduce the set of all steady-state admissible inputs:

$$V := \{v_0 \in \mathbb{R}^m : H_0 v_0 \in Y_{ss}\} \tag{2.14}$$

where $H_0 = C(I - A)^{-1}B_v + D_v$ is the DC gain of system (2.8), and Y_{ss} represents the limit of Y_t, i.e., $Y_{ss} = \lim_{t\to\infty} Y_t$. Note that Y_{ss} requires infinite computations; therefore, we shrink this set by introducing $\bar{Y} := (1 - \epsilon)Y_t$ for some $0 < \epsilon \ll 1$ and large t, and thus define an inner approximation of V:

$$\bar{V} := \{v_0 \in \mathbb{R}^m : H_0 v_0 \in \bar{Y}\} \tag{2.15}$$

Using $\bar{V}$, close approximations to P_t and O_t can be found:

$$\begin{aligned}
\bar{O}_t &:= \bigcap_{i=0}^{t} \bar{P}_i \\
\bar{P}_i &:= P_i \cap (\bar{V} \times \mathbb{R}^n) \\
&= \{(x_0, v_0) \in \mathbb{R}^{n+m} : H_0 v_0 \in \bar{Y}, \\
&\quad CA^i x_0 + (C(I-A)^{-1}(I-A^i)B_v + D_v)v_0 \in Y_i\}
\end{aligned} \tag{2.16}$$

Finally, the set $\bar{O}_\infty$ is defined as the limit of $\bar{O}_t$, and is an inner approximation of O_∞.

Theorem 2.1.1. *Suppose system* (2.8) *is asymptotically stable, with the pair* (C, A) *observable and* $0 \in \text{int}(\bar{Y})$. *Then,* $\bar{O}_\infty$ *is finitely determined.*

For proof, see [16].

2.1.1.1.2 Scalar Reference Governor (RG)

RG was first proposed in the continuous time framework, however a natural extension to the discrete time domain has been widely adopted due to its mathematical simplicity. In this last framework, the static RG was first introduced by Gilbert in 1994 [61]. The command $v(t)$ (see Fig. 1.1) is modified by:

$$v(t) = \kappa r(t)$$

where the parameter $\kappa \in [0, 1]$ is maximized subject to $(x(t), v(t)) \in O_\infty$. The static RG has the disadvantage of generating oscillations on the command signal $v(t)$ when constraint violations are detected. Therefore, the dynamic RG substituted the static approach and the first works of this new scheme appeared in 1995 proposed by Bemporad [62], Gilbert [11] and Kolmanovsky [17].

From (2.2) and (2.5), it is possible to see that O_∞ contains the predictions of the output based on the current states and the input. Based on the predictions, the controller

can anticipate if a constraint may be violated and then take corrective actions over the reference. The idea behind RG is to calculate $v(t)$ based on $\bar{O}_\infty$ by implementing the following dynamic equation:

$$v(t) = v(t-1) + \kappa(r(t) - v(t-1)) \tag{2.17}$$

At each timestep, the RG solves the following linear programming problem:

$$\begin{aligned} \underset{\kappa \in [0,1]}{\text{maximize}} \quad & \kappa \\ \text{s.t.} \quad & (x_0, v_0) \in \bar{O}_\infty \\ & v_0 = v(t-1) + \kappa(r(t) - v(t-1)) \\ & x_0 = x(t) \end{aligned} \tag{2.18}$$

where κ is the factor that manipulates $v(t)$ along the line defined between $v(t-1)$ and $r(t)$. Note that $\kappa = 0$ means that in order to keep the system safe, $v(t) = v(t-1)$, and $\kappa = 1$ means that no violation was detected and, therefore, $v(t) = r(t)$. A graphical representation of how RG provides a solution that satisfy the constraints is presented in Fig. 2.1.

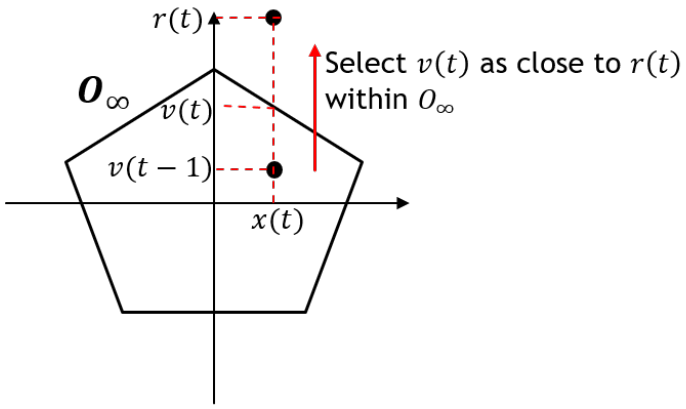

Figure 2.1: RG and O_∞.

2.1.1.1.3 Vector Reference Governor (VRG)

Note that (2.17) and (2.18) can be implemented to a Multi-Inputs Multi-Outputs (MIMO) system. However, since (2.18) solves for one decision variable, κ, this implies that the solution will prioritize the input with higher threat of constraint violation, i.e., RG will lower the input value to avoid constraint violation. Hence, the other input channels may be affected and performance may be deteriorated. When this problem was identified, the Vector Reference Governor (VRG) was proposed to solve it. In VRG a diagonal matrix $\mathbf{K}$ is used in order to command each input-output channel independently [11]. The VRG formulation is defined as:

$$v(t) = v(t-1) + \mathbf{K}(r(t) - v(t-1)) \tag{2.19}$$

where $\mathbf{K} = \text{diag}(\kappa_i(t))$. The values of $\kappa_i(t), i = 1, ..., m$, are chosen to minimize:

$$\begin{aligned} \underset{\kappa_i \in [0,1]}{\text{minimize}} \quad & \|v(t) - r(t)\| \\ \text{s.t.} \quad & v(t) = v(t-1) + K(r(t) - v(t-1)) \\ & (x(t), v(t)) \in O_\infty \end{aligned} \tag{2.20}$$

2.1.1.1.4 Command Governor (CG)

CG tackles the problem by computing $v(t)$ directly, which is the main different with respect to VRG. This technique was proposed by Bemporad [63] and Casavola [64] and it computes $v(t)$ by solving at each time step the quadratic program:

$$\begin{aligned} \underset{v}{\text{minimize}} \quad & \|\text{v(t)} - \text{r(t)}\|^2_{\text{Q}} \\ \text{s.t.} \quad & (x(t), v(t)) \in \bar{O}_\infty \end{aligned} \tag{2.21}$$

with $Q > 0$. One of the advantages of this scheme is that it offers the possibility to have

solutions that satisfy, $|v(t)| < |v(t-1)|$, which is not possible for RG and VRG. However, this approach demands more computational effort compared with the RG.

2.1.1.1.5 Extended Command Governor (ECG)

This technique was used first in 1997 by Bemporad [63] and years later an important work coming back to it was proposed by Gilbert in 2011 [65]. It is based on the notion that predictions (2.2) are a special case of:

$$y(t) = CA^t x_0 + C(I-A)^{-1}(I-A^t)B\hat{v} + D\hat{v} \tag{2.22}$$

where $\hat{v}$ is the sum of two parts. One that comes out of an optimization problem $\bar{\rho}$ and a vanishing part $\hat{\mu}(\cdot)$, that is:

$$\hat{v}(\cdot) = \bar{\rho} + \hat{\mu}(\cdot) \tag{2.23}$$

the variable $\hat{\mu}$ is the output of a fictitious autonomous system, given by:

$$\begin{aligned} \chi(t+1) &= A_\chi \chi(t) \\ \hat{\mu}(t) &= C_\chi \chi(t) \end{aligned} \tag{2.24}$$

where A_χ is Schur matrix. The initial conditions of (2.24) are other design decision that can be used to improve performance of the ECG. The optimization problem defined for ECG is:

$$\begin{aligned} (\bar{\rho}(t), \chi(t)) = \arg\min_{\bar{\rho},\chi} \quad & \frac{1}{2}\|\bar{\rho} - \mathrm{r(t)}\|_{\mathrm{Q}}^2 + \frac{1}{2}\|\chi\|_{\mathrm{P}}^2 \\ \text{s.t.} \quad & \left(\bar{\rho}, \begin{bmatrix} x(t) \\ \chi \end{bmatrix}\right) \in O_\infty \end{aligned} \tag{2.25}$$

where χ is the state for the fictitious autonomous system, $P > 0$ satisfies the discrete time Lyapunov inequality, $A_\chi^\top P A_\chi - P < 0$, and O_∞ is defined based on the predictions (2.22).

Thanks to the introduction of (2.24), ECG has a larger domain of attraction with respect to CG/RG. Also ECG offers a faster response (especially in the case of systems with rate limited actuators).

2.1.1.2 Governor Schemes for Nonlinear Systems

This section explores RG-based schemes that are applied to nonlinear systems. The schemes used for nonlinear systems follow, in some sense, the same ideas presented for linear systems in the previous section.

2.1.1.2.1 RG and CG Design Based on Model Linearization

A heuristic approach proposed by Vahidi can be found in [35], where three different methods were proposed to overcome the challenges of implementing the RG on a nonlinear plant. The first approach implements a look-up table that gathers the steady-state values for the states that correspond to different input command signals, denoted as $\Gamma(v(t))$. Based on this, an adjustment state is computed as:

$$x_{adj} = x_{non}(t) - \Gamma(v(t)) + (I - A)^{-1} B \delta v(t), \tag{2.26}$$

where A and B are obtained from the linearized model, $\delta v(t)$ is the input to the linearized plant, and $x_{non}(t)$ corresponds to the measured nonlinear states at each time step. Note that availability of the nonlinear states is important to implement this method. The adjusted state from (2.26) is fed to O_∞, which was computed based on a linear model. A second approach presented in [35] is to treat the difference between the constrained outputs of the

nonlinear system and linear model as a state with constant dynamics, this is:

$$\begin{aligned} x(t+1) &= Ax(t) + Bv(t) \\ w(t+1) &= w(t) \\ y(t) &= Cx(t) + Dv(t) \end{aligned} \tag{2.27}$$

where the extra state $w(t) = y_{non}(t) - (\delta y(t) + y_{eq})$, with $y_{non}(t)$ being the nonlinear constrained output that needs to be available at each time step, and y_{eq} represents the constrained output around the linearization point. One of the drawbacks of this approach is that the constrained output has to be measurable, which may not always be the case. Also, O_∞ based on (2.27) becomes:

$$O_\infty := \{(x, \delta v, w) \in \mathbb{R}^{n+m+p} : Cx + D\delta v + w \in Y\}. \tag{2.28}$$

Note that the dimensionality of the set defined in (2.28) grows with the number of artificial states w, which depends on the number of constraint outputs. Also, since the assumption behind the augmented state, $w(t)$, is to be constant, recursive feasibility (i.e., obtaining a solution, $v(t)$, at each time step) may not be guaranteed. In order to overcome the issue of dimensionality, a third solution based on the reduction of the linearized model is proposed in [35]. This last solution suffers of conservatism of the input signal commanded to the system. A comparison of the computational burden of each one of the methods proposed in [35] showed that the reduced order scheme has the lowest number of flops operations performed per time step.

An application of the approach that considers the difference between the nonlinear and linear outputs is presented in [42]. Another useful method to approach nonlinear systems is through the implementation of feedback linearization, which renders the state dynamics linear and hence the prediction of the state evolution becomes computationally

straightforward. However, feedback linearization transformations may lead to non-convex admissible sets, for those cases, [66] and [67] propose an approximation of O_∞ through convex regions and mixed-logical-dynamic.

In [3] different approaches are presented to solve constraint management on nonlinear systems by leveraging linear models. One uses a nonlinear/linear compensator that considers the difference between the linear and nonlinear constrained outputs (similar to [35] that uses (2.27)); another approach implements a multi-point linearization, which generates different O_∞ sets. Since nonlinearities may push the RG to infeasibility conditions, [3] proposes a solution that the authors called constrained command RG, which basically computes the governed input that is in a minimum distance from the reference and it satisfies the constrains.

An alternative for constraint management of nonlinear systems based on linear approaches is studied in [68], where the nonlinear system is embedded into a Linear Parameter Varying (LPV) system that is controlled through an RG and a gain-scheduled tracking algorithm. A similar approach is presented in [69], where CG is implemented based on embedding the nonlinear system model into a family of Linear Time Varying polytopic uncertain models. We will discuss more in detail about LPV in Section 2.1.1.3.

2.1.1.2.2 RG and CG Design Based on Nonlinear Model

The first approach, known as Robust RG, was proposed by Bemporad [46]. This technique requires to solve the following optimization problem:

$$
\begin{aligned}
&\kappa(t) = \max_{\kappa \in [0,1]} \quad \kappa \\
&\text{s.t.} \quad v = v(t-1) + \kappa(r(t) - v(t-1)) \\
&\hat{x}(t+k+1|t) = f(\hat{x}(t+k|t), v) \\
&h(\hat{x}(t+k|t), v) \leq 0, k = 0, ...k, k^* \\
&h(\bar{x}_v, v) \leq -\varepsilon
\end{aligned}
\tag{2.29}
$$

where $\hat{x}$ is the predictive trajectory propagation of the state, which is computed by simulating the nonlinear model. The equilibrium of the state is defined as $\bar{x}_v = f(\bar{x}_v, v)$. The parameter $\varepsilon > 0$ is sufficiently small and $k^* \in \mathbb{Z}_+$ is sufficiently large such that if the constraints are satisfied up to k^*, then they are going to be satisfied for $k > k^*$ [46]. In general this is a non-convex problem due to $f(\cdot)$. However, since (2.29) is a scalar optimization problem, a solution can be found by bisections or grid search. Other similar works were proposed by Gilbert and Kolmanovsky [47], Garone [49] and Nicotra [48, 70, 71].

In [47] a generalization of both prediction-based and Lyapunov function-based nonlinear reference governors applicable to constrained systems with disturbances is presented. The idea of this method is to define a continuous function $S(x, v)$ so that, for any pair $(x(t), v)$ such that $S(x(t), v) \leq 0$ is satisfied, if v is kept constant from t onward, the trajectory $\hat{x}(t+k|t), k \geq 0$ is:

- Safe: constraints are never violated, i.e $h(\hat{x}(t+k|t), v) \leq 0, \forall k \geq 0$;
- Strongly returnable: there exists a finite integer k^*, which may depend on $x(t)$, such that $S(\hat{x}(t+k^*|t), v) < 0$, which means that after a finite time k^* the trajectory returns to the interior of the set $\{(x, v) : S(x, v) \leq 0\}$.

Given such a function $S(\cdot)$, $\kappa(t)$ can be chosen at each time instant t by solving the following scalar optimization problem:

$$\begin{aligned}
&\kappa(t) = \max_{\kappa \in [0,1]} \quad \kappa \\
&\text{s.t.} \quad S(x(t), v(t-1) + \kappa(r(t) - v(t-1))) \leq 0 \\
&h(\bar{x}_v, v(t-1) + \kappa(r(t) - v(t-1))) \leq -\varepsilon
\end{aligned} \tag{2.30}$$

A similar approach is presented in [72], where O_∞ is built based on the nonlinear plant trajectories. However, a safe margin is built on the constraint set in order to account for the difference between the nonlinear and linear plant, and parameters uncertainties on the model. Specifically, if the nonlinear system function is given by:

$$\begin{aligned}
\dot{x} &= f(x_n, r, \theta_0) \\
y_{non} &= g(x_n, r, \theta_0).
\end{aligned} \tag{2.31}$$

Under the assumption of having continuous differentiable f and g functions, a sensitivity function can be computed as a solution of the system linearization. This is performed around the nominal values of the parameters and the worst cases of parameter variations. The constraint $y_{non} \in Y$, where Y is the constraint set, becomes:

$$y_{non} + (\delta_\theta^z)^\top (\theta - \theta_0) + M\|\theta - \theta_0\|^2 \mathcal{B} \subset Y, \forall \theta \in \Theta, \tag{2.32}$$

where δ_θ^z is the sensitivity function based on the trajectories of the linearized system; $\mathcal{B}$ is the unit ball; M considers the difference between the nonlinear and linearized system. Note that (2.32) considers the distance among parameters that consider uncertainties.

2.1.1.2.3 Explicit RG (ERG)

The idea is to ensure constraint enforcement by continuously manipulating $v(t)$ so $x(t)$ always belongs to a safe invariant set centered in the steady state $\bar{x}_v(t)$. This scheme is based on Lyapunov function $V(x, v)$ and a bound of this function $\Gamma(v)$. The reference is

computed based on the integration of:

$$\dot{v} = k_{\Gamma}[\Gamma(v) - V(x,v)]\frac{r-v}{max\{\|r-v\|,\varepsilon\}} \tag{2.33}$$

where k_{Γ} and ε are design parameters. The main difficulty of this approach is to find a suitable function $V(x,v)$. Also, by the nature of a Lyapunov-based formulation, ERG may lead to conservative solutions. Main works of this scheme have been proposed by Nicotra [48, 70, 71, 73, 74], and Garone [49].

A extension of this work developed by Nicotra and Garone is presented in [50], where the governed command is computed based on the dynamic function:

$$\dot{v} = \delta(x,v)\rho(r,v), \tag{2.34}$$

where $\delta(x,v)$ is the dynamic safety margin that can be computed based on four methods: trajectory based approach, Lyapunov based approach (similar to finding the threshold Lyapunov function, $\Gamma(v)$, in (2.33)), invariance-based approach, and returnability-based approach. The other part of (2.34) is the navigation field function, which corresponds to $\rho(r,v)$. The latter may be computed based on a waypoint method or a vector field method. The dynamic safety margin presented in [50] is a continuous function that considers the systems' dynamics. On the other hand, the navigation field function provides the trajectory that needs to be followed by the input command, v, up to the reference r. One of the main assumptions in [50] is that the reference is steady-state admissible, which may not always be the case, however, they addressed the case of non admissible references by introducing a distance function that is minimized subject to output constraints satisfaction. An extension to discrete-time systems with parametric uncertainties is presented in [52].

2.1.1.2.4 Parameter Governor

Proposed by Kolmanovsky and Sun J. [75,76], the main idea of parameter RG for nonlinear systems is to adjust a parameter $\theta(t) \in \Theta$, of a nominal control law so as to optimize over a finite horizon the predicted system response subject to constraints. The objective function to minimize is:

$$J(t) = \|\theta\|^2_{\psi_\theta} + \sum_{k=0}^{T} \Omega(\hat{x}(t+k|t), \theta(t), r(t)) \tag{2.35}$$

The parameter θ is assumed to remain constant over the prediction horizon. This reduces computational and implementation effort, and simplifies the analysis. The system model is defined as:

$$\begin{aligned} x(t+1) &= f(x(t), u(t)) \\ x_i(t+1) &= x_i(t) + z(t) - r \\ z(t) &= h_z x(x(t)) \end{aligned} \tag{2.36}$$

where z is the output that is supposed to track the reference, r, and the control law $u(t)$ incorporates the parameter $\theta(t)$. This scheme has similarities with model predictive control, that is analyzed in Section 2.1.2.

2.1.1.2.5 Output Feedback RG

This approach presents a formulation for nonlinear systems with unmeasurable states [77]. This method utilizes an ellipsoidal region in which the state is guaranteed to lie. Such a region can be obtained by using the set-valued observer [78]. For this scheme, in the presence of noise and/or disturbance, somewhat conservative conditions are introduced for the finite-time settling of the modified reference to the original one and for the convergence of the state to the neighborhood of the equilibrium.

2.1.1.2.6 Incremental RG

Proposed by Tsourapas [79] in order to reduce the computational effort to solve the problem

(2.29), which is distributed over time by checking the feasibility of a single value of $v(t)$ that differs from $v(t-1)$ by a fixed and "small enough" step size.

So far, some RG-based schemes have been presented for linear and nonlinear systems. In most of the methods that have been discussed, the main common factor is that they all rely on some type of admissible set to compute an input command that satisfies the constraints. In the methods discussed previously, the admissible set may be computed based on LTI models, Lyapunov methods (i.e., invariant ellipsoids), or considering nonlinear states trajectories. Next, we present the case for Linear Time Varying (LTV) or Linear Parameter Varying (LPV) systems.

2.1.1.3 Set Theoretic Approaches to Build Robust Invariant Sets for Linear and Nonlinear Systems with Polytopic Uncertainty

In this section, we study the techniques based on polytopic set theory approaches that have been proposed for linear time varying (LTV) or linear parameter varying (LPV) systems of the form:

$$\begin{aligned} x(t+1) &= A(t)x(t) + B(t)v(t), \\ y(t) &= Cx(t) + Du(t) \in Y, \end{aligned} \tag{2.37}$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^p$, and $v \in \mathbb{R}^m$. It is assumed that assumption A.2.1.2 holds to build the robust invariant set, hence we can reformulate (2.37) as:

$$\begin{aligned} x(t+1) &= \Phi(t)x(t) \\ y(t) &= \bar{C}x(t) \in Y, \end{aligned} \tag{2.38}$$

where $\Phi(t) = \begin{bmatrix} A(t) & B(t) \\ 0 & I_m \end{bmatrix}$ and $\bar{C} = \begin{bmatrix} C & D \end{bmatrix}$. The constraint set Y is a polytope with $0 \in Y$, which is defined as:

$$Y = \{y : A_y y \leq b_y\} \tag{2.39}$$

It is assumed that $\Phi(t)$ belongs to a given uncertainty polytope:

$$\Delta = \left\{ \Phi \in \mathbb{R}^{(n+m)\times(n+m)} : \Phi = \sum_{i=1}^{L} \lambda_i \Phi_i, \sum_{i=1}^{L} \lambda_i = 1, \lambda_i \geq 0 \right\}. \tag{2.40}$$

Also, it is assumed that (2.38) is robustly asymptotically stable, this is:

$$\lim_{k\to\infty} \left(\max_{\Phi(t)\in\Omega, t=0,\ldots,k-1, ||x(0)||=1} ||x(k)||_2 \right) = 0 \tag{2.41}$$

The problem of constraints satisfaction for systems with polytopic uncertainty has been addressed in several works in the literature. For instance, [80] proposes an efficient algorithm to build a robust invariant admissible sets for (2.38) using the vertices of the uncertainty polytope and by imposing state and input constraints on the predictions. It is shown in [80] that the resulting set is maximal admissible and that it converges in finite time when condition (2.41) is satisfied. The robust positive invariant (RPI) set presented in [80] has the form:

$$O_\infty^r = \cap_{i=0}^{k} S_i = \left\{ x \in \mathbb{R}^{n+m} : A_{s_i} x \leq b_{s_i} \right\} \tag{2.42}$$

where initially $A_{s_0} = A_y C$, $b_{s_0} = b_y$, and for $i = 1, \ldots, k$:

$$A_{s_i} = \begin{bmatrix} A_{s_{i-1}}\Phi_1 \\ \vdots \\ A_{s_{i-1}}\Phi_L \end{bmatrix}, \qquad b_{s_i} = \begin{bmatrix} b_{s_{i-1}} \\ \vdots \\ A_{s_{i-1}} \end{bmatrix} \tag{2.43}$$

Other works explore different strategies to compute RPI sets for systems like (2.38)

under condition (2.40). [81] uses the ideas of contractive sets, where it is stated that if a set is λ-contractive, with $\lambda \in [0, 1]$, then it is robustly positively invariant for a system like (2.38). The construction of RPI sets based on geometrically motivated methods for systems affected by unknown disturbances is explored in [82]. The work in [82] can be considered an extension of the work presented in [80] to systems affected by additive disturbances and an improvement on the algorithm to compute RPI sets. Furthermore, it is shown in [82] that by implementing what they called a tree-structured algorithm, it is possible to significantly reduce the number of linear programs to build the minimal representation of the RPI set. Open source software that support the work presented in [82] can be found in [83, 84].

It is proposed in [85] an inner approximation of the maximal RPI set can be found by starting from an initial RPI set, which is sequentially enlarged by adding at each step vertices until no further addition is possible. Other works that explored the construction of RPI sets for systems like (2.38) can be found in [86–91].

2.1.1.3.1 Data-driven Approach

A data-driven approach uses the measurements from the system or model to build an RPI set. The work presented in [92] proposes an approach of this nature to approximate a control RPI set while simultaneously picking an optimal admissible uncertain model. The latter is defined as a model that explains a finite measurement history. The type of model that is considered in [92] has the form:

$$x(t+1) = \hat{A}x(t) + \hat{B}u(t) + \hat{E}d(t) + \tilde{A}x(t) + \tilde{B}u(t) + \tilde{E}d(t) + e(t) \tag{2.44}$$

where $x \in \mathbb{R}^n$, $u \in \mathcal{U} \subset \mathbb{R}^m$ is the control constrained command, $d \in \mathcal{D} \subset \mathbb{R}^l$ is the exogenous measured disturbance, $\hat{A}, \hat{B}, \hat{E}$ represent the nominal model matrices, $\tilde{A}, \tilde{B}, \tilde{E}$ are the matrices related to the multiplicative uncertainty, $e \in \mathbb{R}^n$ is additive uncertainty, and $\mathcal{D}$ and $\mathcal{U}$ are polytopic sets bounded and known a priori. It is assumed in [92] that

the uncertain model and the uncertainty bounds have an affine structure as a function of a parameter π, e.g., $\hat{A} = \hat{\mathbf{A}}(\pi)$, where $\hat{\mathbf{A}}$ is an affine mapping in π (similarly for $\hat{B}$ and $\hat{E}$). The parameter π is selected based on linear inequalities on π that are solved based on data measurements obtained from the system for different operating conditions. Also, this work proposes two algorithms to build a polytope control RPI set iteratively while selecting the optimal uncertain model based on the available system data.

Another data-driven approach is presented in [93], where an active learning technique is used to build an RPI set by selecting the most informative sample(s) from a pool of unlabeled samples based on its current knowledge of the learning problem, thereby restricting the training set size without degrading performance. The algorithm presented in [93] uses information directly from a discrete nonlinear system (considering the sampling rate at which the data is available) without using a model like (2.38), i.e., it can be applied just from data information without having an actual model of the system. Other works that have explored the implementation of machine learning to obtain RPI sets can be found in [94, 95] and references therein.

2.1.1.3.2 Lyapunov-based

In this section ellipsoidal invariant sets for LTV systems are explored. First, recall that the positive invariant set for a linear system like (2.38) can be expressed as the quadratic Lyapunov inequality:

$$\Phi^\top P \Phi - P < 0$$

In [96] the idea of expressing the RPI as a set of quadratic Lyapunov inequalities is used, this is:

$$\Phi_i^\top P \Phi_i - P < 0, i = 1, \ldots, L \tag{2.45}$$

This set of conditions is nice because it is convex in P, which means that given a P_1 and P_2 that satisfy the inequalities, thus the convex combination of P_1 and P_2 also does.

Besides using the quadratic common Lyapunov inequalities as shown in (2.45), piecewise Lyapunov function can also be involved in RPI, which is introduced in [97]. The piece-wise Lyapunov functions have the form:

$$V(x) = \max\{x^T H_1 x, x^T H_2 x\}, \quad H_1 > 0, H_2 > 0 \tag{2.46}$$

and

$$V(x) = \min\{x^T H_1 x, x^T H_2 x\}, \quad H_1 > 0, H_2 > 0 \tag{2.47}$$

If H_1 and H_2 satisfy the following inequalities:

$$\Phi_1^T H_1 \Phi_1 - H_1 < 0, \quad \Phi_2^T H_2 \Phi_2 - H_2 < 0$$

$$(1-\delta_2)(\Phi_1^T H_2 \Phi_1 - H_2) + \delta_2(H_2 - H_1) < 0$$

$$(1-\delta_1)(\Phi_2^T H_1 \Phi_2 - H_1) - \delta_1(H_2 - H_1) < 0$$

for $\delta_1, \delta_2 \in [0, 1]$. Then, the function $V(x) = \max\{x^T H_1 x, x^T H_2 x\}$ is a valid Lyapunov function to build RPI.

Similarly, if H_1 and H_2 satisfy the following inequalities:

$$\Phi_1^T H_1 \Phi_1 - H_1 < 0, \quad \Phi_2^T H_2 \Phi_2 - H_2 < 0$$

$$(1-\delta_2)(\Phi_1^T H_2 \Phi_1 - H_2) - \delta_2(H_2 - H_1) < 0$$

$$(1-\delta_1)(\Phi_2^T H_1 \Phi_2 - H_1) + \delta_1(H_2 - H_1) < 0$$

Then, $V(x) = \min\{x^T H_1 x, x^T H_2 x\}$ is also a valid Lyapunov function to build RPI.

The differences between the piecewise Lyapunov function and the quadratic Lyapunov function are that the first can only be used when $L = 2$, but it offers a better estimation of RPI than that of the second.

Other types of Lyapunov functions have been involved in RPI such as polynomial functions [98], Set-Induced Lyapunov Functions [99], and parameter dependent Lyapunov Functions [100].

2.1.1.3.3 LMI methods

Other approaches to build Robust Positive Invariant (RPI) sets explore the use of Linear Matrix Inequalities (LMI). The work presented in [101] proposes an algorithm to obtain an RPI for system (2.38). The main idea is to obtain an invariant set given an initial set Ω_0, which is defined as:

$$\Omega_0 = \{x \in \mathbb{R}^n : F_0 x \leq \mathbf{1}\} \tag{2.48}$$

where $F_0 \in \mathbb{R}^{l \times n}$. From here the idea is to expand the given set Ω_0 by finding $\lambda = [\lambda_1, \ldots, \lambda_l]$, where $\lambda_i > 0, \forall i = 1, \ldots, l$. The resulting RPI has the H-form (halfspace form):

$$\Omega = \{x \in \mathbb{R}^n : F_0 x \leq \lambda\} \tag{2.49}$$

In order to get the optimal λ, first, L (i.e., number of uncertainties) Semi-Definite Programs (SDPs) need to be solved and then a single Linear Program (LP) is used to get the final vector λ. An important aspect to correctly implement the algorithm presented in [101] is that Ω_0 has to be invariant.

Another application of linear matrix inequalities is presented in [102] which combines and improves two complimentary algorithms to improve constraint handling of MPC for LPV systems. The first algorithm, from [103], constructs linear robustly stabilizing feedback controllers ($u = -Fx$) to ensure that the given initial state $\bar{x} \in \mathbb{R}^{n_x}$ lies within a feasible

invariant ellipsoid $\mathcal{E}$ of the form

$$\mathcal{E} \equiv \{x | x^\top P x \leq \gamma\}, \tag{2.50}$$

with $\gamma > 0$ and where $P = P^\top \succ 0, P \in \mathbb{R}^{n_x \times n_x}$ satisfying the Lyapunov inequality

$$\begin{aligned} P - (A_i - B_i F)^\top P (A_i - B_i F) > & Q + F^\top R K \\ & i = 1, \ldots, p \end{aligned} \tag{2.51}$$

where $Q = Q^\top > 0$ and $R = R^\top > 0$ are respectively the state and input cost matrices. Additionally, the matrices A_i, B_i, represent the vertices of the uncertainty polytope, with $p \in \mathbb{Z}_+$ denoting the total number of vertices. The above problem is solved via the following optimization problem:

$$\min_{\gamma, Y, Z=Z^T>0} \gamma,$$

subject to

$$\begin{aligned}
\begin{bmatrix} 1 & * \\ \bar{x} & Z \end{bmatrix} &> 0, \\
\begin{bmatrix} Z & * & * & * \\ Q^{\frac{1}{2}}Z & \gamma I & * & * \\ R^{\frac{1}{2}}Y & 0 & \gamma I & * \\ AZ+BY & 0 & 0 & Z \end{bmatrix} &> 0, \\
\begin{bmatrix} Z & * \\ (A_u)_{[j,:]}Y & 1 \end{bmatrix} &> 0, \quad j = 1, ..., m_u \\
\begin{bmatrix} Z & * \\ (A_x)_{[j,:]}Z & 1 \end{bmatrix} &> 0, \quad j = 1, ..., m_x
\end{aligned} \tag{2.52}$$

where m_u and m_x denote the number of rows in matrices A_u and A_x respectively and asterisks are used to denote the corresponding transpose of the lower block part of symmetric matrices. The optimal solutions to this problem are denoted $\gamma^\circ, Y^\circ, Z^\circ$, and the feedback matrix F becomes

$$F = -Y^\circ (Z^\circ)^{-1}, \tag{2.53}$$

the Lyapunov function $V(x) = x^T P x$ can be solved with

$$P = \gamma^\circ (Z^\circ)^{-1}, \tag{2.54}$$

and the RPI set is described by

$$\mathcal{E} = \{x | x^T (Z^\circ)^{-1} x \leq 1\}. \tag{2.55}$$

Furthermore, LMI (2.52) imposes that the invariant ellipsoid $\mathcal{E}$ should lie within the state constraints. The second algorithm that [102] utilizes to build RPI sets is first presented in both [80] and [104]. This algorithm constructs a polyhedral RPI set, denoted $\mathcal{P}$, for LPV systems. The main contribution of [102] is the introduction of polyhedral RPI sets (from [80] and [104]) in the controller synthesis method of [103] to improve constraint handling and create a more optimal controller. This result can be applied either off-line to compute robustly stabilizing linear feedback controllers with guaranteed feasibility, or can be applied on-line in a receding horizon fashion.

Next, we list other governor schemes that may not fit within the categories presented previously.

2.1.1.4 Other Governor Schemes

The following sections present a brief description of other reference governors schemes.

2.1.1.4.1 Reduced Order RG

This was part of Kalabic's Ph.D. thesis [66]. The main idea of this scheme is to decompose the system in slow and fast subsystems, as presented below where the subscripts s and f represent slow and fast respectively, this is:

$$\begin{aligned} x_s(t+1) &= A_s x_s(t) + B_s v(t) \\ x_f(t+1) &= A_f x_f(t) + B_f v(t) \\ y(t) &= C_s x_s(t) + C_f x_f(t) + D v(t) \end{aligned} \tag{2.56}$$

Note from (2.56) that the input $v(t)$ is considered for both subsystem. The reduced order governor enforces the constraint on the reduced system treating the contribution of the fast system as a bounded disturbance. From [105], it is implied that if the reduced order model can be made second order, an efficient and fast implementation of the RG computations is possible.

2.1.1.4.2 Network RG

This scheme applies to cases when there is a communication between plant and the governor in a synchronous or asynchronous fashion. In those cases, the Network RG accounts for the time delays in the communication between the plant and the scheme in order to enforce the constraints. The main references are [46, 106–112].

2.1.1.4.3 Virtual State Governor (VSG)

VSG is a modular control system design that aims at integrating multiple actuators, each equipped with an assigned non-modifiable feedback control law, while enforcing constraints and minimizing the use the actuators, which are expensive to operate [113]. Given a constrained plant with a group of actuators, this scheme generates virtual states $x_i \in \mathbb{R}^n$ and defines a nonlinear mapping function $g(x)$ that is assigned to the controllers in place for the actual system state x. This scheme modulates the effects of the controllers by modifying the state from which the feedback is computed. The virtual states are obtained by decomposing the actual state in a way that minimizes the usage of the expensive actuator while ensuring constraint satisfaction. A quadratic program (QP) is implemented to perform the decomposition of the actual state and is based on a MAS and a Lyapunov function of the loop involving the expensive actuator. This scheme have been used for engine control, energy management in HEV and aerospace.

2.1.1.4.4 Governors and Fault Tolerance

After the occurrence of a fault (e.g. loss of an actuator) if a system is not able to achieve the same nominal performance, then it may not be enough to reconfigure only the feedback control law, but also the control objectives (i.e., references) should be modified [114–116]. This is the idea behind the governors and fault tolerance scheme, which optimizes a design parameter $\theta(t)$ that can be used to add offset to the nominal system after a failure in order to counteract some of the offsets of the fault.

2.1.1.4.5 Decentralized CG and Distributed CG for Large Scale and Multi-Agent Systems

This scheme is applied to systems consisting of N dynamically coupled subsystems which are subject to local and global constraints. The initial solutions make use of the feedforward command governor approach developed in [117], which can reformulate the decentralized reference management problem as a static problem of determining, at each decision step, the local references $v_i(t), i = 1, ..., N$, such that the aggregated vector $v(t) = (v_1(t), ..., v_N(t))$ is admissible. Also, the variation of $v(t)$ within two update times is constrained in the set of admissible variations ΔV, i.e., $v(t+1)-v(t) \in \Delta V$. Sequential approaches of this constraint management scheme [118] allow only one agent at a time to modify its command. Parallel approaches are schemes where all the agents are allowed to move the command at the same time and each agent makes a local decision assuming that the other agents will make the worst choice.

Casavola proposed in [119] the Distributed CG, which consists of the design of distributed supervision strategies based on multi-agent CG ideas for networked interconnected systems in situations where the use of a centralized coordination unit is impracticable. The main idea of this scheme is to allow agents, that are not jointly involved in any coupling constraint, to simultaneously update their control actions without violating constraints.

2.1.1.4.6 Controller State and RG (CSRG)

This scheme proposes a RG capable of resetting the internal closed-loop system states as necessary to avoid constraint violation [120]. Thanks to internal states resetting, CSRG enlarges the constraint admissible domain of attraction and improves performance. In [120], two reset controller states are studied: controller state governors for constrained stabilization problems; and controller state and reference.

2.1.1.4.7 RG for Switching Systems

[121] and [122] propose a method that relies in its capability to avoid constraint violation and loss of stability regardless of any configuration change in the plant/constraint structure by commuting the system configuration (model plant + CG) with a more adequate one. To this end, the concept of model transition dwell time is used within the proposed control framework to formally define the minimum time necessary to enable a switching event under guaranteed conditions on the overall stability and constraint fulfillment.

Another work proposed by Taguchi [123] combines RG with controller switching. Another approach was proposed by Kolmanovsky [76].

2.1.1.4.8 Prioritized RG

Prioritized RG was proposed by Kalabic and Kolmanovsky in [124]. Two methods were proposed to prioritized constraint enforcement and prioritized reference tracking. The first method uses a CG-based formulation and relaxes the constraints through the use of quadratic penalty functions on slack variables. This is done in order to weight constraint infringement against desired reference set-points, and hence achieve a balance between tracking performance and constraint enforcement. 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.

2.1.1.4.9 Feed-Forward CG

This scheme was proposed by Garone, Tedesco and Casavola [117, 125], the feed-forward CG is a strategy for input and state-related constrained discrete-time LTI systems subject to bounded disturbances in the absence of explicit state or output measurements. The idea behind such an approach is that, if sufficiently slow (which makes it very conservative) and smooth transitions in the reference modifications are acted on the CG unit, the state evolutions remain not too far from the space of feasible steady states because of the asymptotic stability of the system.

2.1.1.4.10 RG and Learning Methods

An example of this type of schemes is presented in [51], which proposes a formulation that exploits Lyapunov-based methods, through the implementation of ERG, plus learning algorithms. Another technique that uses machine learning methods to learn the MAS based on data is presented in [126].

In [127] a new RG-based scheme is proposed with an Iterative Learning Control (ILC) for tracking performance improvement of a system without an exact plant model.

As studied so far, the core of RG implementation is the construction of a MAS's. For nonlinear systems or systems with uncertainty, the computation of an admissible set is not an easy task, hence works like the one presented in [128] study the computation of stability regions for constrained nonlinear systems using support-vector machine learning. A similar method is presented in [129], where a learning scheme is used to obtain a constraint admissible region.

Table 2.1 summarizes a comparison between the different governors schemes that have been presented in this overview. Table 2.2 presents some recent works about RG and CG in the different industrial fields.

Table 2.1: Comparative table between different governor schemes

- A: Formulation based on a LTI model.
- B: Formulation based on a nonlinear model.
- C: No computationally expensive.
- D: Non conservative.
- E: Robust to exogenous disturbances.
- F: Robust to plant/model mismatch.

Schemes	A	B	C	D	E	F
RG	✓	✗	✓	✓	✓	✗
VRG	✓	✗	✓	✓	✓	✗
CG	✓	✗	✓	✓	✓	✗
ECG	✓	✗	✓	✗	✓	✓
Robust RG	✗	✓	✗	✓	✓	✗
ERG	✓	✓	✓	✗	✓	✓
RG Based on LTI for Nonlinear Plants	✓	✗	✓	✓	✓	✗
Parameter RG	✗	✓	✗	✓	✗	✓
Output Feedback RG	✗	✓	✓	✗	✓	✓
Incremental RG	✗	✓	✗	✓	✓	✓
RG/CG for LTV	✗	✗	✗	✓	✓	✓
RG for Switching Systems	✓	✗	✓	✓	✓	✓
Reduced Order RG	✓	✗	✓	✓	✓	✗
Network RG	✓	✗	✓	✓	✓	✗
Virtual State Governor	✓	✗	✓	✓	✗	✗
Governors and Fault Tolerance	✓	✗	✓	✓	✓	✗
Decentralized and Distributed CG	✓	✗	✗	✓	✓	✗
CSRG	✓	✗	✓	✓	✓	✗
Prioritized RG	✓	✗	✓	✓	✓	✗
Feed-Forward CG	✓	✗	✓	✓	✓	✗
RG and Learning Techniques	✓	✓	✗	✗	✗	✓

Table 2.2: Applications of RG and CG

Field	Application	References
Automotive	Turbocharged gasoline engines	[42, 66, 130, 131]
	Diesel engines	[43, 132–134]
	Other engine applications	[135–137]
	Fuel cell	[35, 72, 138, 139]
	Chassis and dynamics control	[67, 140, 141]
Aerospace	–	[10, 142–149]
Power Networks	–	[115, 150–153]
Precision Mechatronics Manufacturing	–	[67, 141, 154] [155, 156] [26, 157]

2.1.2 Model Predictive Control

Model predictive control (MPC) is a scheme that tackles the problems of tracking performance and constraint satisfaction for SISO or MIMO systems. MPC has its origins in optimal control and its basic concept is to use a dynamic model to forecast system behavior and optimize the forecast to produce the best decision. An attractive feature of MPC is that it allows the design of multivariable feedback controllers with similar procedural complexity as single variable ones. Also, MPC allows for the specification in the design phase of constraints on system inputs, states, and outputs, which are then enforced by the controller. At the beginning of the 21^{st} century several books were published about MPC [27,158–161].

MPC captured the attention of control engineers for industrial applications thanks to a set of papers at the end of 1970s. These publications were related to petrol-chemical applications [162–164]. Dynamic Matrix Control (DMC) is outlined in [163], which is an important basis of MPC theory. Other important works on MPC were published in the late 1980s in [1, 165, 166]. In [165, 166] the first exposition of Generalized Predictive Control (GPC) was introduced. MPC has been widely applied in petro-chemical and related industries where satisfaction of constraints is particularly important because of efficiency

demands.

Fig. 2.2 shows the main idea of MPC. Reference [1] is a survey paper about MPC theory and implementation, which discusses the MPC algorithm formulation, the implications of DMC, as well as Model Algorithmic Control (MAC).

C. E. García *et al.*

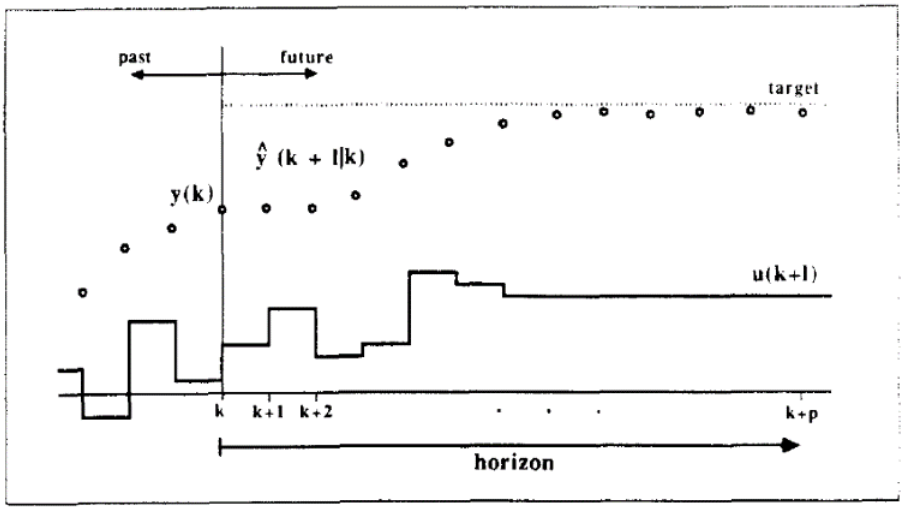

FIG. 1. The "moving horizon" approach of Model Predictive Control.

Figure 2.2: The moving horizon MPC [1]

From [167] the general formulation of MPC is presented in (2.57), where t is the discrete time index and for a vector, v, the notation $v(k|t)$ denotes the value of v predicted k steps ahead of t, based on information up to t. The optimizer of (2.57) is the control input sequence $U(t) = u(0|t), ..., u(N-1|t))$. The objective function is defined based on the terminal cost F and the cost function L. The prediction horizon is defined by N and the control horizon is defined with N_u. Normally $N > N_u$, the infinite horizon is defined when $N = \infty$, however for practical reasons a finite long enough prediction and control horizons are defined.

$$
\begin{aligned}
&\underset{U}{\text{minimize}} \quad F(x(N|t)) + \sum_{k=0}^{N-1} L(x(k|t), y(k|t), u(k|t)) \\
&\text{s.t.} \quad x(k+1|t) = f(x(k|t), u(k|t)), \\
&y(k|t) = h(x(k|t), u(k|t)), \\
&x_{min} \leq x(k|t) \leq x_{max}, \quad k = 1, ..., N \\
&y_{min} \leq y(k|t) \leq y_{max}, \quad k = 1, ..., N \\
&u_{min} \leq u(k|t) \leq u_{max}, \quad k = 1, ..., N_u \\
&x(0|t) = x(t), \\
&u(k|t) = \kappa(x(k|t)), \quad k = N_u, ..., N-1
\end{aligned}
\tag{2.57}
$$

There are several MPC formulations depending on the model type (i.e., LTI, LTV, nonlinear, unmeasured uncertainties, among others.) and based on the definition of the objective function. A common formulation of the cost function is presented in (2.58):

$$
\|x(N|t)\|_P + \sum_{k=0}^{N-1} \|x(k|t)\|_Q + \|u(k|t)\|_R + \|y(N|t)\|_S \tag{2.58}
$$

where Q, R, S, and P are weight matrices (very often diagonal matrices). Also, when (2.58) is applied to LTI systems, we can categorize it as linear MPC.

2.1.2.1 Differences Between RG and MPC

RG can be categorized as a set-theoretic predictive control technique for constraint management. Hence, it is expected to have similarities and differences with MPC. These are studied in this section.

In Section 2.1.1.1.1, it was explained how a MAS is built by leveraging assumption

A.2.1.2 and the predictions of the closed-loop linear system (2.1). In the case of MPC, predictions are used to compute an optimal control command to satisfy both tracking and constraint enforcement. However, in this context, instead of a MAS, the concept of maximal invariant constraint admissible set, which we denote as O_{∞}^{M}, is introduced. This one has a different meaning than O_{∞} and arises when closed-loop stability is enforced with MPC. To better explain this consider the discrete-time LTI plant:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.59}$$

where $x \in \mathbb{R}^n$ is the state, $y \in \mathbb{R}^p$ is tracking output, and $u \in \mathbb{R}^m$ is the control command. For (2.59), we consider, first, the unconstrained problem. The following performance index is defined:

$$J(z, x) = x(N|t)^{\top} P x(N|t) + \sum_{k=0}^{N-1} x(k|t)^{\top} Q x(k|t) + u(k|t)^{\top} R u(k|t) \tag{2.60}$$

where $z = [u(0|t), ..., u(N-1|t))]^{\top}$, $Q = Q^{\top} > 0$, $P = P^{\top} > 0$, and $R = R^{\top} > 0$. The goal is to find the sequence z^* that minimizes (2.60).

From (2.59) we can define the input to state relation as:

$$x(t+k|t) = A^k x(t) + \sum_{j=0}^{k-1} A^j B u(k-1-j|t) \tag{2.61}$$

We can write (2.59) and (2.61) in the matrix form as:

$$\begin{aligned} J(z,x) =& x(0|t)^\top Q x(0|t) + \begin{bmatrix} x(1|t) \\ \vdots \\ x(N-1|t) \\ x(N|t) \end{bmatrix}^\top \overbrace{\begin{bmatrix} Q & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & Q & 0 \\ 0 & \dots & 0 & P \end{bmatrix}}^{\bar{Q}} \begin{bmatrix} x(1|t) \\ \vdots \\ x(N-1|t) \\ x(N|t) \end{bmatrix} \\ &+ \begin{bmatrix} u(0|t) \\ \vdots \\ u(N-2|t) \\ u(N-1|t) \end{bmatrix}^\top \underbrace{\begin{bmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & R \end{bmatrix}}_{\bar{R}} \begin{bmatrix} u(0|t) \\ \vdots \\ u(N-2|t) \\ u(N-1|t) \end{bmatrix} \end{aligned} \tag{2.62}$$

$$\begin{bmatrix} x(1|t) \\ x(2|t) \\ \vdots \\ x(N|t) \end{bmatrix} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\bar{T}} x(t) + \underbrace{\begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}}_{\bar{S}} \underbrace{\begin{bmatrix} u(0|t) \\ \vdots \\ u(N-2|t) \\ u(N-1|t) \end{bmatrix}}_{z} \tag{2.63}$$

where $x(t)$ represents the current time t state value. From (2.62) and (2.63), we can rewrite (2.60) as follows:

$$J(z,x) = \frac{1}{2} z^\top \underbrace{(\bar{R} + \bar{S}^\top \bar{Q} \bar{S})}_{H} z + x(t)^\top \underbrace{2\bar{T}^\top \bar{Q} \bar{S}}_{F^\top} z + \frac{1}{2} x(t) \underbrace{2(Q + \bar{T}^\top \bar{Q} \bar{T})}_{L} x(t) \tag{2.64}$$

Now, by noticing that the L part of (2.64) can be disregarded by the optimization problem, we get:

$$\underset{z}{\text{minimize}} \quad f(z) = \frac{1}{2} z^\top H z + x(t)^\top F^\top z \tag{2.65}$$

Recall that $x(t)$ is the current initial condition that is assumed to be available. Finally, the solution to (2.65) is given by:

$$\begin{aligned} &\nabla f(z) = Hz + Fx(t) = 0 \implies z^* = -H^{-1}Fx(t) \\ &\implies u(t) = -\begin{bmatrix} I_m & 0 & \dots 0 \end{bmatrix} H^{-1}Fx(t) = Kx(t) \end{aligned} \tag{2.66}$$

From (2.66) it is possible to imply that an unconstrained linear MPC problem is just a linear state-feedback problem.

Now, if we consider polytopic constraints on the control command and state as $u \in U$ and $x \in X$ respectively. Problem (2.65) can be formulated as in (2.57), this is:

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & \frac{1}{2} z^\top H z + x(t)^\top F^\top z \\ \text{s.t.} \quad & Gz \leq W \end{aligned} \tag{2.67}$$

where G and W are computed similarly to the matrices in (2.62) and (2.63) and based on the constraint values for the state and input, as well as matrices $\bar{S}$, $\bar{T}$, and initial condition $x(t)$. Now, (2.67) is a standard quadratic programming problem with Nm optimization variables (i.e., $z \in \mathbb{R}^{Nm}$). Fig. 2.3 shows a graphical representation of how a solution to (2.67) is found by solving a QP problem.

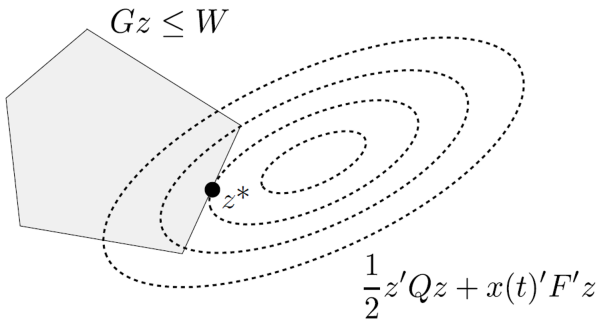

Figure 2.3: Admissible set and cost function curves.

After solving (2.67), the input $u(0|t)$ is applied and the remaining optimal values are ignored. By doing this, it is possible to have feedback at each time step through $x(t)$.

Implementing MPC by solving problem (2.67) for a stable plant (2.59) may suffice to have a closed-loop stable system. However, in the case of regularizing an unstable plant (2.59) with the same MPC, there are no guarantees of stability. In such a case, the idea of terminal constraint and terminal invariant set are introduced [161]. Problem (2.67) is modified to:

$$\begin{aligned} &\underset{z}{\text{minimize}} \quad \frac{1}{2} z^\top H z + x(t)^\top F^\top z \\ &\text{s.t.} \quad Gz \leq W \\ &\quad x(N|t) \in \mathcal{X}_f \end{aligned} \tag{2.68}$$

where $\mathcal{X}_f$ is an invariant terminal set under the control law $u(t) = K_f x(t)$ that satisfies:

$$\begin{aligned} &x(t+1) = (A + BK_f)x(t) \in \mathcal{X}_f, \forall x \in \mathcal{X}_f \\ &\mathcal{X}_f \subset X, K_f x \in U, \forall x \in \mathcal{X}_f \end{aligned} \tag{2.69}$$

where K_f is a LQR controller gain. The terminal constraint can be defined as presented in (2.68) or can be defined as $x(N|t) = 0$. The difference between the two methods is that the latter leads to a smaller domain of attraction [168]. A common choice for $\mathcal{X}_f$ is the maximal invariant constraint admissible set, O_∞^M, for a controlled system. The set O_∞^M for system (2.59) under the control law $u(t) = K_f x(t)$ is given by:

$$O_\infty^M := \left\{ x | \bar{A}^k x \in X, K_f \bar{A}^k x \in U \, \forall k \geq 0 \right\} \tag{2.70}$$

where $\bar{A} = A + BK_f$. All input and state constraints are satisfied for the closed-loop system using the LQR control law for $x \in O_\infty^M$. In this case the terminal cost function, i.e., $x(N|t)^\top P x(N|t)$ given in (2.60), can be designed to be a continuous Lyapunov function in the terminal set by ensuring that P is the solution to the corresponding Lyapunov equation.

This ensures that (2.68) enforces closed-loop stability [161].

To summarize, note that the definition of O_∞^M is similar to the definition of O_∞ presented in Section 2.1.1.1.1, with the main difference on how the input command is assumed to be computed. However, their applicability is conceptually different. The former is used in MPC to compute a maximal invariant terminal set that helps to enforce stability of the closed-loop system, while in RG, O_∞ is used for constraint enforcement only. In RG, closed-loop stability is enforced by noticing from (2.17) that for a constant $r(t)$, $v(t)$ forms a monotonic sequence over a compact set, which implies that $v(t)$ must converge to a constant [59]. In MPC, the prediction horizon, N, is a design parameter which must be carefully selected to avoid excessive computational burden. This problem can be solved by Explicit MPC [169], however, for high order systems the computational load may be significant. This is not a problem for RG, since O_∞ is computed offline, and even in the presence of high order systems, there are techniques in the literature that allow further reductions of the number of rows of O_∞ to minimize memory and computational burden [17]. Even though MPC can use different prediction and control horizons to reduce the number optimization variables to consider (as suggested by (2.57)), RG solves just for one optimization variable at each time step, which drastically reduces computation load.

In terms of the practical considerations for the implementation of MPC vs. RG. The former may be used in the cases when tracking and constraint enforcement are required, and when there is zero or no significant effort to replace legacy controllers. The latter, since it is an add-on mechanism, does not require to change legacy controllers and the implementation and calibration effort is significantly less, for example in automotive applications.

Next, we briefly study some of the different MPC schemes that have been proposed in the literature.

2.1.2.2 Implicit MPC

Implicit MPC is a form of control in which the control action is obtained by solving online, at each sampling instant, a finite horizon optimal control problem in which the initial state is the current state of the plant. The challenge of MPC on embedded hardware is to obtain the optimal solution while taking into account limited resources of the implementation hardware. Traditionally, the optimal solution is obtained either by an iterative numerical procedure (referred to as implicit MPC), or by evaluating the explicit representation of the MPC feedback law, which is obtained off-line using parametric programming (referred to as explicit MPC). Both approaches have their pros and cons. Implicit MPC requires more computational resources, but it is able to handle large systems. Explicit MPC, on the other hand, requires less online computation, but the off-line construction of the feedback law scales badly with increasing dimensionality of the problem. Moreover, the memory footprint of the explicit solutions can easily violate limits of the available memory storage. Implicit MPC is implemented as explained in the previous section.

2.1.2.3 Hybrid and Time Varying Models

LTI dynamics can be substituted by piecewise affine (PWA) dynamics as presented in (2.71), this is:

$$\begin{aligned} x(k+1|t) &= A_{i(k|t)}x(k|t) + B_{i(k|t)}u(k|t) + \phi_{i(k|t)} \\ y(t) &= C_{i(k|t)}x(k|t) + D_{i(k|t)}u(k|t) + \zeta_{i(k|t)} \\ i(t) &: H^x_{i(k|t)}x(k|t) + H^u_{i(k|t)}u(k|t) \leq K^x_{i(k|t)} \\ i(t) &\in \{1, ..., s\} \end{aligned} \tag{2.71}$$

It is possible to implement the objective function (2.58) in the system (2.71) for the case when the one-norm or infinity-norm is used, in such case the problem becomes a Mixed Integer Linear Program. If the 2-norm is used, then the problem becomes a Mixed

Integer Quadratic Program [170].

For a time varying system, (2.58) can not be used since the LTV system is changing at each step. Therefore, different formulations should be applied.

Other works related with hybrid systems as PWA, continuous and discrete system, discontinuous systems are in the literature [171–174]

2.1.2.4 Distributed MPC (DMPC)

For multi-agent systems where a centralized controller is not implementable, DMPC can be used to ease the control and state constraints in MIMO systems. It is easier to implement than centralized MPC if it is possible to decompose the original control problem into a set of smaller problems of controlling a set of subsystems of the original system, each subsystem having its own controller (or agent) and, possibly, its own objective. Further work in the area are found in [175–178]

2.1.2.5 Embedded MPC

MPC implementation is normally used for systems with "slow" dynamics. However, some researchers have considered the development of MPC models to control small but fast dynamic systems using an embedded system, (i.e., a computer system, usually a micro-controller or microprocessor, with a dedicated function). Bemporad developed a method for ultra-fast MPC [179]. Other works using MCU and FPGA are in [180, 181]

2.1.2.6 Robust MPC

This scheme considers the case when the decision variable is also a control sequence, but the disturbance is taken into account in the optimal control problem solved online. The state and control constraints, and the terminal constraint if employed, are required to be satisfied for all possible disturbance sequences. The objective function is defined as:

$$\underset{w}{\text{maximize}} \quad \{F(x^{u,w}(N|t)) + \sum_{k=0}^{N-1} L(x^{u,w}(k|t), y(k|t), u(k|t))|w \in W\} \tag{2.72}$$

where w is the disturbance sequence. Works using this formulation are [182, 183]. An extension of robust MPC to nonlinear systems is presented in [184].

2.1.2.7 Stochastic MPC

For robust MPC, the disturbances were assumed to be bounded and constraints were required to be satisfied for all possible realizations of the disturbance process. In stochastic MPC, on the other hand, disturbances are assumed to be stochastic and not necessarily bounded and (at least some) constraints are softened, i.e., not required to be satisfied for all realizations of the disturbance. Methods related to this problem are in [185–188]

2.1.2.8 Adaptive MPC

The problem of adaptive MPC has received very little attention. For this problem, it is assumed that the states are accessible and a parameter θ is defined, such that it lies in a known compact set Θ. The continuous nonlinear system has the form $\dot{x} = f(x, u) + g(x, u)\theta$. The adaptive control problem is more difficult if uncertainty in the form of additive disturbance and measurement noise are present. Works related to this topic are [189, 190]

2.1.2.9 Explicit MPC and Multiparametric Programming

One of the main reasons why MPC is categorized as a technique for systems with slow dynamics is because a solution to the problem (2.57) is computed online each time step. This is significantly computational expensive due to solvers programming that needs to be implemented to solve the optimization problem online. Explicit MPC proposes a solution to this problem by implementing multiparametric programming to pre-solve the objective

function offline. By doing so, the control law is computed by a continuous and piecewise-affine function [169]. In other words, given the optimization problem:

$$\begin{aligned} &\underset{z}{\text{minimize}} \quad h(z,x) \\ &\text{s.t.} \qquad g(z,x) \leq 0 \end{aligned} \tag{2.73}$$

the explicit MPC solves it offline for all $x(t) \in X$, where X is assumed a polytope of the form:

$$X = \{x \in \mathbb{R}^n : Sx \leq s\} \subset \mathbb{R}^n \tag{2.74}$$

The explicit MPC control law that explicitly defines the relation between the control law $u(x)$ and the state $x(t)$ is defined by

$$u(x) = \begin{cases} F_1 x + g_1 & if H_1 x \leq k_1 \\ \vdots & \vdots \\ F_M x + g_M & if H_M x \leq k_M \end{cases} \tag{2.75}$$

Solutions of (2.75) are obtained by implementing multiparametric programs [191, 192].

2.1.2.9.1 Mutiparametric Quadratic Programming

The main tool to solve explicit MPC problems is multiparametric QP (mpQP). The algorithms proposed in the literature to solve this kind of problems are based on Karush-Kuhn-Tucker (KKT) conditions for optimality. The general formulation for the mpQP is defined as [192]:

$$\begin{aligned} V^* = \frac{1}{2} x' Y x + \min_z \quad & \frac{1}{2} z' H z + z' F' z \\ \text{s.t.} \qquad & Gz \leq W + Sx \end{aligned} \tag{2.76}$$

Bemporad proves in [169] that the optimizer $z^* : X \mapsto \mathbb{R}^n$ is piecewise affine and continuous over the set X of parameters x for which the problem is feasible and the function $V^* : X \mapsto \mathbb{R}$ associating with every $x \in X$ the corresponding optimal (2.76) is continuous, convex, and piecewise quadratic. The paper [192] and reference within , presents a detailed explanation of multiparametric programming methods for explicit MPC.

2.1.3 Other Constraint Management Methods

2.1.3.1 Barrier Lyapunov Functions

A Barrier Lyapunov Function (BLF) is a scalar function $V(x)$, defined with respect to an autonomous system, $\dot{x} = f(x)$, on an open region $\mathcal{D}$, with the following properties [33]:

- Containing the origin.
- It is continuous.
- Positive definite.
- It has continuous first-order partial derivatives at every point of $\mathcal{D}$.
- It has the property $V(x) \to \infty$ as x approaches the boundary of $\mathcal{D}$.
- It satisfies $V(x(t)) \leq b \, \forall t \geq 0$ along the solution of $\dot{x} = f(x)$ for $x(0) \in \mathcal{D}$ and some positive constant b.

Constraint control is tackled by using BLFs that grow to infinity when its arguments approach some limits. Hence, by ensuring boundedness of the BLF in the closed loop, it is ensured that those limits are not transgressed. Several works related to nonlinear systems and BLF methods have shown significant results in the constraint management domain. The work presented in [33] shows the implementation of a symmetric and asymmetric BLF for a

SISO nonlinear system with constrained output. Also, [33] shows the implementation of an adaptive control scheme with BLF to handle parametric uncertainties. A similar approach is presented in [193] which proposes an adaptive controller with BLF for a pure-feedback nonlinear system with full state constraints. In [193] the design scheme combines BLF with dynamic surface control (DSC).

BLF implemented on nonlinear systems with time-varying output constraints is analyzed in [194], where an asymmetric time-varying BLF is implemented to handle constraint satisfaction and an adaptive controller is used to work with parametric model uncertainty. This combination helps to ensure constraint satisfaction during the transient phase of online parameter adaptation. A similar work for static constraints is developed in [195]. Other works related to BLF implementations with adaptive and robust control are [196–198] and references within.

2.1.3.2 Constraint Management and Machine Learning

Artificial intelligence techniques have been combined with BLF in order to work with model parametric uncertainties and constraint satisfaction. [199] proposes an adaptive neural network (NN) controller with BLF for the trajectory tracking of a marine surface vessel in the presence of constrained output and parametric uncertainties. The adaptive neural network approximates the unknown model parameters and with the BLF the complete control scheme avoid constraint violations. Reference within [199] provides more information about works using artificial intelligence for models with parametric uncertainties.

In [200] a Takagi-Sugeno Fuzzy logic controller with nonlinear local models is implemented for constraint satisfaction in the controller input and output; the results are verified on an inverted pendulum. The work proposed in [201] gathers neural networks with DSC in order to propose an adaptive NN control for a strict feedback nonlinear system. A similar

work is proposed in [202], where a NN is used with adaptive controller for tracking performance and a BLF is implemented for constraint satisfaction, also a disturbance observer is implemented for a nonlinear system.

Other works that explore learning techniques for constraint management are presented in [203–205], where algorithms are provided to learn parametric constraints based on demonstrations. Robotics applications are considered for this type of constraint learning methods.

Chapter 3

Stochastic Reference Governor

A stochastic RG formulation based on SR-MAS is presented in this chapter [54]. This one is based on a Stochastic Robustly invariant MAS (SR-MAS), which is built by leveraging the ideas presented in [24]. To accomplish this, first, several important structural properties of MAS and SR-MAS are proved, which, to the best of the author knowledge, have not been previously presented in the literature. These properties are used to extend the results in [24] to systems that are Lyapunov stable, which arise in RG applications, and systems with output constraints instead of state constraints. An algorithm to compute the SR-MAS is presented. Numerical simulations are presented to show that the stochastic RG can provide probabilistic guarantees of constraint satisfaction.

3.1 Structural Analysis of $\bar{O}_\infty$

An important element for the development of the SR-MAS, and hence stochastic RG, is the effect of $\bar{P}_0$ on the structure of $\bar{O}_\infty$, which we analyze in this section.

First, as we show in the following theorem, even with initial conditions outside of $\bar{O}_\infty$, the system states converge to $\bar{O}_\infty$ under the system's natural dynamics and, consequently constraints imposed over the output will be satisfied.

Theorem 3.1.1. *Assume (A.2.1.1) and (A.2.1.2) hold. If* $(x(0), v(0)) \notin \bar{O}_\infty$ *and* $v(0) \in \text{int}(\bar{V})$*, then there exists a time* $T \in \mathbb{Z}_+$ *such that* $\forall t \geq T, (x(t), v(t)) \in \bar{O}_\infty$*, which implies that* $y(t) \in Y$*.*

Proof. Definition (2.4), together with the condition $v(0) \in \text{int}(\bar{V})$, implies that the steady state value of $y(t)$, i.e., $y_{ss} := \lim_{t\to\infty} y(t)$, satisfies $y_{ss} \in \text{int}(\bar{Y})$. This, in turn, implies that there exists a ball around y_{ss} such that $\mathcal{B}(\alpha, y_{ss}) \subset \bar{Y}$ for some $\alpha > 0$. Finally, by convergence of $y(t) \to y_{ss}$, there exists a finite time $T \in \mathbb{Z}_+$, such that $\forall t \geq T, y(t) \in \mathcal{B}(\alpha, y_{ss})$, which implies that $y(t) \in \bar{Y}$, $\forall t \geq T$. Therefore, by definition of $\bar{O}_\infty$, we have

that $(x(t), v(t)) \in \bar{O}_\infty$. □

Next, to study the effects of the halfspace $\bar{P}_0$ on $\bar{O}_\infty$, we proceed to remove this halfspace from (2.13). The resulting set will be evaluated under the system's natural dynamics.

Theorem 3.1.2. *Assume (A.2.1.1) and (A.2.1.2) hold. Let* $\Omega := \bigcap_{i=1}^{t^*} \bar{P}_i$. *If* $(x(0), v(0)) \in \Omega \setminus \bar{P}_0$, *then,* $y(0) \notin Y$ *and* $(x(1), v(1)) \in \bar{O}_\infty$.

Proof. From the definition of $\bar{P}_0$ in (2.16), if $(x(0), v(0)) \in \Omega \setminus \bar{P}_0$, then $(x(0), v(0)) \notin \bar{P}_0$ and, thus, $y(0) \notin Y$. The second condition in the theorem follows by definitions of Ω above and $\bar{P}_i$ in (2.16). □

From Theorems 3.1.1 and 3.1.2, we note that by changing the structure of $\bar{O}_\infty$ and introducing a set like Ω, we may have constraints violation for a finite time. However, as time progresses, the constraints will eventually be satisfied (under the assumptions stated in Theorem 3.1.1). From this observation, it is possible to think about clever mechanisms to increase the size of MAS (i.e., to make it less conservative), for systems that can handle constraint violation for a finite time. This idea is exploited in the Section 3.2 to create the SR-MAS.

Remark 3.1.1. *Theorem 3.1.2 states that excluding* $\bar{P}_0$ *from* $\bar{O}_\infty$ *(i.e., when* $(x(0), v(0)) \in \Omega$*) may lead to constraint violation; however, the system recovers from violation after one timestep under its natural dynamics. This conclusion may not hold if assumption (A.2.1.2) is violated, i.e., if the input is not constant. As an example, consider a system with direct feedthrough between the input and the output (i.e.,* $D_v \neq 0$*) and a reference governor in the loop operating on* Ω *(instead of* $\bar{O}_\infty$*). In this case, the system may never recover from constraint violation because, with RG,* $v(t)$ *is not held constant for all times, and RG may actively keep the state in* $\Omega \setminus \bar{O}_\infty$ *for a sustained period of time. Note, however, that if* $D_v = 0$, *then* $\bar{P}_0$ *can be safely excluded because this halfspace does not contribute to the computation of the control signal* $v(t)$.

In this section we studied the structure of $\bar{O}_\infty$ under conditions that are not normally evaluated in the literature, that is, $(x(0), v(0)) \notin \bar{O}_\infty$. This condition showed importance of $\bar{P}_0$ and correspondingly Y_0.

3.2 Stochastic Robustly Invariant Maximal Admissible Sets (SR-MAS)

In section 2.1.1.1.1.2, we defined the set $\bar{O}_\infty$ based on the worst case scenario of the disturbance sequence $w(t) \in W$. This idea tends to build a large conservative margin over the constrained output, which is not always the best solution. Thus, we introduce in this section a SR-MAS based on chance constraints, with the aim of providing a less conservative solution for systems like (2.8).

In order to ease the exposition that follows, we first proceed to rewrite system (2.8) based on the assumption (A.2.1.2). For notational simplicity, we consider the case of one input, i.e., $m = 1$. The resulting augmented dynamics are:

$$\begin{aligned} \bar{x}(t+1) &= \bar{A}\bar{x}(t) + \bar{B}_w w(t) \\ y(t) &= \bar{C}\bar{x}(t) + D_w w(t) \end{aligned} \tag{3.1}$$

where $\bar{x}(t) = \begin{bmatrix} x(t) \\ v(t) \end{bmatrix}, \bar{A} = \begin{bmatrix} A & B_v \\ 0 & 1 \end{bmatrix}, \bar{C} = \begin{bmatrix} C & D_v \end{bmatrix}, \bar{B}_w = \begin{bmatrix} B_w \\ 0 \end{bmatrix}$. Considering system (3.1), we impose the probabilistic constraint

$$Pr(y(t) \in Y | \bar{x}_0) \geq 1 - \beta \tag{3.2}$$

for some $0 < \beta < 1$.

Expression (3.2) is known as a chance constraint. Based on this type of constraint and by adopting the ideas presented in [24], we proceed to define the SR-MAS for (3.1). First, expression (2.12) now becomes:

$$\begin{aligned} P_t^\beta := \{&\bar{x}_0 \in \mathbb{R}^{n+1} : \bar{x}(0) = \bar{x}_0, Pr(y(t) \in Y) \geq 1 - \beta, \\ &\forall \{w(j)\} \in W, j = 0, \ldots, t\} \end{aligned} \tag{3.3}$$

and the SR-MAS O_∞^β is defined as:

$$O_\infty^\beta := \bigcap_{t=0}^{\infty} P_t^\beta \tag{3.4}$$

Similar to the study developed in [24], the set O_∞^β is not positively invariant and it is not easy to compute. However, a positively invariant inner approximation, which is finitely determined, can be obtained, thanks to the introduction of the following set:

$$S_z(\beta) := \{\Gamma : Pr(z \in \Gamma) \geq 1 - \beta\} \tag{3.5}$$

where z is a vector of random variables. The set $S_z(\beta)$ is the collection of all sets with probability measure greater than $1 - \beta$.

Based on the set $S_z(\beta)$, the chance constraint (3.2) is redefined by noting that:

$$\begin{aligned} 1 - \beta &\leq Pr(y(t) \in Y | \bar{x}(0) = \bar{x}_0) \\ &= Pr(\bar{C}\bar{A}^t\bar{x}_0 + \bar{C}\sum_{j=0}^{t-1} \bar{A}^{t-j-1}\bar{B}_w w(j) \\ &\quad + D_w w(t) \in Y | \bar{x}_0) \\ &= Pr(\mathbf{w}_t \in \Gamma_t(\bar{x}_0)) \end{aligned} \tag{3.6}$$

where $\mathbf{w}_t := [w(0); ...; w(t)] \in W^{t+1}$ and $\Gamma_t(\bar{x}_0) := \{\mathbf{w}_t : \bar{C}\bar{A}^t\bar{x}_0 + \bar{C}\sum_{j=0}^{t-1} \bar{A}^{t-j-1}\bar{B}_w w(j) +$

$D_w w(t) \in Y\}$. Note that for a $\bar{x}_0$ that satisfies (3.6), $\Gamma_t(\bar{x}_0) \in S_{\mathbf{w}_t}(\beta)$. An alternative choice in $S_{\mathbf{w}_t}(\beta)$, which we use below, is:

$$Z_{t,\beta} := \underbrace{W \times \cdots \times W}_{t} \times W_\beta \tag{3.7}$$

where $W_\beta \subset W$ with the property $\int_{W_\beta} f_w(w) \geq 1 - \beta$ and, thus, $Z_{t,\beta} \in S_{\mathbf{w}_t}(\beta)$. We assume the following for the set W_β:

A. 3.2.1. *The set W_β is a compact polytope with the origin in its interior.*

Thanks to the introduction of the set $Z_{t,\beta}$, we can proceed with the approximation of P_t^β as:

$$\begin{aligned} \hat{P}_t^\beta &:= \{\bar{x}_0 : Z_{t,\beta} \subseteq \Gamma_t(\bar{x}_0)\} \\ &= \{\bar{x}_0 : \bar{x}(0) = \bar{x}_0, y(t) \in Y, \forall w(t) \in W_\beta, \\ &\forall w(j) \in W, j \in \mathbb{Z}_{t-1}\} \end{aligned} \tag{3.8}$$

and consequently the approximation of O_∞^β as:

$$\hat{O}_\infty^\beta := \cap_{t=0}^{\infty} \hat{P}_t^\beta \tag{3.9}$$

To show that $\hat{P}_t^\beta$ is indeed an inner approximation of P_t^β, note that if $\bar{x}_0 \in \hat{P}_t^\beta$, then, $Z_{t,\beta} \subseteq \Gamma_t(\bar{x}_0)$ by (3.8). This implies that $Pr(\mathbf{w}_t \in \Gamma_t(\bar{x}_0)) \geq Pr(\mathbf{w}_t \in Z_{t,\beta})$, which, by construction, is larger than $1 - \beta$. Therefore, $\bar{x}_0 \in P_t^\beta$, which implies that $\hat{P}_t^\beta \subseteq P_t^\beta$.

The set $\hat{O}_\infty^\beta$ is robustly positively invariant, as shown by the following theorem.

Theorem 3.2.1. *Assume $\hat{O}_\infty^\beta$ is non-empty. If $\bar{x}(t) \in \hat{O}_\infty^\beta$, then $\bar{x}(t+1) \in \hat{O}_\infty^\beta$ with respect to system (3.1), i.e., $\hat{O}_\infty^\beta$ is robustly positively invariant.*

Proof. We introduce the following notation to proceed with the proof:

1. Let $\bar{x}(k|t)$ and $\bar{x}(k-1|t+1)$ represent the same predicted state $\bar{x}(t+k)$ at time t and $t+1$ respectively, $k \geq 2, t \in \mathbb{Z}_+$

2. Let $y(k|t)$ and $y(k-1|t+1)$ represent the same predicted constrained output $y(t+k)$ at time t and $t+1$ respectively, $k \geq 1, t \in \mathbb{Z}_+$

3. Let $w(k|t)$ and $w(k-1|t+1)$ represent the same unrealized disturbance $w(t+k)$ at time t and $t+1$ respectively, $k \geq 1, t \in \mathbb{Z}_+$

Now, let $k \geq 1$. If $\bar{x}(t) \in \hat{O}_\infty^\beta$, then by (3.9), $\bar{x}(t) \in \hat{P}_k^\beta$ for the chosen k. We prove below that $\bar{x}(t+1) \in \hat{P}_{k-1}^\beta$ and, because k was arbitrary, it follows that $\bar{x}(t+1) \in \hat{O}_\infty^\beta$. Before we begin, to be consistent with the above notation, we restate the definition of $\hat{P}_k^\beta$ at time t and $\hat{P}_{k-1}^\beta$ at time $t+1$:

$$\begin{aligned}
\hat{P}_k^\beta =& \{\bar{x}_0 : \bar{x}(0|t) = \bar{x}_0, y(k|t) \in Y, \forall w(j|t) \in W, j \in \mathbb{Z}_{k-1}, \\
& \forall w(k|t) \in W_\beta\} \\
\hat{P}_{k-1}^\beta =& \{\bar{x}_0 : \bar{x}(0|t+1) = x_0, y(k-1|t+1) \in Y, \\
& \forall w(j|t+1) \in W, j \in \mathbb{Z}_{k-2}, \forall w(k-1|t+1) \in W_\beta\}
\end{aligned}$$

where t in (3.8) is replaced by the prediction time (k or $k-1$), and the initial time of 0 in (3.8) is replaced by the current time (t or $t+1$). Using these notations, we have:

$$
\begin{aligned}
&\bar{x}(t) \in \hat{P}_k^\beta \Rightarrow \bar{x}(0|t) \in \hat{P}_k^\beta \\
&\Rightarrow \bar{C}\bar{A}^k\bar{x}(0|t) + \bar{C}\sum_{j=0}^{k-1}\bar{A}^{k-j-1}\bar{B}_w w(j|t) + D_w w(k|t) \\
&\in Y, \forall w(j|t) \in W, j \in \mathbb{Z}_{k-1}, \forall w(k|t) \in W_\beta \\
&\Rightarrow \bar{C}\bar{A}^{k-1}(\bar{A}\bar{x}(0|t) + \bar{B}_w w(0|t)) + \bar{C}\sum_{j=1}^{k-1}\bar{A}^{k-j-1}\bar{B}_w w(j|t) \\
&+ D_w w(k|t) \in Y, \forall w(j|t) \in W, j \in \mathbb{Z}_{k-1}, \forall w(k|t) \in W_\beta \\
&\Rightarrow \bar{C}\bar{A}^{k-1}\bar{x}(0|t+1) + \bar{C}\sum_{j=0}^{k-2}\bar{A}^{k-j-2}\bar{B}_w w(j|t+1) \\
&+ D_w w(k-1|t+1) \in Y, \forall w(j|t+1) \in W, j \in \mathbb{Z}_{k-2}, \\
&\forall w(k-1|t+1) \in W_\beta
\end{aligned}
$$

Therefore, $\bar{x}(t+1) \in \hat{P}_{k-1}^\beta$, and, by definition of $\hat{O}_\infty^\beta$, $x(t+1) \in \hat{O}_\infty^\beta$. □

Remark 3.2.1. *The construction of $\hat{P}_t^\beta$ in* (3.8) *indicates that the most recent unrealized disturbance is assumed to belong to W_β, whereas all previous disturbances belong to W (i.e., worst-case disturbance). Therefore, while this approximation is more conservative than the actual SR-MAS (O_∞^β), it is less conservative than the standard approach, where the worst case disturbance is assumed at every time, including the most recent time. These ideas are illustrated with an example in Section VI.*

3.3 Computation of $\hat{O}_\infty^\beta$ and the Stochastic Reference Governor

3.3.1 Computation of $\hat{O}_\infty^\beta$

Algorithm 1 below summarizes the computation of $\hat{O}_\infty^\beta$. Note that we have redefined the sets in (2.10) for the computation of the SR-MAS. The issue of finite determinism of $\hat{O}_\infty^\beta$ is analyzed afterwards.

Algorithm 1 Compute $\hat{O}_\infty^\beta$

Let $Y_0 = Y \sim D_w W_\beta$
if $Y_0 = \emptyset$ **then**
  $\hat{O}_\infty^\beta = \emptyset$
  Stop
else
  $\hat{O}_0^\beta = \{\bar{x} : \bar{C}\bar{x} \in Y_0\}$
  $t = 0$
end if
while true **do**
  $Y_{t+1} = Y_t \sim \bar{C}\bar{A}^t\bar{B}_w W$
  if $Y_{t+1} = \emptyset$ **then**
    $\hat{O}_\infty^\beta = \emptyset$
    Stop
  else
    $\hat{O}_{t+1}^\beta = \hat{O}_t^\beta \cap \{\bar{x} : \bar{C}\bar{A}^{t+1}\bar{x} \in Y_{t+1}\}$
    if $\hat{O}_{t+1}^\beta = \hat{O}_t^\beta$ **then**
      $\hat{O}_\infty^\beta = \hat{O}_t^\beta$
      Stop
    end if
  end if
end while

Recall that $\sim$ denotes the P-subtraction, which can be implemented via linear programming. For more details, see [16].

Note, from Algorithm 1, that W_β enters the calculations only through the matrix D_w. Therefore, for systems with $D_w = 0$ (i.e., no feedthrough between the disturbance input and the constrained output), the SR-MAS is exactly the same as the standard MAS, which is not what we desire. This issue can be addressed by redefining the set $Z_{t,\beta}$ in (3.7) by using $t-1$ products instead of t. For this case, definition (3.8) becomes:

$$\begin{aligned}\hat{P}_t^\beta := \{&\bar{x}_0 : \bar{x}(0) = \bar{x}_0, y(t) \in Y, \forall w(t-1) \in W_\beta, \\ &\forall w(j) \in W, j \in \mathbb{Z}_{t-2}\},\end{aligned} \tag{3.10}$$

and (3.9) remains unchanged. These changes lead to the following modifications to Algorithm 1:

1. Set Y_0 to Y.

2. For the first iteration of the while loop, replace W with W_β.

The set $\hat{O}_\infty^\beta$ defined using these modifications is still robustly positively invariant. This is provable by following the same logic presented for the proof of Theorem 3.2.1.

Now consider again the general case (D_w not necessarily 0). As with standard MAS, the set $\hat{O}_\infty^\beta$ may not be finitely determined. However, an inner approximation, which is finitely determined, can be obtained by shrinking the steady state: $\bar{Y} := (1-\epsilon)Y_t$, for t large enough, where Y_t is now redefined by including W_β as follows: $Y_t := Y \sim D_w W_\beta \sim \bar{C}\bar{B}_w W \sim \cdots \sim \bar{C}\bar{A}^{t-1}\bar{B}_w W$.

The following theorem provides the conditions for finite determinism of $\hat{O}_\infty^\beta$.

Theorem 3.3.1. *Assume $\bar{Y}$ is non-empty, assumption (A.2.1.1) holds, the pair $(\bar{C}, \bar{A})$ is observable, and $0 \in \text{int}(\bar{Y})$. Then, the set $\hat{O}_\infty^\beta$ with respect to system (3.1) is finitely determined.*

Proof. The proof follows the same logic as the proof of Theorem 2.1.1. □

3.3.2 Stochastic Reference Governor

Finally, to implement a stochastic RG, we modify the standard RG formulation presented in Section 2.1.1.1.1.2, to employ the SR-MAS instead of MAS:

$$\begin{aligned} \underset{\kappa\in[0,1]}{\text{maximize}} \quad & \kappa \\ \text{s.t.} \quad & (x_0, v_0) \in \hat{O}_\infty^\beta \\ & v_0 = v(t-1) + \kappa(r(t) - v(t-1)) \\ & x_0 = x(t) \end{aligned} \tag{3.11}$$

Using the same logic as standard RG, it can be shown that this formulation is also stable and recursively feasible.

Remark 3.3.1. *For the case where $D_v = 0$, stochastic RG can exclude the halfspace $\hat{P}_0^\beta$ (i.e., the halfspace corresponding to the initial time can be removed from $\hat{O}_\infty^\beta$). This is consistent with Remark 3.1.1 for standard RG. Note that, for this case, even though $\hat{P}_0^\beta$ is removed, the calculation of the sets Y_t and $\bar{Y}$ still include the contribution from Y_0, for example, $Y_1 = Y \sim D_w W_\beta \sim \bar{C}\bar{B}_w W$.*

3.4 Stochastic Reference Governor: Numerical Simulation

In this section, we use a simple mass-spring-damper system to show the advantages of Stochastic RG (SRG) over standard RG. The continuous-time model is given by: $m\ddot{y} =$

$F(t) - b\dot{y} - ky$, where $b = 0.2\,\mathrm{Ns/m}, k = 1\,\mathrm{N/m}, m = 1\,\mathrm{kg}$. We introduce the state vector $x = (x_1, x_2)$, where $x_1 = y$ and $x_2 = \dot{y}$, and rewrite this model in state space (controllable canonical) form. A state feedback law with feedforward is then designed to control this system: $F = K_x x + K_v v + F_d$, where the state feedback gain, K_x, is designed so that the closed-loop poles are placed at $-0.5 \pm 0.995j$, the feedforward gain $K_v = 1.24$ is selected to ensure unity dc-gain from v to y, v is the signal to be governed by the SRG, and F_d is the disturbance on the control actuator. This disturbance is generated randomly from a uniform distribution with $F_d \in [-0.2\,\mathrm{N}, 0.2\,\mathrm{N}]$. Finally, we assume that the position and force are constrained by: $|F| \leq 1\,\mathrm{N}, \quad |x_1| \leq 0.7\,\mathrm{m}$. Note that this example considers the case where $D_w \neq 0$ and $D_v \neq 0$.

The closed loop model is discretized using the zero-order hold technique with a sampling time of $T_s = 0.1s$. For ease of illustration, it is assumed that F_d is constant during the sampling period. Using the resulting discretized model, we compute both MAS and SR-MAS. The confidence level for SR-MAS is taken to be 70 % (i.e., $\beta = 0.3$). The latter is selected to illustrate clearly the difference between the two schemes. In practice, confidence levels are normally higher, for example 95%. We perform the numerical test explained next 100 times with different realizations of the disturbance. The numerical test consists of simulating the response of the system starting from zero initial conditions to $r(t)$ shown by the black dashed line in Fig. 3.1.

The constrained position and force outputs for SRG and standard RG are depicted in Fig. 3.2 for one run of the simulation. As can be seen, the former has a less conservative response compared to the latter. This is possible to see from both the position and force outputs, where the output corresponding to the SRG (red solid line) is not limited as much as in the standard RG (blue dashed line).

The position output, for both schemes, satisfies the constraints for all times. However, for the force output, only the standard RG scheme satisfies the constraints for all times. This

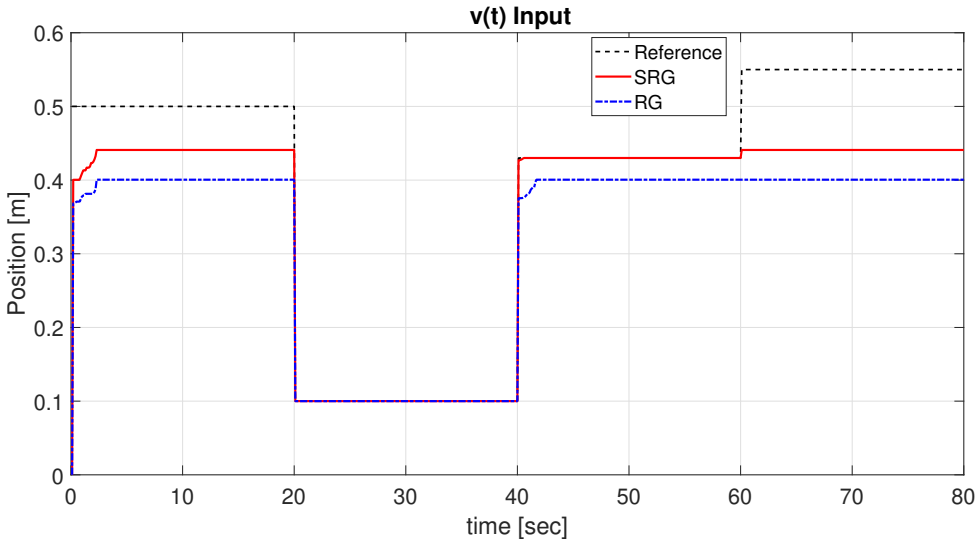

Figure 3.1: Reference governor output for SR-MAS and MAS

is because of the conservative margin created by the worst case disturbance approach. On the other hand, the SRG has some constraint violation within the confidence level of 30%. This is evident from Fig. 3.3, which, for each timestep, shows the percentage of simulations in which the output violated the constraint. The worst case probability of violation for SRG is 24 %, which is lower than the 30 % used in design. Note that the actual violation is below the 30% used for design because the SR-MAS, $\hat{O}_\infty^\beta$, is an inner approximation of the actual stochastic MAS.

The corresponding RG outputs, for both stochastic and standard RG, are depicted in Fig. 3.1, where it can be seen that the SRG produces higher control signal values compared to standard RG. This leads to a less conservative constrained output response, as is depicted in Fig. 3.2. Thus, SRG leads to improved system performance.

Finally, Fig. 3.4[1] shows a cross section of the polytopes $\hat{O}_\infty^\beta$ (used in SRG) and $\bar{O}_\infty$ (used in RG). The conservative margin introduced by the worst case approach can be clearly seen.

[1]Plot is generated by MPT 3.0 toolbox [83].

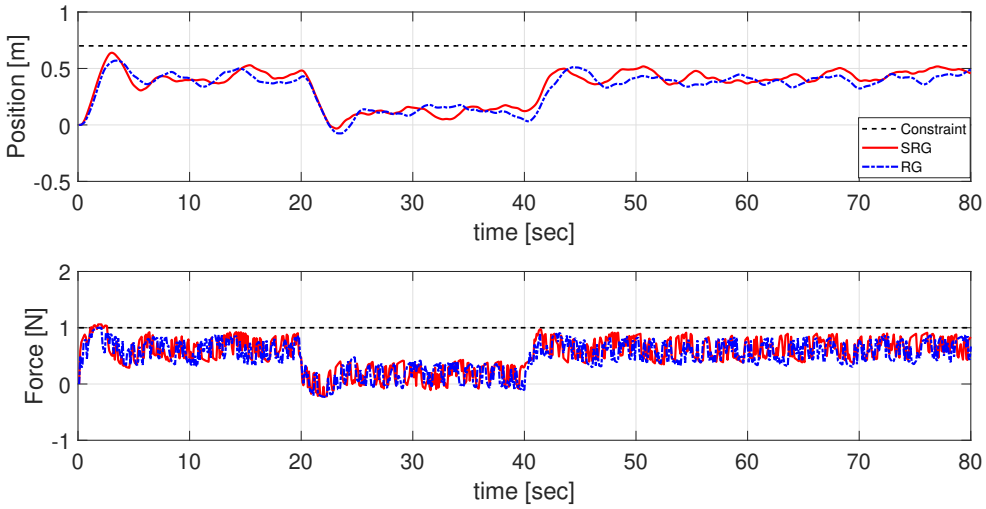

Figure 3.2: Position and force outputs

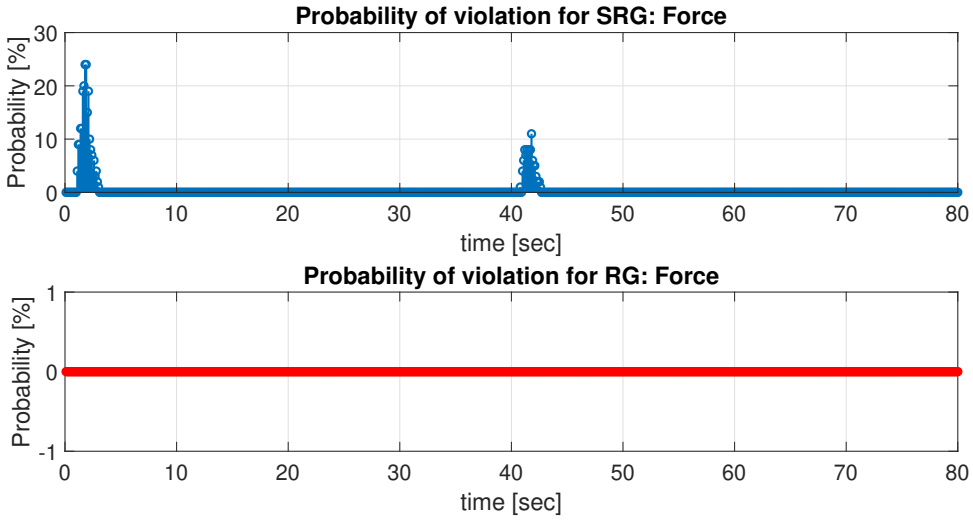

Figure 3.3: Probability of constraint violation over 100 simulations for Stochastic RG and RG.

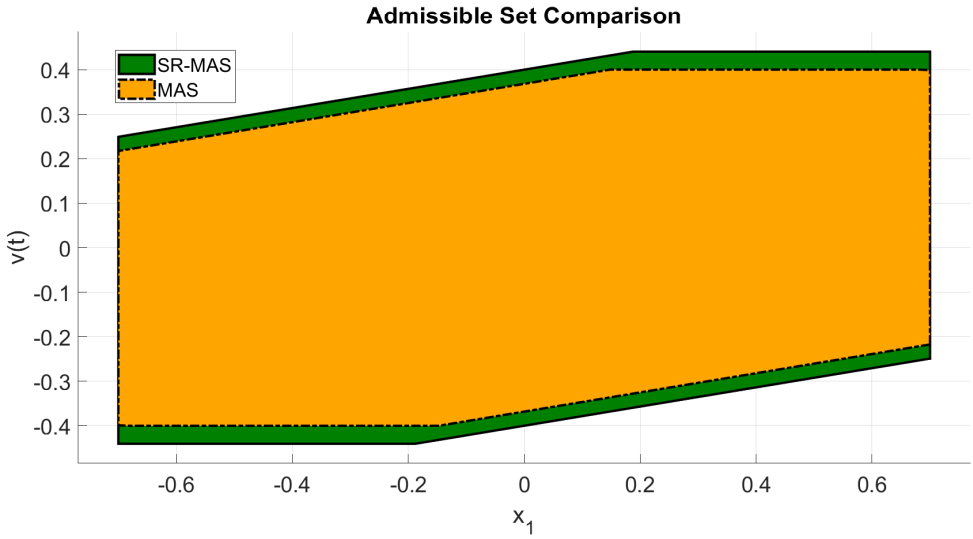

Figure 3.4: SR-MAS and MAS polytopes

Chapter 4

Recovery Reference Governor

This chapter addresses the infeasibility problem of standard RG by proposing a simple and real-time feasible solution that can recover the closed-loop system to a feasible operating condition after a constraint violation is detected. The solution, referred to as the Recovery RG (RRG), is based on a set-theoretic approach. Recovery is achieved by using an estimation of the external disturbances affecting the system. These estimated disturbances could model exogenous disturbances or plant/model mismatch. Once constraint violation is detected, the RRG is executed in order to contract the governed input $v(t)$ (see Fig. 1.1). When the constrained output recovers from violation, the standard RG formulation is executed based on a new MAS, which has an updated disturbance information.

4.1 Recovery Reference Governor (RRG)

In order to motivate the RRG, the following example is used. Consider the discrete-time first order system:

$$\begin{aligned} x(t+1) &= 0.5x(t) + v(t) + w(t) \\ y(t) &= x(t) + v(t), \end{aligned} \tag{4.1}$$

with the constraint $y(t) \in [-1, 1], \forall t \in \mathbb{Z}_+$, and disturbance $w(t) \in [-0.01, 0.01]$. Applying the standard RG based on the theory presented in Section 2.1.1.1.1.2, the corresponding $\bar{O}_\infty$ for system (4.1), for $t = 10$ (i.e., large enough for this example) and $\epsilon = 0.0001$ in the

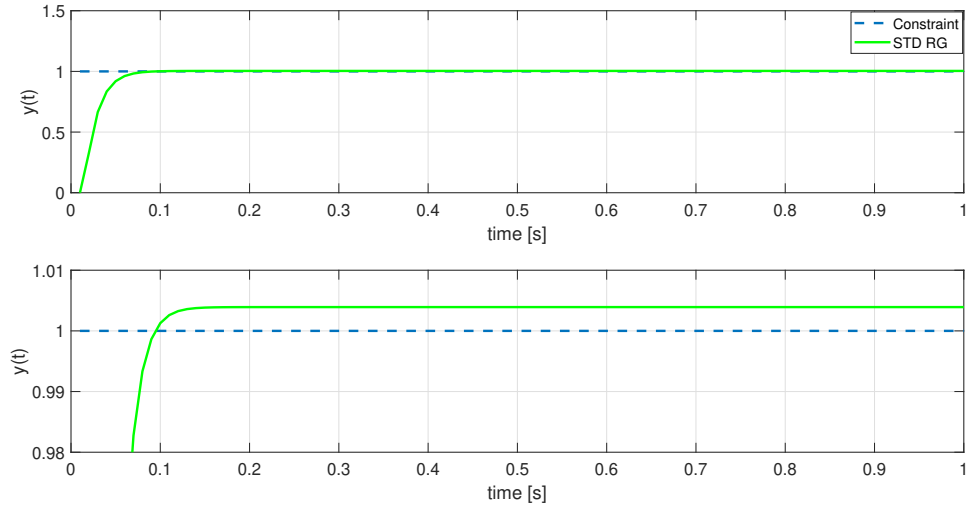

Figure 4.1: Constrained output, example model. Bottom plot shows a zoom in of the ordinate axis corresponding to the constrained output (top).

computation of $\bar{Y}$, is given by:

$$\bar{O}_\infty = \left\{(x_0, v_0) : \begin{bmatrix} 0 & 3 \\ 1 & 1 \\ 0.5 & 2 \\ 0 & -3 \\ -1 & -1 \\ -0.5 & -2 \end{bmatrix} \begin{bmatrix} x_0 \\ v_0 \end{bmatrix} \leq \begin{bmatrix} 0.9799 \\ 1 \\ 0.99 \\ 0.9799 \\ 1 \\ 0.99 \end{bmatrix} \right\}.$$

Assume that $x(0) = 0$, and $r(t) = 1, \forall t \geq 0$. Now, suppose the disturbance set assumed during the design phase is not accurate and the actual disturbance affecting the system is $w(t) = 0.012$ for $t \geq 0$. Then, under this condition, applying the RG defined by (2.18) will compute $v(t) = 0.3266$, $\forall t \geq 0$, which will lead to $y(t)$ violating the constraint in the presence of this disturbance, without possibility of recovering. This is illustrated in Fig. 4.1.

In order to remedy the problem exposed in the previous example, we propose a formulation capable of handling unknown disturbances affecting the system. Note from the example that the violation arises due to an inaccurate assumption regarding the set W. Thus, we first tackle this part of the problem.

4.1.1 Disturbance Estimation

A persistent violation may occur when an unknown disturbance affects the system for a sustained period of time. In order to recover from constraint violation, we first investigate an effective way to estimate the disturbance. If the model is well known, a standard approach is to design an observer based on system (2.8). However, the way the disturbance enters the system is not always available, i.e., B_w and D_w in (2.8) may be unknown, for example when the disturbance is due to plant/model mismatch. We overcome this lack of information by extending some of the ideas presented in [206], with the distinction that we implement a fixed-gain Kalman filter (i.e., a Luenberger observer) to estimate the states and disturbances.

Consider system (2.8), where B_w and D_w are unknown. Assume that $w(t)$ represents unmeasured signals lumping the effects of plant/model mismatch and all exogenous disturbances affecting the real process.

To see how $w(t)$ can capture the effects of plant/model mismatch, suppose that the real plant is given by the following nonlinear model:

$$x(t+1) = f(x(t), v(t), d(t)), \quad y(t) = g(x(t), v(t), d(t)), \tag{4.2}$$

where d is a vector of real disturbances. Note that (4.2) can be transformed into (2.8) by defining $B_w w(t) = f(x(t), v(t), d(t)) - Ax(t) - B_v v(t)$, and $D_w w(t) = g(x(t), v(t), d(t)) - Cx(t) - D_v v(t)$. Hence, without loss of generality, in our reference governor and observer

models discussed below, we assume that $w \in \mathbb{R}^{n+p}$ and the *disturbance model*:

$$B_w = \begin{bmatrix} I_n & 0_{n\times p} \end{bmatrix}, \quad D_w = \begin{bmatrix} 0_{p\times n} & I_p \end{bmatrix}, \tag{4.3}$$

so that the disturbance affects every state and every output. Note that, in general, the constrained output y is not measured. In these situations, the observer must be designed using an appropriate model of the measured output (i.e., C, D_v, and D_w matrices).

A. 4.1.1. *For observer and RRG models, assume $\hat{w}$ is constant for all time, i.e., $\hat{w}(t+1) = \hat{w}(t), \forall t \in \mathbb{Z}_+$.*

Using the above assumptions, we design a Luenberger observer to estimate the unmeasured disturbance $\hat{w}$.

Remark 4.1.1. *A different disturbance model may be considered for the design of the observer. For instance, in [206], a dynamic observer is presented to estimate disturbances and uncertainties based on H_∞ control theory. However, for the scope of this paper, and since, rather than tracking performance, we are interested in constraint management, we use the disturbance model defined in* (4.3).

4.1.2 Recovery Reference Governor (RRG)

The problem of recovery from constraint violation is addressed with the RRG, whose flow diagram is shown in Fig. 4.2. Note from (2.8) that the constrained output, $y(t)$, depends on the current input, $v(t)$, which has not yet been computed at time t. Hence, the RRG will be executed when the *previous* output violates the constraint (i.e., $y(t-1) \notin Y$), as shown in Fig. 4.2. Under this condition, the RRG computes the governed input, for the current time step, based on an estimate of the disturbance that produced the constraint violation.

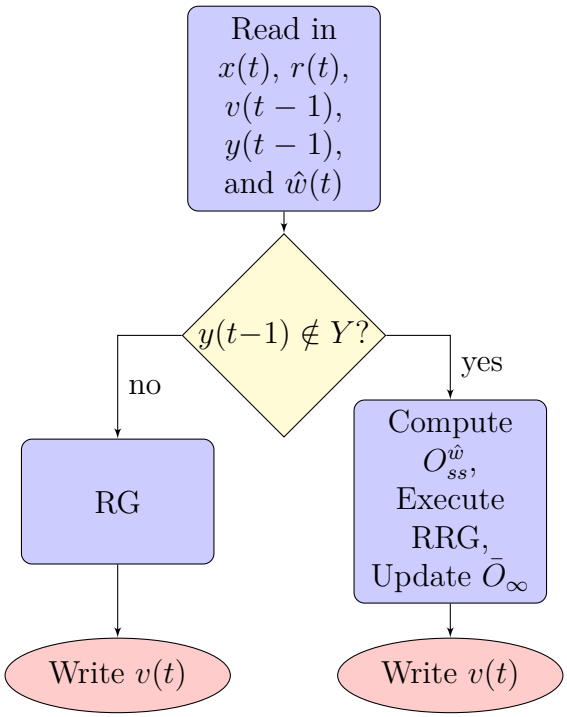

Figure 4.2: RRG flow diagram, executed at each time step

Remark 4.1.2. *Although not explicitly shown in Fig. 4.2, it is a good practice to consider a hysteresis band around the constraint to avoid toggling due to the activation of the RRG because of a small violation.*

The recovery scheme is basically composed by three main elements: Computation of $O_{ss}^{\hat{w}}$, execution of the RRG, and the updating process of $\bar{O}_\infty$. These are explained below.

4.1.2.1 Computing $O_{ss}^{\hat{w}}$ and executing RRG

During constraint violation, we switch from the RG based on $\bar{O}_\infty$ to the RRG based on the set $O_{ss}^{\hat{w}}$ defined as follows:

$$O_{ss}^{\hat{w}} := \{(v_0, \hat{w}_0) : H_0 v_0 + H_{0_w} \hat{w}_0 \in (1-\alpha)\bar{Y}\}, \tag{4.4}$$

where $H_{0_w} = C(I-A)^{-1}B_w + D_w$, is the DC gain from the disturbance input to the output. The positive calibration parameter $\alpha \in \mathbb{R}$ is introduced to allow flexibility on the steady state contraction. It is trivial to see that (4.4) only uses the steady-state properties of the system, which means that finite determination of $O_{ss}^{\hat{w}}$ is not an issue. Note that, without α and $H_{0_w}\hat{w}_0$, (4.4) is the steady-state halfspace in (2.16).

Remark 4.1.3. *Another approach for defining $O_{ss}^{\hat{w}}$ is using the steady-state halfspace in* (2.16)*: $O_{ss}^{\hat{w}} := \{v_0 : H_0 v_0 \in \bar{Y}\}$, where $\bar{Y}$ is now updated as follows. We first update the set W to capture the worst-case estimated $\hat{w}(t)$, and recompute the iteration* (2.10) *and, hence, $\bar{Y}$. While this approach takes the new information about the worst-case disturbance into account and can recover from violation at steady state, it is too computationally demanding. Therefore, we prefer to introduce a calibration parameter α and use the formulation* (4.4)*. Note that the recovery speed may be affected by the parameter α. If α is too small, then the recovery process may be slowed down. On the other hand, if this parameter is too large, then the recovery process may be fast, but may lead to an excessively conservative response.*

Based on $O_{ss}^{\hat{w}}$, the RRG formulation is given by:

$$v(t) = \kappa r(t),$$

where κ is computed from:

$$\begin{aligned} \max_{\kappa \in [0,1]} \quad & \kappa \\ \text{s.t.} \quad & (v_0, \hat{w}_0) \in O_{ss}^{\hat{w}} \\ & v_0 = \kappa r(t), \\ & \hat{w}_0 = \hat{w}(t), \end{aligned} \tag{4.5}$$

where $\hat{w}(t)$ is the estimated disturbance at time t. Note that (4.5) is similar to the static RG studied by Gilbert et. al. in [207], but with the distinction of the introduction of the set $O_{ss}^{\hat{w}}$. Because of the use of $O_{ss}^{\hat{w}}$, this formulation ensures that if $\hat{w}(t)$ converges to $w(t)$, then the computed input, v_0, ensures recovery from violation at steady state. This is shown in Section 4.2.

Note from (4.5) that in order for the equilibrium (i.e., $v_0 = 0$, which corresponds to the equilibrium input at which the nonlinear system is linearized) to be admissible, the following condition must be satisfied:

$$H_{0_w} \hat{w}_0 \in (1 - \alpha)\bar{Y}. \tag{4.6}$$

A. 4.1.2. *It is assumed that the equilibrium is admissible for the implementation of the RRG.*

Future works will explore the case when this assumption is not satisfied.

4.1.2.2 Update $\bar{O}_\infty$

Once the constrained output recovers from violation, the standard RG formulation is used, i.e., Equation (2.17), with the difference that $\bar{O}_\infty$ has been updated with the estimated disturbance. Specifically, we redefine the halfspaces in (2.16) as:

$$\begin{aligned}\bar{P}_i := \{(\hat{x}_0, \hat{w}_0, v_0) \in \mathbb{R}^{2n+p+1} : H_0 v_0 + H_{0_w} \hat{w}_0 \in \bar{Y}, \\ CA^i \hat{x}_0 + H_v(i) v_0 + H_w(i) \hat{w}_0 \in Y_i\},\end{aligned} \tag{4.7}$$

where $H_v(i) = (C(I-A)^{-1}(I-A^i)B_v + D_v)$, and $H_w(i) = (C(I-A)^{-1}(I-A^i)B_w + D_w)$. In definition (4.7), $\hat{w}(t)$ is assumed to be known and constant for all times, which is consistent with Assumption 4.1.1.

Remark 4.1.4. *Note from* (4.7) *that for the implementation of the RRG, the two extra components that need to be pre-stored in memory are* H_{0_w} *and* $H_w(i), i = 0, \ldots, t^*$. *The other components of* (4.7) *(i.e.,* $H_0, CA^i, H_v(i), \bar{Y}$, *and* Y_i*) are known from the standard RG.*

Remark 4.1.5. *If the disturbance affecting the system vanishes, then* (4.7) *becomes* (2.16) *for the given disturbance model (i.e.,* B_w *and* D_w*).*

4.2 Analysis of Recovery Reference Governor

In this section, we introduce two conditions on the disturbance that lead to constraint violation and, hence, the RRG execution. We also present the theoretical result that shows under what conditions the RRG recovers the output from constraint violation.

Below, we denote the transfer function matrix from the disturbance, w, to the constrained output, y, by $G(z)$.

Proposition 4.2.1. *Suppose that for all $i = 1, \ldots, p$, there exists a j, such that $G_{i,j}(z) \neq 0$ (i.e., either the ij-th element of $D_w \neq 0$ or the ij-th element of $CA^kB_w \neq 0$ for some k). Then, for all $t_0 \geq k$, there exists a sequence $\{w(t)\}_{t=0}^{t_0}$, with one or more $w(t) \notin W$, such that $y(t_0) \notin Y$.*

Proof. If $D_w \neq 0$, then $w(t)$ large enough will lead to constraint violation, since the constraint set Y is compact. If $D_w = 0$, then $CA^kB_w \neq 0$, which implies that the effect of the disturbance will manifest in the output after $k+1$ delays. Thus, $w(t-k-1)$ large enough will lead to constraint violation. □

Proposition 4.2.1 provides a sufficient condition for the RRG to be executed. Next, we show, for a specific case, that there is an upper bound of the sequence $\{w(i)\}_{i=0}^{t}$, which, if violated, will lead to the execution of the RRG at steady-state.

Corollary 4.2.1. *Assume that the DC gains from each disturbance to each of the constrained outputs are positive, the disturbance is held constant for all times, and suppose $r(t) \notin \bar{V}$. Then, there exist upper bounds $\bar{w}_j$, with $j = 1, \ldots, n+p$, such that the RRG will be executed at steady-state if any one $w_j > \bar{w}_j$.*

Proof. We show the proof for the case when (2.8) has one output. The proof for the case with multiple outputs follows similar logic. Let the polytopic constraint set Y be given by:

$$Y = \{y : Sy \leq s\}, \tag{4.8}$$

where S is a scaling matrix and s is the constraint imposed on the output. Next, let the

polytope $\bar{Y}$ used in (2.4) be given by:

$$\bar{Y} = \{y : Sy \leq \tilde{s}\}. \tag{4.9}$$

For simplicity, let the scaling matrix $S = 1$. Note that, since $r(t) \notin \bar{V}$, the RG will ensure that $v(t)$ converges to the boundary of $\bar{V}$ at steady-state, i.e., $H_0 v_0(t) \to \tilde{s}$. Thus, we have from (4.8) and because the disturbance is assumed to be held constant, that to satisfy the constraint at steady-state:

$$\begin{aligned} & H_0 v_0 + H_{0_w} w \leq s \\ \Rightarrow\ & H_{0_w} w \leq s - H_0 v_0. \\ \Rightarrow\ & H_{0_w} w \leq s - \tilde{s}. \end{aligned} \tag{4.10}$$

Let the j^{th} element of H_{0_w} be defined by $h_{0_{w_j}}$, then for each w_j (j^{th} element of w), $j = 1, \ldots, n+1$, we can find each corresponding upper bound as:

$$\bar{w}_j = \frac{1}{h_{0_{w_j}}}(s - \tilde{s}).$$

It is obvious that any j^{th} disturbance bigger than $\bar{w}_j$ will produce constraint violation at steady-state, which will trigger the execution of the RRG scheme (see Fig. 4.2). □

The following theorem provides the theoretical guarantees of applying the RRG to recover from constraint violation.

Theorem 4.2.1. *For system* (2.8) *affected by unknown disturbances, assume that there is an asymptotically stable observer such that* $\hat{w}$ *converges to* w *at steady state. Also assume that condition* (4.6) *and Assumption 4.1.1 hold. Then, the RRG scheme will recover the constrained output from violations at steady-state.*

Proof. Since there is an observer to estimate $\hat{w}$, which converges to w at steady-state and thanks to Assumption 4.1.1, one can define the output of (2.8) at steady-state as

$y_{ss} = H_0 v + H_{o_w} \hat{w}$. With the RRG in the loop, the output becomes:

$$y_{ss} = H_0 \kappa r(t) + H_{o_w} \hat{w}(t), \tag{4.11}$$

which by definition (4.4) must satisfy $y_{ss} \in (1-\alpha)\bar{Y}$, and since $(1-\alpha)\bar{Y} \subset Y$, this implies that $y_{ss} \in Y$ at steady-state. Then, there will be a $\kappa \in [0, 1]$ for (4.11) such that the constrained output recovers from violations at steady-state.

□

Remark 4.2.1. *Note that the existence of an asymptotically stable observer based on* (2.8) *to estimate the disturbance at steady state will ensure that the constrained output recovers from violations. Once the observer guarantees $\hat{w} = w$ at steady state, the updated version of $\bar{O}_\infty$ based on* (4.7) *will capture the disturbance effects.*

Remark 4.2.2. *During constraint violation, the RRG implements* (4.5)*. After recovery, depending on the system characteristics, one may have the condition where the standard RG with an updated $\bar{O}_\infty$ is still infeasible, because the RRG only enforces constraints at steady-state not transient. In such a case, we enforce $\kappa = 0$ to command the previous input; however, this leads to constraint violation. The latter will activate the RRG execution, which will compute a new admissible steady-state input, based on the new disturbance. The output eventually converges to a value that satisfies the constraint. Thus, the RRG may not ensure recursive feasibility, but it enforces constraints at steady-state.*

4.3 Turbocharged Engine Simulation

In this section, we illustrate the application of the RRG to a turbocharged gasoline engine model, depicted in Fig. 4.3. In a turbocharged engine, the throttle controls the airflow into the engine, which directly controls the engine torque. The turbine extracts energy from

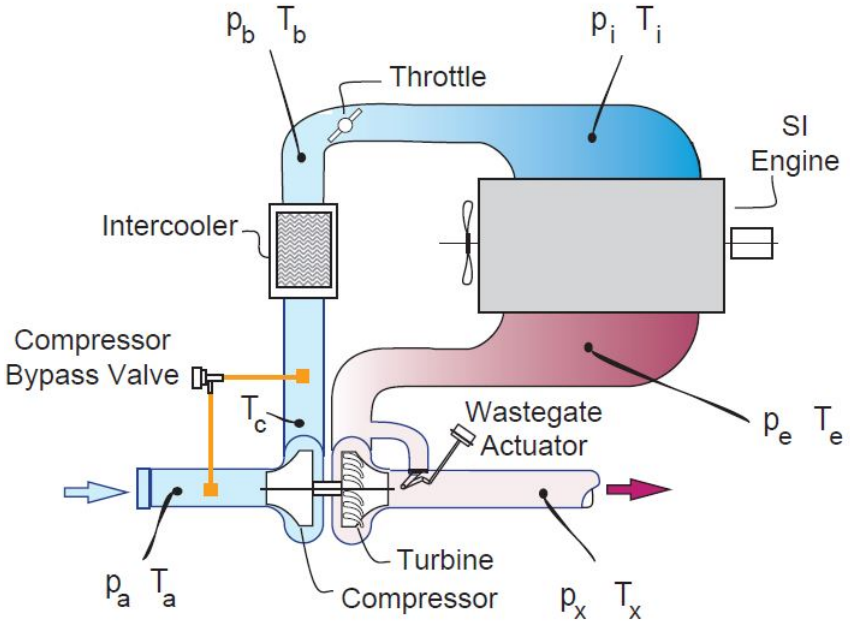

Figure 4.3: Schematic of a turbocharged gasoline engine [2]

exhaust gasses and drives the compressor, which increases pressure at the throttle inlet, allowing more airflow to enter the engine.

The dynamic equations for the turbocharged gasoline engine are as given in [208]. The turbocharged gasoline engine model includes an air-path controller, which is set to control the desired engine air mass flow rate, which is the reference r being governed. To this end, a gain scheduled PI controller to control the wastegate flow, and a feedforward controller to regulate throttle position are implemented. The closed-loop system is linearized at an operating point.

The model has B_w and D_w defined based on (4.3). The state vector

$$x = \begin{bmatrix} P_i, & P_b, & P_e, & N_{tc}, & W_{wg}, & x_{c_1}, & x_{c_2} \end{bmatrix}^\top,$$

represents the intake pressure, boost pressure, exhaust pressure, turbo speed, wastegate

flow, and two controller states. The constrained output is the turbo speed, which for the purpose of this paper we choose to be a delta value above the linearization point, this is:

$$\delta N_{tc} \leq 2,000\, RPM. \tag{4.12}$$

Note that constraint (4.12) does not represent a compact constraint set as assumed earlier in this paper. However, for the purpose of this application, not considering a compact constraint set does not affect the implementation of the RG or the RRG. For this reason we use the constraint presented in (4.12). The system has one constrained output, i.e., $p = 1$. This implies that the unknown disturbance w affecting the system belongs to $\mathbb{R}^8$. We assume the real disturbance affects the turbo speed and wastegate flow states, and an estimation noise affects the constrained output. The disturbance affecting the two states is defined as $w_s \in [-0.05, 0.05]$ and the disturbance affecting the constrained output is $w_y \in [-1.5, 1.5]$. The final disturbance vector is given by:

$$w = [0, 0, 0, w_s, w_s, 0, 0, w_y]^\top.$$

With all the elements in place, we discretize the closed loop system with a sample time of $0.015s$. The model has the structure of (2.8), where A, B_v, C, and D_v are provided in the Appendix of [45]. This linear model was obtained from linearizing a nonlinear model of the turbocharged for an operating condition corresponding to a desired air mass of 19 lb/min. Note that B_w and D_w are as in (4.3). The fixed gain Kalman Filter designed for this system is tuned by more heavily weighting those disturbance states that correspond to turbo speed, wastegate flow, and the output.

The parameter α was defined for these simulations to be 0.1. The selection of this parameter is based on how much margin, with respect to the constraint, is desired to have at steady-state, in this case a 10% margin was added after constraint violation. Also, as

previously explained, this is introduced to the RRG formulation in order to allow some flexibility in the computation of (4.4).

Next, we compute $\bar{O}_\infty$, and perform the following numerical experiment: First, starting the system from zero initial conditions, we request a fixed reference (delta of the desired air mass value above the equilibrium), and during normal operation, fixed unknown disturbances affect the system through the channels that correspond to turbo speed, wastegate flow, and constrained output as explained above. We assume $w_s = 0.12$ and $w_y = 3.6$ starting from $t = 4.5s$. Both values are outside of W.

Fig. 4.4 shows the fixed desired air mass reference that is requested from the engine in blue. The dash-dot black signal corresponds to the governed input computed by the standard RG. The solid red line is the governed input computed by the RRG.

Fig. 4.5 shows the turbo speed output from the engine. The dotted cyan response corresponds to the unconstrained output (i.e., no constraint management scheme in the loop), which as expected violates the constraint. The dash-dot black response corresponds to the standard RG response, which satisfies the constraint until the unknown disturbances affect the system at $4.5s$, and it keeps violating while the disturbances are present. The solid red line corresponds to the RRG response, which recovers as the estimated disturbance is detected and fed back to the RRG.

A second test is performed, with the distinction that now the reference varies. Similar to the first experiment, in this second test the disturbances affect the system at time $t = 4.5s$ and are held constant. The results of the second experiment are in Fig. 4.6 and Fig. 4.7. The former shows a variable reference that is requested from the engine and how the RG and the RRG restrict the full reference in order to avoid constraint violation. Since unknown disturbances affect the system, the RG holds the reference constant based on an inaccurate MAS. Even as $r(t)$ becomes admissible at $t = 11.25s$, the RG still maintains $v(t)$ at a constant value because the optimization problem (2.17) is infeasible. On the other hand,

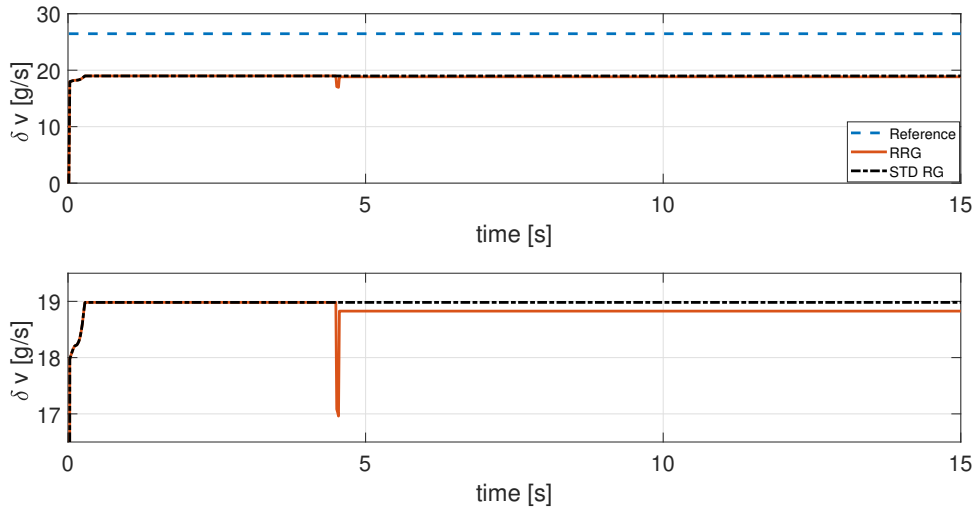

Figure 4.4: RRG vs. RG for fixed reference input. Bottom plot corresponds to a zoom in of the ordinate of the desired air mass delta input (top).

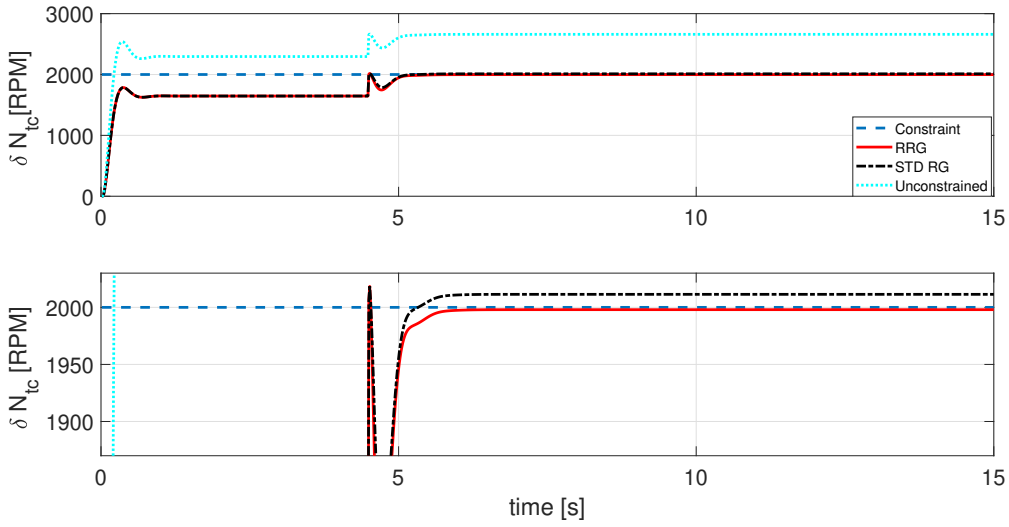

Figure 4.5: Turbo speed output for RRG vs. RG with fixed reference input. Bottom plot corresponds to a zoom in of the ordinate of the turbo speed delta output (top) above the equilibrium point.

the RRG restricts the reference, updates the MAS and considers information provided by the observer to transform the unknown disturbance into a known disturbance. With the

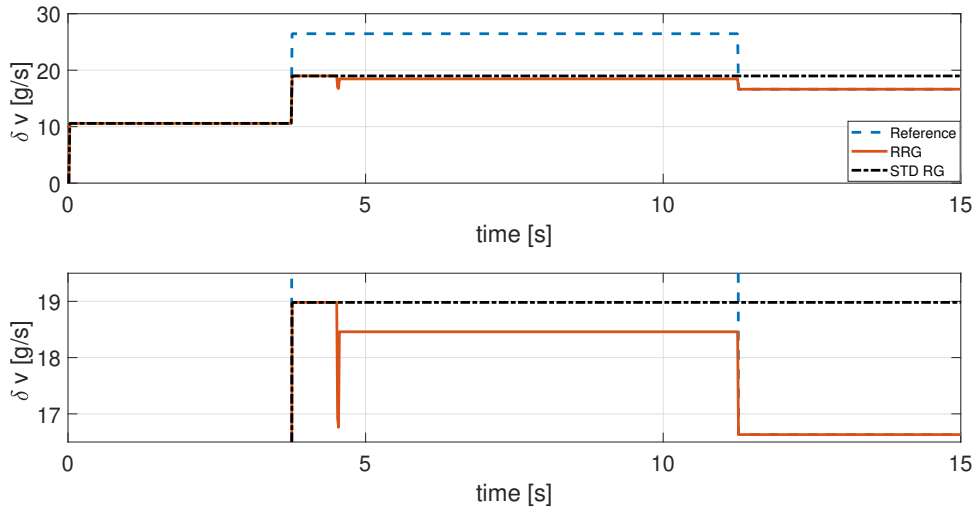

Figure 4.6: RRG vs. RG for variable reference input. Bottom plot corresponds to a zoom in of the ordinate of the desired air mass delta input (top) above the equilibrium point.

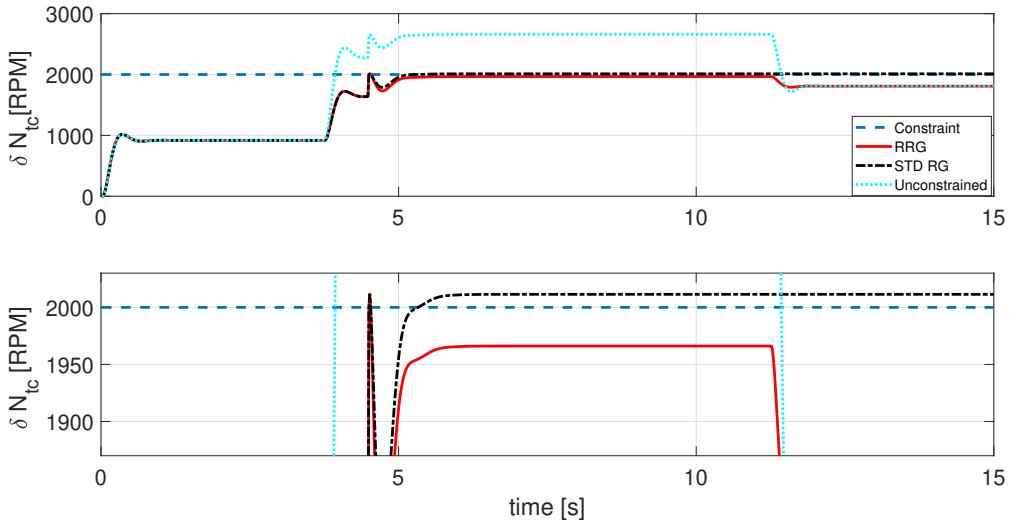

Figure 4.7: Turbo speed output for RRG vs. RG for variable reference input. Bottom plot corresponds to a zoom in of the ordinate of the turbo speed delta output (top) above the equilibrium point.

RRG in the loop, the system is able to recover from constraint violations, and, since $\bar{O}_\infty$ is updated based on the estimated disturbance, violations due to a similar disturbance are

prevented from happening in the future.

Remark 4.3.1. *The design of the disturbance model (i.e., B_w and D_w) affects the DC gain from disturbance input to the output, which will be reflected on the performance of the RRG in the loop.*

Note that the time constant of the observer is important for a smooth recovery response. A bad observer design may lead to toggling among different RRG modes, which is undesirable. However, increasing the estimation speed of the fixed gain Kalman Filter can produce a higher noise sensitivity, which may be undesirable.

Chapter 5

Reference Governor for Nonlinear Systems

This chapter presents a novel scheme for constraint management of nonlinear systems. The Transient Robust RG (TR-RG) addresses the difficulties that were discussed in Chapter 1, Section 1.2.3. This scheme, whose block diagram is presented in Fig. 5.1, leverages a reference governor, as well as the steady-state characterization of the nonlinear system, which is often available in practice, for instance in the automotive industry, where engine maps are developed on dynamometers. More specifically, Fig. 5.1 shows a closed-loop nonlinear system with a constrained output $y_{non}(t)$, which is constrained as follows: $y_{non}(t) \in Y, \forall t$, where Y is a specified set. To compute $v(t)$ that enforces these constraints, the TR-RG leverages a linearized prediction model of the nonlinear system (similar to [3, 35, 42–45]). However, two mechanisms are proposed to effectively account for the mismatch between the linear model and the nonlinear system outputs: the first one mitigates plant/model mismatch at steady-state by incorporating the forward (ζ) and inverse (ζ^{-1}) steady-state characterization of the nonlinear system in the loop (illustrated by blue curves in Fig. 5.1); and the second one is an RG that is implemented using a novel Robust Output Admissible Set (ROAS). The latter is obtained by using a data-driven approach to explicitly capture the transient mismatch between the responses of the nonlinear system and the linearized model. Note that if no modification of the governed input $v(t)$ is required, then $r(t) = v(t)$, which is possible thanks to the introduction of ζ and ζ^{-1}.

The TR-RG guarantees closed-loop system stability, as well as constraint satisfaction at steady-state without introducing any conservative margins. However, to enforce constraints during transients, TR-RG introduces a dynamic transient margin thanks to the ROAS. Note that the introduction of this margin is similar in philosophy to Lyapunov-based approaches [47, 49], where the choice of the Lyapunov function may lead to a conservative response. However, since TR-RG does not rely on a Lyapunov function, it may lead to a less conservative response as compared to Lyapunov-based methods, at the expense of theoretical guarantees of constraint satisfaction during transients. Finally, other approaches that

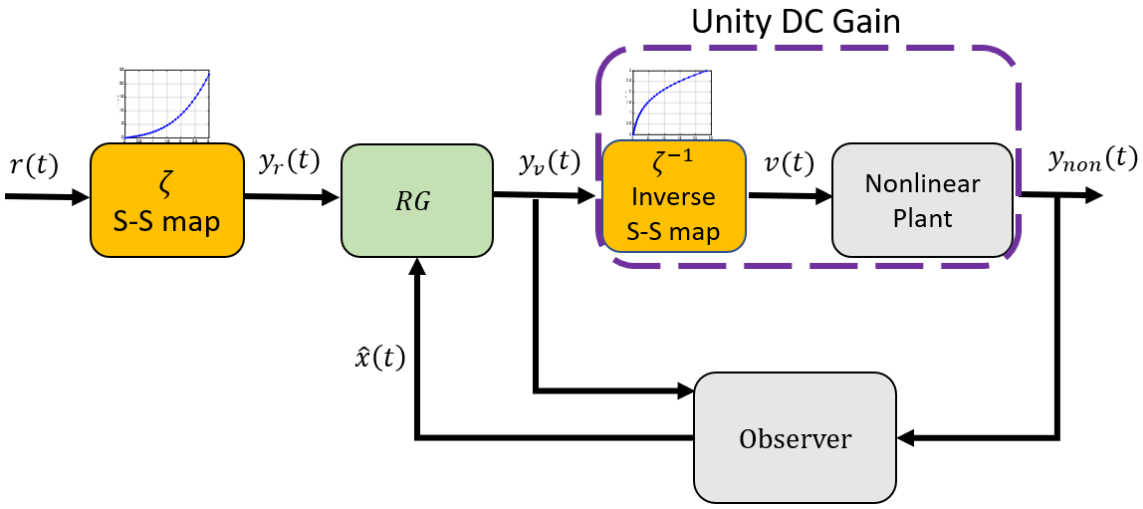

Figure 5.1: TR-RG block diagram. The TR-RG scheme refers to the use of the RG with an ROAS, together with the forward and inverse steady-state mappings (i.e., the green and yellow blocks).

exploit Lyapunov-based methods plus learning algorithms [51], as well as machine learning techniques that aim to learn the MAS [126] based on data, have been studied for constraints management of nonlinear systems. The work presented in this chapter is similar in spirit to [51, 126], since an admissible set is constructed using data. However, TR-RG offers theoretical guarantees at steady-state, maintains the simplicity of standard RG update law, and relies on ROASs that preservers properties of polytopic MASs.

Two remarks on TR-RG are in order. First, due to the data-driven nature of TR-RG, it can be applied to constraint management of black-box uncertain system (i.e., systems with inaccurate or even no analytical models) with the only requirement of having input-output measurements/estimations available. This offers an alternative to existing solutions that rely on data but require a parameterized structure to define admissible linear models to build a Robust Positively Invariant (RPI) set [92]. Also, TR-RG is different from formulations that study systems with parametric uncertainties based on RPI sets (e.g., [80, 209] for linear

systems and [69,210] for nonlinear systems) in that TR-RG only introduces margins during transients but not at steady-state.

Second, the decomposition of the TR-RG design process into two parts (namely, steady-state and transient) is similar, in spirit, to traditional robust control methods [211], wherein low frequency components of the system's frequency response are assumed to be accurately known, while the high frequency components are uncertain. In these methods, similar to TR-RG, controllers are designed to be robust against these uncertainties.

5.1 Problem Formulation and Motivating Example

5.1.1 Problem Formulation

Consider Fig. 5.1, where the nonlinear plant is defined by:

$$\begin{aligned} \dot{x}_{non} &= f(x_{non}, v) \\ y_{non} &= h(x_{non}, v) \end{aligned} \tag{5.1}$$

where $y_{non} \in \mathbb{R}^p$ is the constrained output, $x_{non} \in \mathbb{R}^n$ is the nonlinear system state vector, and $v \in \mathbb{R}$ is the governed input of the nonlinear system. Over the output we impose the constraint $y_{non} \in Y$, where Y has the structure: $Y = Y_1 \times Y_2 \times \ldots \times Y_p$, each $Y_i = \{y_i \in \mathbb{R} : y_i \in [\underline{s}_i, \bar{s}_i]\}$ and $\underline{s}_i < 0$ and $\bar{s}_i > 0$ are scalars. Note that this implies that $0 \in \text{int}(Y_i)$, which is consistent with the assumptions over the set Y established in Section 2.1.1.1.1. The following assumptions are imposed on (5.1).

A. 5.1.1. *It is assumed that the equilibrium of* (5.1) *is globally asymptotically stable for*

every constant input v. In addition, the system is BIBO stable from v to y_{non}.

A. 5.1.2. *It is assumed that y_{non} can be accurately measured or estimated in real-time. Furthermore, the steady-state mapping from v to y_{non} is assumed to be available. This mapping is denoted by the function $\zeta_i : \mathbb{R} \longmapsto \mathbb{R}, i = 1, \ldots, p$. Also, it is assumed that each ζ_i is continuous, satisfies $\zeta_i(0) = 0$, and is invertible.*

Note that assumption (A.5.1.2) covers a wide range of applications of nonlinear systems that have the steady-state mapping available. For example, it is common practice in the automotive industry to map combustion engines at different operating conditions. Note that if the steady-state map is not known accurately, or if the map varies slowly with time, our method can still be applied, as discussed in Sections 5.2.2 and 5.3.2.

The goal of this paper is to design an RG-based scheme for system (5.1), which can compute an input v as close as possible to a given reference r at each time, such that the constraints on the output are enforced for all times.

5.1.2 Constraint Management of Turbocharged Engines

The main application considered in this work is a turbocharged gasoline engine, which is a highly nonlinear system that satisfies the assumptions already established. In a gasoline turbocharged engine, the primary airpath actuators are the throttle and the wastegate, shown in the schematic diagram of Fig. 4.3. To deliver a desired engine airflow and, hence, torque, the throttle is actuated to control the airflow into the engine and the wastegate is actuated to control the pressure at the throttle inlet, known as the boost pressure. To ensure hardware durability, it is important to design constraint management strategies that

enforce constraints on the turbocharger speed and throttle inlet gas temperature [36]. Conventional hardware-protection strategies impose static limits on the desired boost pressure or desired engine airflow/torque that are determined based on the steady-state relationships between pressure, temperature and turbocharger speed. Although effective at enforcing the constraints, this static approach does not take into account system dynamics. Therefore, large robustness margins must often be included in the static limits to avoid constraint violation during transients. These offsets reduce maximum achievable boost pressure and therefore engine torque, potentially impacting performance.

We now present an example of standard RG (formulation explained in Section 2.1.1.1.1) applied to a nonlinear turbocharged gasoline engine system to motivate the need for the theory developed in this paper. This system is represented by a mean value turbocharged engine model described in [208, 212], together with an air-path controller, which is set to control the engine air mass flow rate to follow a desired value (which is the reference to govern). The closed-loop system is linearized at an operating point, and based on this linearization the RG is implemented. The constraint is imposed on the turbo speed, with a conveniently low constraint value of 140 kRPM. The simulation results for a step input in the desired air mass are presented in Fig. 5.2, where the top subplot shows the desired air mass reference (dashed line in red) and the governed reference (dashed-dotted line in black), and in the bottom the constraint (brown dashed line), the turbo speed response without an RG (dotted line in red), and the turbo speed response with standard RG (dashed-dotted line in black). As expected, because of the plant/model mismatch, the constraint imposed on the nonlinear plant cannot be satisfied by implementing a standard RG. An alternative solution is to implement multiple standard RGs based on linearized models obtained at different operating points. However, due to the mismatch between the nonlinear plant and the linear models, the maximal admissible sets within these RGs would not be positively invariant, which means that constraint satisfaction would still not be guaranteed. The

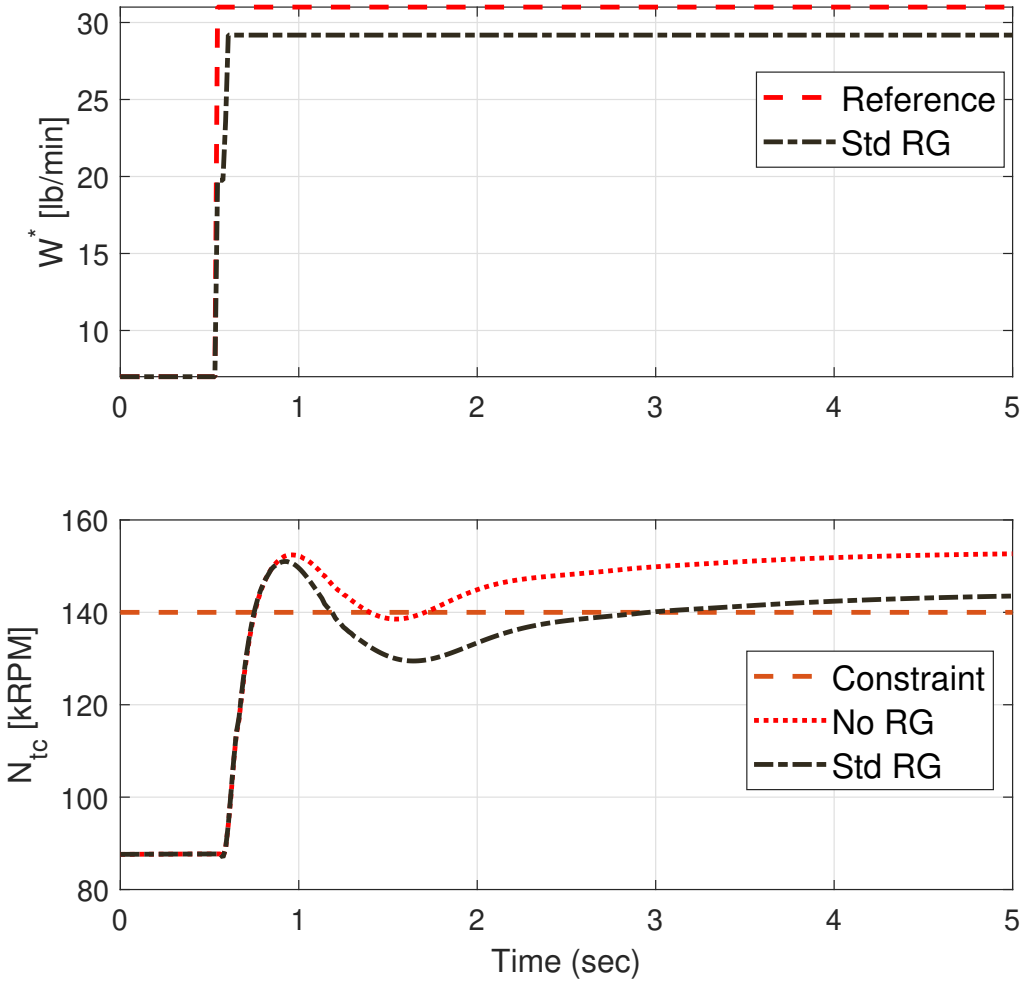

Figure 5.2: Nonlinear turbocharged engine simulation. Top plot shows the desired air mass input and bottom plot shows turbo speed

proposed scheme overcomes these issues by explicitly handling the plant/model mismatch, as explained next.

5.2 Transient Robust RG (TR-RG)

As explained in the Introduction and Section 5.1, the goal of TR-RG is to enforce the constraints $y_{non}(t) \in Y$ for the nonlinear system (5.1). This is achieved by breaking up the design in two stages: steady-state and transient, which we describe next. For the sake of clarity, we assume that $p = 1$ (i.e., one output) in this section. We will relax this assumption in Section 5.3.

To enforce the constraints at steady-state, we leverage assumption A.5.1.2 and introduce the steady-state mapping, as shown in Fig. 5.1, where the blue curves illustrate the forward mapping (ζ) and its inverse (ζ^{-1}). In Section 5.2.1, we show that this structure enforces the constraints at steady-state.

To enforce the constraints during transients, we use a linear approximation of the nonlinear system from $y_v(t)$ to $y_{non}(t)$, which can be obtained from linearization or system identification of the nonlinear system around an operating point. Let the linear model be described by:

$$\begin{aligned} x(t+1) &= Ax(t) + By_v(t) \\ y(t) &= Cx(t) + Dy_v(t), \end{aligned} \tag{5.2}$$

where the DC gain (i.e., $H_0 := C(I - A)^{-1}B + D$) is equal to 1, thanks to the introduction of ζ. Note that to avoid introducing new symbols, we have used, with a slight abuse of notation, the same A, B, C, D notation as in Section 2.1.1.1.1. For the rest of this discussion, we assume that system (5.2) satisfies Assumption A.2.1.1. Based on (5.2), we compute a Robust Output Admissible Set (ROAS), which we denote by O^r_∞. As will be explained in Section 5.2.3, ROAS is a subset of the standard Maximal Output Admissible set (MAS), but

shrunk using a data-driven approach to take transient plant/model mismatch into account. The RG, shown in Fig. 5.1, follows the standard formulation in (2.17) and (2.18), except that the RG uses the ROAS instead of the MAS to enforce the constraints on the nonlinear system. Specifically, the RG update law in our TR-RG framework is given by:

$$y_v(t) = y_v(t-1) + \kappa(y_r(t) - y_v(t-1)), \tag{5.3}$$

where κ is optimized using:

$$\begin{aligned} &\underset{\kappa\in[0,1]}{\text{maximize}} \quad \kappa \\ \text{s.t.} \quad &(x(t), y_v(t)) \in O_\infty^r \\ &y_v(t) = y_v(t-1) + \kappa(y_r(t) - y_v(t-1)) \end{aligned} \tag{5.4}$$

To summarize the scheme (Fig. 5.1), we have that at time t, given a reference $r(t)$, the signal $y_r(t)$ is calculated through the steady-state map ζ. Next, the RG computes $y_v(t)$ based on $y_r(t)$, $y_v(t-1)$, and $\hat{x}(t)$, where $\hat{x}(t)$ is an estimate of $x(t)$ obtained using an observer based on (5.2). If no violation is predicted, then $y_v(t) = y_r(t)$, otherwise a constraint-admissible input $y_v(t)$ is computed. Finally, the input of the nonlinear plant (5.1), $v(t)$, is computed by the inverse mapping ζ^{-1}.

Remark 5.2.1. *In the offline stage of the TR-RG design process, the signal $y_{non}(t)$ must be available for measurement or estimation. This is because $y_{non}(t)$ is needed to construct the steady-state map and to tune the ROAS. On the other hand, during the real-time implementation of TR-RG, this is no longer required. Indeed, if y_{non} is still available for measurement or estimation, the observer can be designed with y_{non} as the output injection term. However, if y_{non} is not available, then an open loop observer can be used (driven purely by y_v). Furthermore, note that the observer will have an estimation error that con-*

verges to zero at steady-state thanks the to introduction of the steady-state map in the loop (i.e., outputs of (5.1) *and* (5.2) *match at steady-state).*

5.2.1 Steady-State Constraint Enforcement

The result of introducing the steady-state mapping with RG for the case when $p = 1$ is summarized in the following theorem.

Theorem 5.2.1. *Assume that reference $r(t)$ is held constant. By applying the scheme presented in Fig. 5.1, the constraints on* (5.1) *are enforced at steady-state, i.e.,* $\lim_{t\to\infty} y_{non}(t) \in Y$.

Proof. For this proof, we refer to the MAS of (5.2) as $\bar{O}_\infty$, which follows the same structure as in (2.16). We will use the subscript 'ss' to denote steady-state value of the signal. Since $r(t)$ is constant, the input to the RG, $y_r(t)$, is also constant. From (5.3) and (5.4), the RG computes an input $y_v(t)$ which converges at steady state, that is, $y_{v_{ss}} := \lim_{t\to\infty} y_v(t)$ exists and is finite (Theorem 4.1 of [59]). Now, recall that the ROAS satisfies $O^r_\infty \subset \bar{O}_\infty$ and that $\bar{O}_\infty$ contains the shrunk steady-state constraint, $H_0 y_{v_{ss}} \in (1-\epsilon)Y$. Because we know that $H_0 = 1$, we have that $y_{v_{ss}} \in (1-\epsilon)Y$, which in turn implies that $y_{v_{ss}} \in Y$. Next, by assumption A.5.1.2, we can compute the input to the nonlinear plant as $v(t) = \zeta^{-1}(y_v(t))$, which converges to $v_{ss} = \zeta^{-1}(y_{v_{ss}})$. Hence by assumption A.5.1.1, the output of system (5.1), $y_{non}(t)$, converges to $y_{non_{ss}} = \zeta(v_{ss}) = (\zeta \circ \zeta^{-1})(y_{v_{ss}}) = y_{v_{ss}}$, which finally implies that $y_{non_{ss}} \in Y$, as desired. □

Next, we study the ROAS and analyze TR-RG's ability to enforce the constraints during transients.

5.2.2 Transient Constraint Enforcement

As mentioned previously, system (5.2) is obtained by linearizing (5.1) around an operating point. Hence, for operating conditions far from the linearization point, there may be a significant plant model mismatch, and therefore the predictions of the constrained output based on the linearized model may not be accurate. However, since the linear model and the nonlinear system are matched at steady-state (thanks to the steady-state map ζ), we know that the mismatch only manifests during transients. For this reason, we propose a novel scheme (i.e., data-driven construction of ROAS) that combines linear system predictions with a dynamic margin to capture the transient mismatch. Using the ROAS, the TR-RG can enforce the constraints both at steady-state and during transients.

This idea is formally presented as follows. Recall that we want to impose $\underline{s} \leq y_{non}(t) \leq \bar{s}$ for all $t \in \mathbb{Z}_+$. To do so, we introduce the functions $\underline{\mathrm{G}}(\cdot)$ and $\bar{G}(\cdot)$ such that the following inequalities hold:

$$y(t) - \underline{\mathrm{G}}(x(t), y_v(t)) \leq y_{non}(t) \leq y(t) + \bar{G}(x(t), y_v(t)) \tag{5.5}$$

where $y(t)$ is the output of (5.2) and $y_{non}(t)$ is the output of (5.1). The functions $\underline{\mathrm{G}}$ and $\bar{G}$ are positive and have additional properties that will be defined later, together with some examples. These functions are intended to capture the difference between the outputs of the linear model and the nonlinear system. Now, to enforce $\underline{s} \leq y_{non}(t) \leq \bar{s}$, we enforce the following constraints using the TR-RG:

$$\begin{aligned} \underline{s} \leq & y(t) - \underline{\mathrm{G}}(x(t), y_v(t)), \\ & y(t) + \bar{G}(x(t), y_v(t)) \leq \bar{s} \end{aligned} \tag{5.6}$$

From (5.5) and (5.6), it follows that:

$$\begin{aligned} \underline{\mathrm{s}} \leq & y(t) - \underline{\mathrm{G}}(x(t), y_v(t)) \leq y_{non}(t), \\ & y_{non}(t) \leq y(t) + \bar{G}(x(t), y_v(t)) \leq \bar{s} \end{aligned} \tag{5.7}$$

which implies that $\underline{\mathrm{s}} \leq y_{non}(t) \leq \bar{s}$ as desired. Based on (5.7), we define the ROAS as follows, where similar to Assumption A.2.1.2, we have assumed that, for the construction of ROAS, $y_v(t) = y_{v_0}$ is held constant:

$$O_\infty^r := \left\{ \begin{array}{l} (x_0, y_{v_0}) \in \mathbb{R}^{n+1} : y(t) + \bar{G}(x(t), y_{v_0}) \leq \bar{s} \\ -y(t) + \underline{\mathrm{G}}(x(t), y_{v_0}) \leq -\underline{\mathrm{s}}, \forall t \in \mathbb{Z}_+ \end{array} \right\} \tag{5.8}$$

where $x(t) = A^t x_0 + (I-A)^{-1}(I-A^t)By_v$ and $y(t) = Cx(t) + Dy_v(t)$, are the predicted state and output of (5.2). Note that definition (5.8) represents the set of all initial conditions such that the predictions of the linear model (5.2) plus a dynamic margin, and hence the nonlinear system output, are within the constraints for all times. The properties of this ROAS are analyzed in Section 5.2.3 and the numerical properties and computational aspects of it are discussed in Section 5.4.

We now elaborate on the properties and tuning of the functions $\underline{\mathrm{G}}$ and $\bar{G}$. We endow these functions with the following structure:

$$\begin{aligned} \bar{G}(x, y_v) &= \gamma_1 g(x, y_v) + \gamma_2 \\ \underline{\mathrm{G}}(x, y_v) &= \gamma_3 g(x, y_v) + \gamma_4 \end{aligned} \tag{5.9}$$

where $\gamma_j, j = 1, \ldots, 4$ are positive scalars to be tuned, and the function $g(x, y_v)$ satisfies the following properties: positive ($g(\cdot) > 0$), convex, and $g(x, y_v) = 0$ if and only if x is the equilibrium corresponding to y_v, i.e., $x = (I - A)^{-1}By_v$. We will explain in Section 5.2.3

why these conditions are required. Some specific choices are as follows:

$$g(x, y_v) = ||x - x_{ss}||_P \tag{5.10a}$$

$$g(x, y_v) = ||x - x_{ss}||_P^2 \tag{5.10b}$$

$$g(x, y_v) = ||\Delta x||_P \tag{5.10c}$$

where $x_{ss} := (I - A)^{-1}By_v$ is the steady-state value of x for the constant input y_v, and $\Delta x := (A - I)x + By_v$ represents the difference in the state at two successive timesteps. The matrix P is a positive definite matrix whose diagonal elements are computed based on the expected range of the states (more on this in Section 5.4). Functions (5.10a) and (5.10b) are similar with the difference that, in terms of implementation, (5.10b) offers a higher numerical robustness as compared to (5.10a), due to the absence of the square root. On the other hand, (5.10b) may introduce issues with order of magnitude (i.e., calculating large values due to squaring the vector elements). The function (5.10c) offers an alternative that uses the difference between states in consecutive timesteps rather than the difference with respect to the steady-state as in (5.10a) and (5.10b). This implies that under sudden input variations, (5.10c) may be more conservative, which might lead to computing bigger values for the parameters γ_1 to γ_4 (see below for the tuning process). Note that in all cases, γ_2 and γ_4 generally represent the level of uncertainty/mismatch at steady-state (if any), while γ_1 and γ_3 represent the level of uncertainty during transients.

Once the function $g(\cdot)$ is defined, we proceed to tune the parameters γ_j in (5.8) based on data collected from the nonlinear system. To do so, we excite both the linear model and the nonlinear system with the same sequence of inputs (y_v) and gather their outputs (y and y_{non}). These data are used to solve offline an optimization problem to compute the parameters γ_i. The complete process is outlined as follows:

1. Define a sequence of inputs, for example steps and/or ramps, to capture the nonlinear

system and linear model responses in the region of interest. The definition of the inputs may depend on the nature of the application and the range of frequencies that we want to analyze (see Remark 5.2.2 for clarification, and Section 5.5 for an illustration).

2. Apply the pre-defined input sequence to (5.1) and (5.2) to capture the systems' response in the operating region of interest.

3. Collect all the data points corresponding to: y_{non}, y, x, and y_v (see (5.1) and (5.2)). For the tuning process, y_v is directly controlled.

4. For all data points, solve the following optimization problem:

$$\begin{aligned} \underset{\gamma_j}{\text{minimize}} \quad & \sum_{j=1}^{4} \rho_j \gamma_j \\ \text{s.t.} \quad & y_{non}(k) \leq y(k) + \gamma_1\, g(x(k), y_v(k)) + \gamma_2, \\ & y(k) - \gamma_3\, g(x(k), y_v(k)) - \gamma_4 \leq y_{non}(k) \\ & \gamma_2 < \bar{s}, -\gamma_4 > \underline{s}, \\ & \gamma_j \geq 0, j = 1, \ldots, 4 \end{aligned} \tag{5.11}$$

where $\rho_j > 0$ are weighting factors. Specifically, ρ_1 and ρ_3 penalize γ_1 and γ_3 respectively, and ρ_2 and ρ_4 penalize γ_2 and γ_4. As a rule of thumb, since the steady-state mapping of the system is assumed to be accurately known, ρ_2 and ρ_4 should be selected to be large to ensure that γ_2 and γ_4 are small.

Remark 5.2.2. *Notice that once the TR-RG is introduced in the loop (see Fig. 5.1), the input trajectories of the nonlinear system are not the same as the inputs that were used during the data collection process. More specifically, the signals computed by the TR-RG may be slower than the ones applied during data collection. For these reasons, the tuning*

process must consider fast inputs that sufficiently excite the nonlinear system and capture its behavior in the operating region of interest.

Remark 5.2.3. *Since the systems* (5.1) *and* (5.2) *satisfy A.5.1.1 and A.2.1.1 respectively, the signals* $y_{non}(t)$ *and* $y(t)$ *are bounded, which implies that the above optimization problem always has a solution. It is, however, possible that, at the optimal solution,* γ_i *are large. If this is the case, the ROAS will be small, leading to a conservative transient response. To remedy this, one can use a different function* $g(\cdot)$*.*

With the ROAS defined and tuned above, the TR-RG can enforce the constraints during both steady-state and transients, as long as the nonlinear system operates in the operating region in which data was collected. The TR-RG has two additional, important characteristics. First, the formulation is recursively feasible, i.e., there exists a feasible solution to (5.4) at every timestep. This follows from the fact that O_∞^r is positively invariant (Proposition 5.2.1 in Section 5.2.3), which implies that if $(x(t), y_v(t)) \in O_\infty^r$ then $(x(t+1), y_v(t)) \in O_\infty^r$. This, together with (5.3) and (5.4), imply that $\kappa = 0$ (i.e., $y_v(t+1) = y_v(t)$) is always a feasible solution to the optimization problem. Second, TR-RG preserve stability of the closed-loop system. This can be shown by noticing from (5.3) that for a constant $r(t)$ and, hence, $y_r(t)$, $y_v(t)$ forms a monotonic sequence over a compact set, which implies that $y_v(t)$ must converge to a constant.

A deeper analysis of the properties of the ROAS is presented in the following section.

5.2.3 Analysis of the Robust Output Admissible Set (ROAS)

As mentioned previously, the construction of ROAS relies on the assumption that $y_v(t)$ is held constant, that is $y_v(t+1) = y_v(t), \forall t \in \mathbb{Z}_+$. To ease the discussion that follows, we

simplify the notation by augmenting system (5.2) with the above constant input dynamics:

$$\begin{aligned} \bar{x}(t+1) &= \bar{A}\bar{x}(t) \\ y(t) &= \bar{C}\bar{x}(t) \end{aligned} \tag{5.12}$$

where $\bar{x} = \begin{bmatrix} x \\ y_v \end{bmatrix}$, $\bar{A} = \begin{bmatrix} A & B \\ 0 & 1 \end{bmatrix}$, and $\bar{C} = \begin{bmatrix} C & D \end{bmatrix}$. Based on (5.12) and (5.8) we have that O_∞^r is:

$$O_\infty^r := \left\{ \begin{array}{l} \bar{x} \in \mathbb{R}^{n+1} : \bar{C}\bar{A}^t\bar{x} + \gamma_1 g(\bar{A}^t\bar{x}) + \gamma_2 \leq \bar{s} \\ -\bar{C}\bar{A}^t\bar{x} + \gamma_3 g(\bar{A}^t\bar{x}) + \gamma_4 \leq -\underline{s}, \forall t \in \mathbb{Z}_+ \end{array} \right\} \tag{5.13}$$

Next, we study the conditions under which O_∞^r satisfies the following properties (similar to O_∞): $0 \in \text{int}(O_\infty^r)$, compactness, convexity, positive invariance, and finite determinism. We show that for O_∞^r to satisfy these characteristics, $g(\cdot)$ must satisfy the properties that were previously mentioned, namely: *convexity, positiveness, and being zero at all equilibria.* These properties of O_∞^r are important to ensure an effective implementation of TR-RG. Note that convexity also implies continuity.

Theorem 5.2.2. *Assume that* $g(0) = 0$ *and* γ_i *in* (5.9) *are tuned using* (5.11)*. Then,* $0 \in \text{int}(O_\infty^r)$.

Proof. Recall from (5.9) and the definition of the set Y that $0 < \gamma_2 < \bar{s}$ and $\underline{s} < -\gamma_4 < 0$. Define the positive parameter $r = \min(\bar{s} - \gamma_2, |\underline{s} + \gamma_4|)$, from which we have that $\mathcal{B}_r(0) \subset Y$. This allows us to define the set Ω, which satisfies, $\Omega \subset O_\infty^r$ and is given by:

$$\Omega := \left\{ \bar{x} : \begin{array}{c} \bar{C}\bar{A}^t\bar{x} + \gamma_1 g(\bar{A}^t\bar{x}) \leq r \\ -\bar{C}\bar{A}^t\bar{x} + \gamma_3 g(\bar{A}^t\bar{x}) \leq r \end{array}, \forall t \in \mathbb{Z}_+ \right\} \tag{5.14}$$

Finally, from the condition $g(0) = 0$ and (5.14), we have that there is a $\beta > 0$ such that

$\mathcal{B}_\beta(0) \subset \Omega \subset O^r_\infty$, which implies that $0 \in \text{int}(O^r_\infty)$. □

Note from the result of Theorem 5.2.2 that O^r_∞ is non-empty.

Theorem 5.2.3. *Assume that the pair $(\bar{C}, \bar{A})$ is observable and the assumptions in Theorem 5.2.2 hold, then O^r_∞ is compact.*

Proof. To show that O^r_∞ is compact, we show that it is closed and bounded. The former can be proved directly from definition (5.13) and Theorem 5.2.2, because O^r_∞ is non-empty and is the intersection of closed sets, hence O^r_∞ is closed. To show that O^r_∞ is bounded, we can use existing results from [12]. Consider the linear system (5.12), with the MAS given by:

$$O^s_\infty := \{\bar{x} \in \mathbb{R}^{n+1} : \bar{C}\bar{A}^t\bar{x}(t) \in Y, \forall t \in \mathbb{Z}_+\} \tag{5.15}$$

Since the pair $(\bar{C}, \bar{A})$ is observable, we know that O^s_∞ is compact (Theorem 2.1 [12]). We need the following claim to complete the proof.

Claim: Since $g(\cdot)$ is positive, $O^r_\infty \subseteq O^s_\infty$.

To show this, select an arbitrary $\bar{x}$ in O^r_∞. Being in O^r_∞ implies that for all $t \geq 0$:

$$\begin{aligned} \bar{C}\bar{A}^t\bar{x} + \gamma_1 g(\bar{A}^t\bar{x}) + \gamma_2 \leq \bar{s} \Rightarrow \quad & \bar{C}\bar{A}^t\bar{x} \leq \bar{s} - \gamma_1 g(\bar{A}^t\bar{x}) - \gamma_2 \\ -\bar{C}\bar{A}^t\bar{x} + \gamma_3 g(\bar{A}^t\bar{x}) + \gamma_4 \leq -\underline{s} \Rightarrow \quad & \underline{s} + \gamma_3 g(\bar{A}^t\bar{x}) + \gamma_4 \leq \bar{C}\bar{A}^t\bar{x} \end{aligned}$$

Then we have:

$$\underline{s} \leq \underline{s} + \gamma_3 g(\bar{A}^t\bar{x}) + \gamma_4 \leq \bar{C}\bar{A}^t\bar{x} \leq \bar{s} - \gamma_1 g(\bar{A}^t\bar{x}) - \gamma_2 \leq \bar{s} \tag{5.16}$$

This means that $\bar{x} \in O^s_\infty$ as well, which implies that $O^r_\infty \subseteq O^s_\infty$. Thanks to this result we have that since O^s_∞ is bounded, O^r_∞ is bounded as well, and hence compact. □

Remark 5.2.4. *Note that, for values of $\gamma_j, j = 1, \ldots, 4$, different from zero, O^r_∞ is an output admissible set for* (5.12)*, but it is not maximal. This is expected, since our goal is to*

impose constraints on the nonlinear system by considering predictions of the linear system plus a dynamic margin.

Theorem 5.2.4. *Assume that the function g is convex, then O_∞^r is convex.*

Proof. Consider two arbitrary $\bar{x}_1$ and $\bar{x}_2$ that belong to O_∞^r. By (5.13), $\bar{x}_1$ and $\bar{x}_2$ satisfy for all $t \in \mathbb{Z}_+$:

$$\bar{C}\bar{A}^t\bar{x}_i + \gamma_1 g(\bar{A}^t\bar{x}_i) + \gamma_2 \leq \bar{s} \tag{5.17a}$$

$$-\bar{C}\bar{A}^t\bar{x}_i + \gamma_3 g(\bar{A}^t\bar{x}_i) + \gamma_4 \leq -\underline{s} \tag{5.17b}$$

where $i = 1, 2$. To prove convexity of O_∞^r, it suffices to show that $\bar{x}_3 := \lambda\bar{x}_1 + (1-\lambda)\bar{x}_2$ also belongs to O_∞^r, i.e., (5.17a) and (5.17b) hold for $\bar{x}_3$ (i.e., $i = 3$). Below, we only derive the result for (5.17a), as the derivation for (5.17b) is similar. We start with the left hand side of (5.17a) with $i = 3$:

$$\begin{aligned}
&\bar{C}\bar{A}^t(\lambda\bar{x}_1 + (1-\lambda)\bar{x}_2) + \gamma_1 g\left(\bar{A}^t(\lambda\bar{x}_1 + (1-\lambda)\bar{x}_2)\right) + \gamma_2 \\
&\leq \lambda\left(\bar{C}\bar{A}^t\bar{x}_1 + \gamma_1 g(\bar{A}^t\bar{x}_1) + \gamma_2\right) + \\
&\quad (1-\lambda)\left(\bar{C}\bar{A}^t\bar{x}_2 + \gamma_1 g(\bar{A}^t\bar{x}_2) + \gamma_2\right) \\
&\leq \lambda\bar{s} + (1-\lambda)\bar{s} = \bar{s}
\end{aligned} \tag{5.18}$$

where the first step above follows from the convexity of $g(\cdot)$:

$$g(\bar{A}^t(\lambda\bar{x}_1 + (1-\lambda)\bar{x}_2)) \leq \lambda g(\bar{A}^t\bar{x}_1) + (1-\lambda)g(\bar{A}^t\bar{x}_2)$$

This completes the proof. □

We now discuss the positive invariance of O_∞^r with respect to the dynamics of linear

system (5.12). The positive invariance condition states:

$$\bar{x} \in O_\infty^r \Rightarrow \bar{A}\bar{x} \in O_\infty^r \tag{5.19}$$

Note that we cannot formulate the positive invariance condition with respect to the original nonlinear system (5.1). This is because the ROAS is constructed using the linear system (5.12), whose states are not the same as the states of (5.1). Nevertheless, since the output of (5.1) is bounded by that of (5.12) plus a margin, the TR-RG can ensure that the constraints will be satisfied.

Proposition 5.2.1. *O_∞^r is positively invariant.*

The proof follows simply from the definition of O_∞^r. It is therefore omitted for brevity.

We now evaluate the conditions for which O_∞^r is finitely determined. To do so, we introduce the sets O_t^r as:

$$O_t^r := \left\{ \begin{array}{l} \bar{x} \in \mathbb{R}^{n+1} : \bar{C}\bar{A}^i\bar{x} + \gamma_1 g(\bar{A}^i\bar{x}) + \gamma_2 \leq \bar{s} \\ -\bar{C}\bar{A}^i\bar{x} + \gamma_3 g(\bar{A}^i\bar{x}) + \gamma_4 \leq -\underline{s}, i = 0, \ldots, t \end{array} \right\} \tag{5.20}$$

From (5.20), the following condition holds $\forall t_1, t_2 \in \mathbb{Z}_+$, $t_2 > t_1$:

$$O_\infty^r \subset O_{t_2}^r \subset O_{t_1}^r \tag{5.21}$$

Condition (5.21) is used to prove finite determinism of the ROAS. We are now ready to formally define finite determinism, similar to [12]: the set O_∞^r is finitely determined if there exists a t^* such that $O_\infty^r = O_{t^*}^r$.

Theorem 5.2.5. *O_∞^r is finitely determined iff there exists a t such that, $O_t^r = O_{t+1}^r$.*

Proof. **Sufficiency:** Suppose O^r_∞ is finitely determined. Then, by definition, there exists a t^* such that $O^r_\infty = O^r_{t^*}$. Combining this with (5.21) with $t_1 = t^*$ and $t_2 = t^* + 1$, we obtain $O^r_{t^*} = O^r_\infty \subset O^r_{t^*+1} \subset O^r_{t^*}$, which implies that $O^r_{t^*+1} = O^r_{t^*}$.

Necessity: Suppose $O^r_t = O^r_{t+1}$. We must show that there is a $t^* \in \mathbb{Z}_+$, such that $O^r_\infty = O^r_{t^*}$. To do so, we first claim that $O^r_{t+1} = O^r_{t+2}$. To prove this claim, select an arbitrary $\bar{x} \in O^r_{t+1}$. By definition of O^r_{t+1} we have:

$$\begin{aligned} \bar{C}\bar{A}^i\bar{x} + \gamma_1 g(\bar{A}^i\bar{x}) + \gamma_2 &\leq \bar{s} \\ -\bar{C}\bar{A}^i\bar{x} + \gamma_3 g(\bar{A}^i\bar{x}) + \gamma_4 &\leq -\underline{s}, \quad i = 0, \ldots, t+1 \end{aligned}$$

To proceed, we express $\bar{A}^i\bar{x}$ as $\bar{A}^{i-1}(\bar{A}x)$ and perform a change of variable $l = i - 1$ to obtain:

$$\begin{aligned} \bar{C}\bar{A}^l(\bar{A}\bar{x}) + \gamma_1 g(\bar{A}^l(\bar{A}\bar{x})) + \gamma_2 &\leq \bar{s} \\ -\bar{C}\bar{A}^l(\bar{A}\bar{x}) + \gamma_3 g(\bar{A}^l(\bar{A}\bar{x})) + \gamma_4 &\leq -\underline{s}, \quad l = -1, \ldots, t \end{aligned}$$

This means that $\bar{A}\bar{x} \in O^r_t$. This and the fact that $O^r_t = O^r_{t+1}$ leads to $\bar{A}\bar{x} \in O^r_{t+1}$, which implies:

$$\begin{aligned} \bar{C}\bar{A}^i(\bar{A}\bar{x}) + \gamma_1 g(\bar{A}^i(\bar{A}\bar{x})) + \gamma_2 &\leq \bar{s} \\ -\bar{C}\bar{A}^i(\bar{A}\bar{x}) + \gamma_3 g(\bar{A}^i(\bar{A}\bar{x})) + \gamma_4 &\leq -\underline{s}, \quad i = 0, \ldots, t+1 \end{aligned}$$

$$\begin{aligned} \Rightarrow \bar{C}\bar{A}^l\bar{x} + \gamma_1 g(\bar{A}^l\bar{x}) + \gamma_2 &\leq \bar{s} \\ -\bar{C}\bar{A}^l\bar{x} + \gamma_3 g(\bar{A}^l\bar{x}) + \gamma_4 &\leq -\underline{s}, \quad l = 1, \ldots, t+2 \end{aligned}$$

where we have performed the change of variable $l = i + 1$. Since we know that $\bar{x} \in O^r_{t+1}$, we know that the last two inequalities hold for $l = 0$ as well. We thus conclude that $\bar{x} \in O^r_{t+2}$. Since $\bar{x}$ was arbitrarily chosen in O^r_{t+1}, we conclude that $O^r_{t+1} \subset O^r_{t+2}$. This, together with (5.21) finally imply that $O^r_{t+1} = O^r_{t+2}$. We can apply the same logic to show that $O^r_{t+2} = O^r_{t+3}$ and so on, and by induction we have that $O^r_\infty = O^r_t$. □

In practice, it is not possible to know a priori if O^r_∞ is finitely determined. In fact, O^r_∞ may not be. We now show that, similar to the results from standard MAS theory [12], we can obtain a finitely-determined inner approximation of O^r_∞ by introducing a tightened version of the steady-state constraint in the definition of O^r_∞. To explain, we modify (5.8) and (5.9) to:

$$\bar{O}^r_\infty := \left\{ \begin{array}{l} (x_0, y_{v_0}) : y_{ss} + \gamma_2 \leq (1-\epsilon)\bar{s}, \\ -y_{ss} + \gamma_4 \leq -(1-\epsilon)\underline{s}, \\ y(t) + \gamma_1 g(x(t), y_{v_0}) + \gamma_2 \leq \bar{s}, \\ -y(t) + \gamma_3 g(x(t), y_{v_0}) + \gamma_4 \leq -\underline{s}, \forall t \in \mathbb{Z}_+ \end{array} \right\} \tag{5.22}$$

where $0 < \epsilon \ll 1$, $y_{ss} = H_0 y_{v_0}$, $H_0 = C(I-A)^{-1}B + D$ is the DC gain of (5.2), and $x(t)$ and $y(t)$ are the state and output of system (5.2) starting from the initial condition x_0 and constant input y_{v_0}, namely $x(t) = A^t x_0 + (I - A^t)(I-A)^{-1}By_{v_0}$ and $y(t) = Cx(t) + Dy_{v_0}$. Note that the first two inequalities in (5.22) are the tightened steady-state constraints, where we have used the fact that $g(\cdot) = 0$ at steady-state. Note also that we have used the (x, y_v) coordinates instead of the augmented $\bar{x}$ coordinates to define $\bar{O}^r_\infty$, which will simplify the discussion below. We now prove that $\bar{O}^r_\infty$ is finitely determined.

Theorem 5.2.6. *Assume that the function $g(\cdot) = 0$ at the equilibrium and is continuous, then $\bar{O}^r_\infty$ in* (5.22) *is finitely determined.*

Proof. Select an arbitrary point $(x_0, y_{v_0}) \in \bar{O}^r_\infty$. We know that this point satisfies the following conditions:

$$\begin{array}{l} H_0 y_{v_0} + \gamma_2 \leq (1-\epsilon)\bar{s} \\ -H_0 y_{v_0} + \gamma_4 \leq -(1-\epsilon)\underline{s} \end{array} \tag{5.23}$$

where H_0 is the DC gain of (5.2). Let us denote the left hand sides of the third and fourth

inequalities in (5.22) by $l_1(t, x_0, y_{v_0})$ and $l_2(t, x_0, y_{v_0})$, respectively:

$$l_1(t, x_0, y_{v_0}) = y(t) + \gamma_1 g(x(t), y_{v_0}) + \gamma_2$$
$$l_2(t, x_0, y_{v_0}) = -y(t) + \gamma_3 g(x(t), y_{v_0}) + \gamma_4$$

By stability of (5.2) and continuity of l_1 and l_2, we have that $l_1(t, x_0, y_{v_0}) \to l_1(\infty, x_0, y_{v_0})$ and $l_2(t, x_0, y_{v_0}) \to l_2(\infty, x_0, y_{v_0})$. Furthermore, by (5.23), $l_1(\infty, x_0, y_{v_0}) \leq (1-\epsilon)\bar{s}$ and $l_2(\infty, x_0, y_{v_0}) \leq -(1-\epsilon)\underline{s}$. Therefore, by convergence of l_1 and l_2, we have that there exists $k^* > 0$ such that $\forall k > k^*$, $l_1(k, x_0, y_{v_0}) \leq \bar{s}$ and $l_2(k, x_0, y_{v_0}) \leq -\underline{s}$. This implies that all inequalities after k^* are already covered by (5.23) and are redundant. Therefore (5.22) is finitely determined. □

From this point forward, we will use $\bar{O}^r_\infty$ in our analyses.

5.3 Extensions of TR-RG

In this section, we study two practical extensions of the TR-RG formulation presented in Section 5.2. First is the extension to multi-output nonlinear systems, and second is the extension to systems whose steady-state map depends on a slowly-varying parameter.

5.3.1 TR-RG for Multiple Outputs

Consider again system (5.1), but now consider the more general case of $p > 1$, i.e., multiple constrained outputs. We assume that the steady-state mappings ζ_i (from the input v to the output y_{non_i}) are accurately available and satisfy Assumption A.5.1.2. To handle this case, we modify the TR-RG block diagram from the one in Fig. 5.1 to the one in Fig. 5.3. The basic idea is to apply the design methodology in Section 5.2 to each output separately,

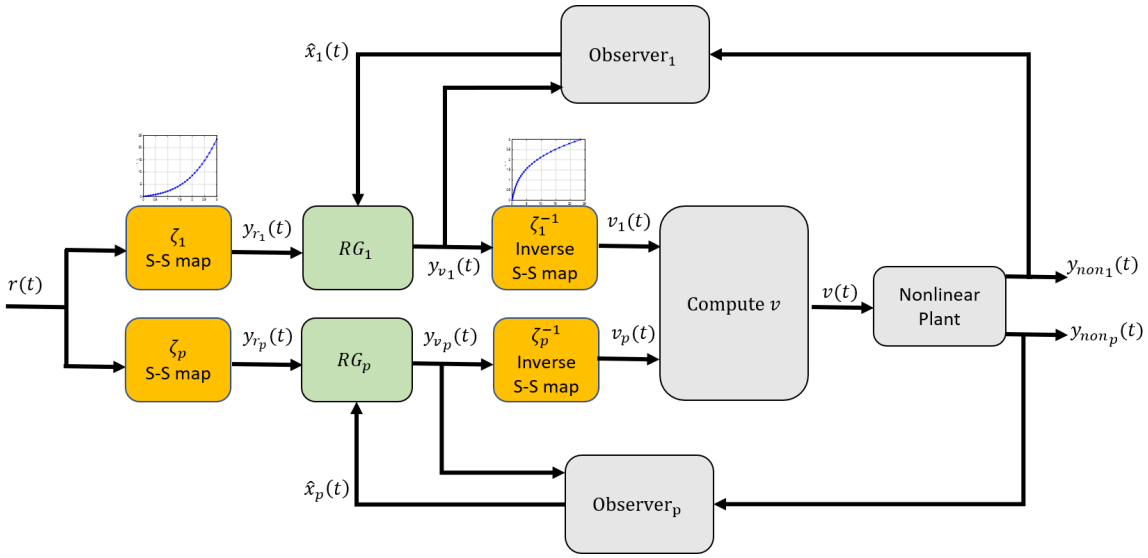

Figure 5.3: TR-RG for multiple outputs

which leads to the computation of p different inputs $v_i(t)$, $i = 1, \ldots, p$, in Fig. 5.3 (note that the subscript i here does not refer to the i-th element of v). The $v_i(t)$'s are then fused together to compute a $v(t)$ that satisfies the constraints on the nonlinear system for all times. Each RG in this formulation is implemented using a ROAS that is computed using a linearized model of the nonlinear system. Similar to Section 5.2, each linear model reflects the combined dynamics of the nonlinear plant together with the inverse mapping (ζ_i^{-1}); therefore, they each have DC gain equal to 1. For clarity of presentation, we use the same A and B notation as before, but these do not necessarily represent the same matrices. These linear systems are described by:

$$\begin{aligned} x_i(t+1) &= Ax_i(t) + By_{v_i}(t) \\ y_i(t) &= C_i x_i(t) + D_i y_{v_i}(t), \end{aligned} \tag{5.24}$$

where we have assumed the linear models have the same A and B matrices (i.e., the same internal dynamics), but different output matrices. The ROAS's, denoted by $\bar{O}^r_\infty$, are

computed offline based on (5.24) using the data-driven approach described in Section 5.2. In real-time, the RGs are implemented by solving the following optimization problem at every timestep:

$$\begin{aligned} & \underset{\kappa_i \in [0,1]}{\text{maximize}} \quad \kappa_{\text{i}} \\ \text{s.t.} \quad & (x_i(t), y_{v_i}(t)) \in \bar{O}^r_\infty \\ & y_{v_i}(t) = y_{v_i}(t-1) + \kappa_i (y_{r_i}(t) - y_{v_i}(t-1)) \end{aligned} \tag{5.25}$$

Notice that because we treat each output separately, the TR-RG scheme now requires p different observers, each designed based on the i-th linear model. The observers satisfy the observations mentioned in Remark 5.2.1. Finally, the fusion operation ("Compute v" block in Fig. 5.3) computes $v(t)$ as follows:

$$v = \underset{\theta \in \{v_1, \ldots, v_p\}}{\text{argmin}} \ |\theta| \tag{5.26}$$

Note that (5.26) selects among the v_i's the one that is closest to zero. The following theorem investigates the constraint management properties of this scheme at steady-state.

Theorem 5.3.1. *Assume that A.5.1.2 holds and the reference $r(t)$ is held constant. By implementing TR-RG as presented in Fig. 5.3 the constraints of the nonlinear system* (5.1) *are enforced at steady-state.*

Proof. We first introduce the following set of steady-state admissible inputs:

$$\Upsilon := \cap_{i=1}^{p} V_i \tag{5.27}$$

where $V_i = \{v : \zeta_i(v) \in (1-\epsilon) Y_i\}$. By assumption A.5.1.2 we have that each set V_i is convex and compact with $0 \in \text{int}(V_i)$, hence the set Υ is also convex and compact with $0 \in \text{int}(\Upsilon)$. Now, by a similar arguments as in Theorem 5.2.1, we know that each $y_{v_i}(t)$ converges to a constant value $y_{v_{ssi}}$, which satisfies $y_{v_{ssi}} \in (1-\epsilon) Y_i$. From Fig. 5.3, we know

that $v_i(t) = \zeta_i^{-1}(y_{v_i}(t))$, which converges to $v_{ss_i} = \zeta_i^{-1}(y_{v_{ssi}})$. Since these inputs are fused by operation (5.26), we get a $v_{ss_i} \in \Upsilon$, which finally implies that $y_{non_i} \in Y_i, i = 1, \ldots, p$ at steady-state. □

The TR-RG for $p > 1$ uses p different ROAS that are computed based on data collected from the nonlinear plant and (5.24) around an operating region of interest. Tuning each ROAS using this data within an optimization framework similar to (5.11), we can ensure that the constraints are enforced during transients as long as the system operates in the same region of interest.

Remark 5.3.1. *As mentioned previously,* (5.24) *assumes that the internal dynamics (i.e., the A and B matrices) of all p linear systems are the same. However, this assumption is not necessary. For example, it may be advantageous to use different dynamics for each linear system by linearizing the nonlinear system around each constraint. This will ensure that the linear models reflect the dynamics of the nonlinear plant more accurately in the vicinity of the constraints, thereby leading to less conservative ROAS's. Since the treatment of this case is similar, we will not explore it further for the sake of brevity.*

5.3.2 TR-RG for Systems with Fast and Slow Dynamics

To conclude our analysis, this section extends TR-RG to nonlinear systems whose steady-state mappings depend on parameters that vary slowly with time. As a practical example, consider the turbocharged gasoline engine presented in Section 5.1. Typically, the dynamics of the airpath control system (e.g., turbo speed, manifold pressures, and actuator dynamics) are much faster than the dynamics of the engine speed, which is slow due to the large engine inertia. Due to this separation of timescales, engine speed can be modeled as a

quasi-constant parameter (or a disturbance) to the airpath control system. In particular, the steady-state mapping (from the input to the constrained outputs) can be modeled to be a function of the (slowly-varying) engine speed.

Recall that the TR-RG presented in Section 5.2 uses a fixed steady-state characterization in the formulation (i.e., ζ and ζ^{-1} in Fig. 5.1 and ζ_i and ζ_i^{-1} in Fig. 5.3 are fixed). Therefore, implementing the TR-RG on a system whose steady-state mapping varies slowly with time will require large values of γ_1 to γ_4 to ensure that (5.5) and (5.9) are satisfied, which in turn leads to a conservative ROAS and loss of performance, even at steady-state. To remedy this, our solution is to make the steady-state mapping ζ and its inverse ζ^{-1} functions of the slowly-varying parameter (both during the tuning process and real-time implementation). To ensure that the TR-RG can be implemented despite this change, the following additional assumption is required.

A. 5.3.1. *We assume that for each y_{non_i}, the steady-state mapping, denoted as $\zeta_i(v, z)$, is available, where each ζ_i is continuous, satisfies $\zeta_i(0, 0) = 0$, and for a fixed value of z, each ζ_i is invertible.*

Typically, a system with a separation of slow and fast timescales is modeled by:

$$\begin{aligned} \lambda \dot{x}_{non} &= f_x(x_{non}, z, v) \\ \dot{z} &= f_z(x_{non}, z, v) \\ y_{non} &= h(x_{non}, z, v) \end{aligned} \tag{5.28}$$

where $\dot{z}$ represents the slow dynamics (engine speed in our example), $\dot{x}_{non}$ the fast dynamics (airpath dynamics in our example), and $0 < \lambda \ll 1$. A singular perturbation argument can be made to show that for sufficiently small λ, the output y_{non} can be approximated by $y_{non} \approx \zeta(v, z)$. Thus, the introduction of ζ^{-1} in Figs. 5.1 and 5.3 can accurately invert the steady-state characterization of the plant if λ is sufficiently small.

Note that the effects of plant/model mismatch can still be captured by γ_1 to γ_4 during the tuning process (Eq. (5.11)). As before, γ_2 and γ_4 capture the steady-state (or quasi-steady-state) modeling errors and γ_1 and γ_3 capture the transient errors. The constraints can therefore be satisfied using this technique if γ_j are tuned using data that were collected by operating the system in a domain of interest that includes varying values of z.

In the discussion that follows we address the implementation issues of TR-RG, and implement the proposed scheme on a nonlinear turbocharged model and on a vehicle equipped with a turbocharged gasoline engine.

5.4 Implementation of TR-RG and Computational Considerations

As discussed so far, in order to build $\bar{O}^r_\infty$ defined in (5.22), we need a suitable function $g(\cdot)$ and data to properly tune γ_i's and build a dynamic margin in $\bar{O}^r_\infty$. In practice, this dynamic margin must be large enough to enforce the constraints, but not excessive, since performance may deteriorate (i.e., the response might slow down when constraint violation is predicted). Notice from (5.11) that the parameters γ_1 to γ_4 are computed such that each nonlinear constrained output is bounded above and below for all times. This, however, may be too restrictive and lead, in some cases, to conservative results. Hence, to alleviate this issue in practical implementations, we propose a slightly modified tuning process based on a relaxed version of (5.7). The general idea is that for each constrained output, we consider the data points that satisfy $y_{non_i} \in \mathcal{B}_{\phi_i}(\bar{s}_i)$ or $y_{non_i} \in \mathcal{B}_{\beta_i}(\underline{s}_i)$, where $\phi_i > 0$ and $\beta_i > 0$ are desired thresholds. Based on these data points we tune the parameters γ_1 to γ_4 such that (5.7) is satisfied. The rationale behind this is that, if y_{non_i} is far from the constraints, then plant/model mismatch is not likely to cause any constraint violation. The complete

algorithm to tune the parameters in (5.22) is provided below for the sake of completeness:

Algorithm 2 General RG Tuning Process

1: Define a sequence of inputs which may be composed by steps and/or ramps. These may depend on the nature of the application and the operating region of interest (see Remark 5.2.2).
2: Define for each upper and lower constraint value (i.e., $\bar{s}_i$ and $\underline{s}_i$) the margins $\phi_i > 0$ and $\beta_i > 0$ respectively.
3: Apply the pre-defined input sequence to the nonlinear plant and to (5.24) and record their outputs.
4: For all data points that satisfy $y_{non_i}(k) > \bar{s}_i - \phi_i$ or $y_{non_i}(k) < \underline{s}_i + \beta_i$, run the optimization problem (5.11).

Remark 5.4.1. *Remark 5.2.3 applies to Algorithm 2 as well.*

Let us consider the same turbocharged model presented at the beginning of Section 5.1, but now implementing TR-RG. For this model, the state vector of the linear model is:

$$x = \begin{bmatrix} P_i, & P_b, & P_e, & N_{tc}, & W_{wg}, & x_{c_1}, & x_{c_2} \end{bmatrix}^\top,$$

which represents the intake pressure, boost pressure, exhaust pressure, turbo speed, waste-gate flow, and two controller states respectively. For this example $p = 1$ and the constraint is imposed on the turbo speed of 140 kRPM. We assume the following form for the function g in (5.22): $g(x, y_v) = ||x - x_{ss}||_P^2 = ||x - (I - A)^{-1}By_v||_P^2$, where the P-norm is defined as: $\|x\|_P := x^T P x$. For this example, the matrix P is tuned based on data to normalize the entries of x and avoid excessively large values: $P = \text{diag}(\frac{1}{r_i})$, where r_i is the total range of the values for the i-th state, x_i. Note that this function satisfies all the conditions studied in Section 5.2.3, that is, convex (and hence continuous), positive, and it is zero at the equilibrium.

Using this choice of $g(\cdot)$, we compute the ROAS as explained in Section 5.2. In this case since we have a constraint on the turbo speed, which is positive, we present how $\bar{O}^r_\infty$ is

computed. At each time step the rows of $\bar{O}^r_\infty$ as a function of the initial conditions (x_0, v) are given by:

$$\begin{aligned}
&t = 0; CAx_0 + (CB + D)v + \gamma_1(Ax_0 + (B - \Gamma)y_v)^\top P(Ax_0 + (B - \Gamma)y_v) + \gamma_2 \leq \bar{s} \\
&t = 1; CA^2x_0 + (CAB + CB + D)y_v + \gamma_1(A^2x_0 + (AB + B - \Gamma)y_v)^\top \\
&\qquad P(A^2x_0 + (AB + B - \Gamma)y_v) + \gamma_2 \leq \bar{s} \\
&\qquad \vdots \\
&\text{Any } t; H_x x_0 + H_v y_v + \gamma_1(H_{x_2}x_0 + (H_{v_2} - \Gamma)y_v)^\top P((H_{x_2}x_0 + (H_{v_2} - \Gamma)y_v)) + \gamma_2 \leq \bar{s}
\end{aligned} \tag{5.29}$$

where $H_x = CA^t, H_v = (C(I - A)^{-1}(I - A^t)B + D), H_{x_2} = A^t$, $H_{v_2} = (I - A)^{-1}(I - A^t)B$, and $\Gamma = (I - A)^{-1}B$. Since the liner system satisfies A.2.1.1, and $x(t) \to x_{ss}$ at steady-state, we have that the halfspace that corresponds to the steady-state is given by:

$$(C\Gamma + D)y_v \leq \bar{s} \tag{5.30}$$

By (5.29) and (5.30) we can build the ROAS to implement TR-RG in the turbocharged engine. A geometric comparison between the MAS (using the corresponding linear model) and the ROAS (highlighted in red) is presented in Fig. 5.4. Since the ROAS is high-dimensional, the figure shows a cross-section of $\bar{O}^r_\infty$ with all states other than x_4 equal to 0. This plot illustrates that the ROAS is a robustified version of the MAS to capture the mismatch between the linear model and the nonlinear plant.

The parameters in the ROAS are obtained after running Algorithm 2. For a pre-defined input sequence, $\phi = 1\,\text{kRPM}$, $\beta = 0\,\text{kRPM}$, $\rho_1 = 1$, and $\rho_2 = 1 \times 10^6$, we get: $\gamma_1 = 9379.76$ and $\gamma_2 = 0$. To illustrate the result of the tuning process see Fig. 5.5. This figure shows how the nonlinear system output (y_{non}) is bounded by the linear model output plus the dynamic margin ($y + \bar{G}$).

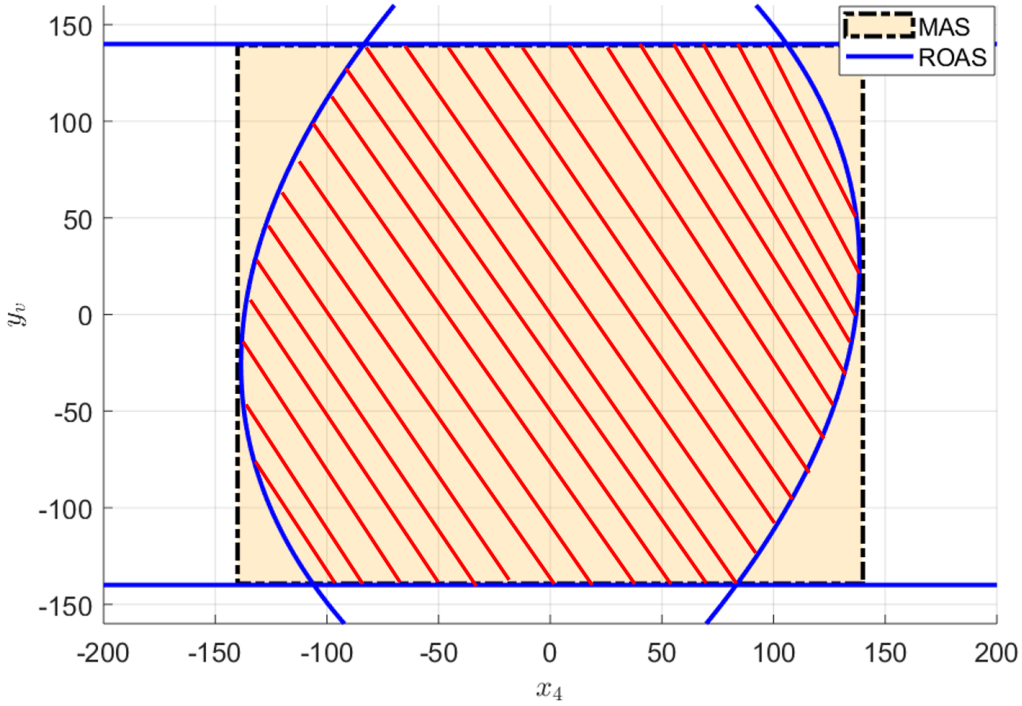

Figure 5.4: Cross section comparison between the MAS ($\bar{O}_\infty$) and ROAS ($\bar{O}^r_\infty$, highlighted in red).

Finally, in terms of real-time solution of the optimization problem in (5.4), notice that for this selection of $g(\cdot)$, each row of (5.22) has a linear and quadratic terms as a function of $y_v(t)$, and to find a solution for (5.4) we just need to solve in a simple FOR loop as many quadratic equations as rows of $\bar{O}^r_\infty$, and select the κ that satisfies all the inequalities in (5.22). This is similar to the algorithm presented in [52]. To elaborate, note that the matrices in (5.29) and (5.30) can all be pre-computed offline and stored in memory, thus avoiding real-time matrix multiplications and additions. Now, it can be seen that each row of the above inequalities can be placed into the form:

$$ay_v^2 + by_v + c \leq 0 \tag{5.31}$$

where a, b, and c are appropriate scalars that are composed of the initial condition, x_0, the system matrices, and constraint value. By substituting (5.3) into (5.31) and dropping the time arguments for notational simplicity, we get:

$$a_1\kappa^2 + a_2\kappa + a_3 \leq 0 \tag{5.32}$$

where $a_1 = a(y_r - y_v)^2$, $a_2 = 2ay_v(y_r - y_v) + b(y_r - y_v)$, and $a_3 = ay_v^2 + by_v + c$. To discuss the computation details of (5.32) are summarized in the Algorithm 3.

Algorithm 3 Compute κ

Let $\kappa = 1$, $j^* =$ number of rows of $\bar{O}^r_\infty$, $i = 1$, $0 < \alpha \ll 1 \times 10^{-3}$
while $i \leq j^*$ **do**
  if $a_3 > \alpha$ **then** ▷ a_3 must be strictly negative for recursive feasibility
    $\kappa = 0$
  else if $|a_1| < \alpha$ and $a_2 > 0$ **then** ▷ If first order inequality and sign of a_2 is good
    $\kappa = \min(\kappa, -a_3/a_2)$
  else if $|a_1| \geq \alpha$ and $a_2^2 - 4a_1a_3 \geq 0$ **then** ▷ if second order inequality and real roots exist
    $r_1 = \frac{-a_2+\sqrt{a_2^2-4a_1a_3}}{2a_1}$
    $r_2 = \frac{-a_2-\sqrt{a_2^2-4a_1a_3}}{2a_1}$
    if $a_1 \geq \alpha$ and $\max(r_1, r_2) \geq -\alpha$ **then** ▷ The parameter α is used to account for numerical tolerance
      $\kappa = \min(\kappa, max(r_1, r_2))$
    else if $a_1 \leq -\alpha$ and $\min(r_1, r_2) \geq -\alpha$ **then**
      $\kappa = \min(\kappa, min(r_1, r_2))$
    end if
  end if
  $i \to i + 1$
end while
$\kappa = \max(\kappa, 0)$

We simulate the nonlinear model of the engine for a single step in the desired air mass, W^*. Fig. 5.6 shows the response, both for the standard RG and the TR-RG. As can be seen, unlike standard RG, TR-RG enforces the constraint for all time. Finally, Fig. 5.7 shows that for this maneuver, the nonlinear system output is bounded by the linear plant plus the dynamic margin, and that they converge to the same value at steady-state, as expected.

5.5 Experimental Results

In this section we present the experimental implementation of the TR-RG using the same function $g(\cdot)$ defined in Section 5.4. The scheme is implemented in a Ford Explorer equipped

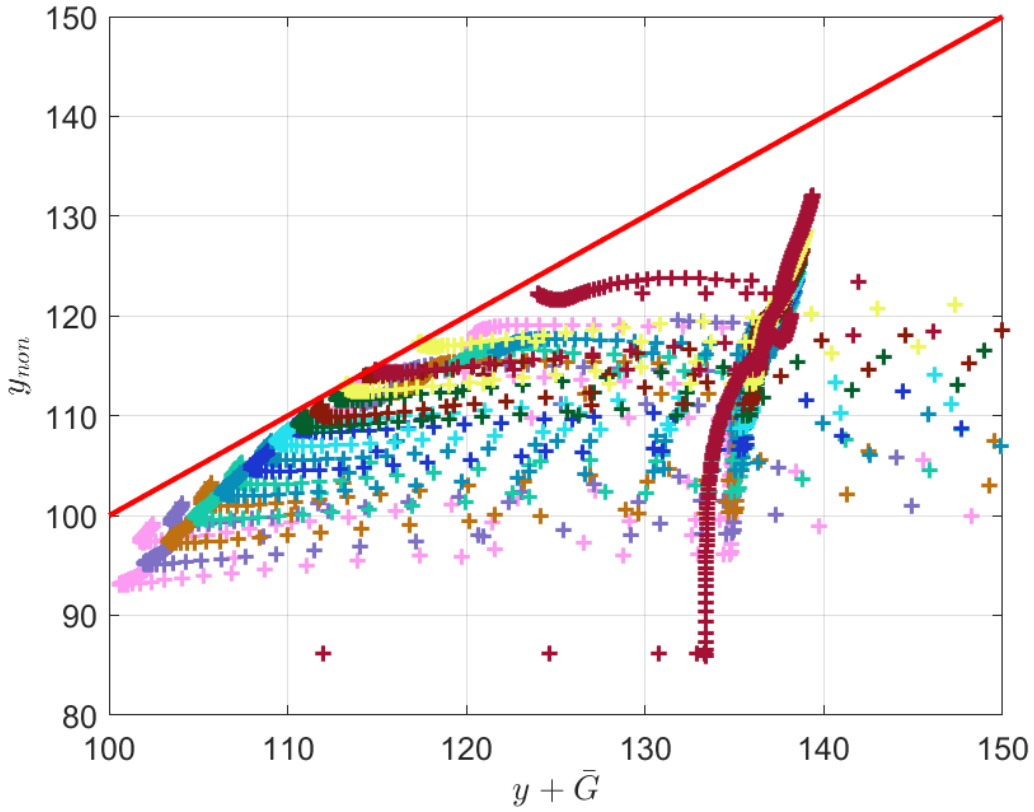

Figure 5.5: Tuning data. Comparison between the nonlinear system output (y_{non}) vs. linear system output plus dynamic margin ($y+\bar{G}$). The red line in the plot is the unity map, which shows that the bounds in (5.5) *have been enforced.*

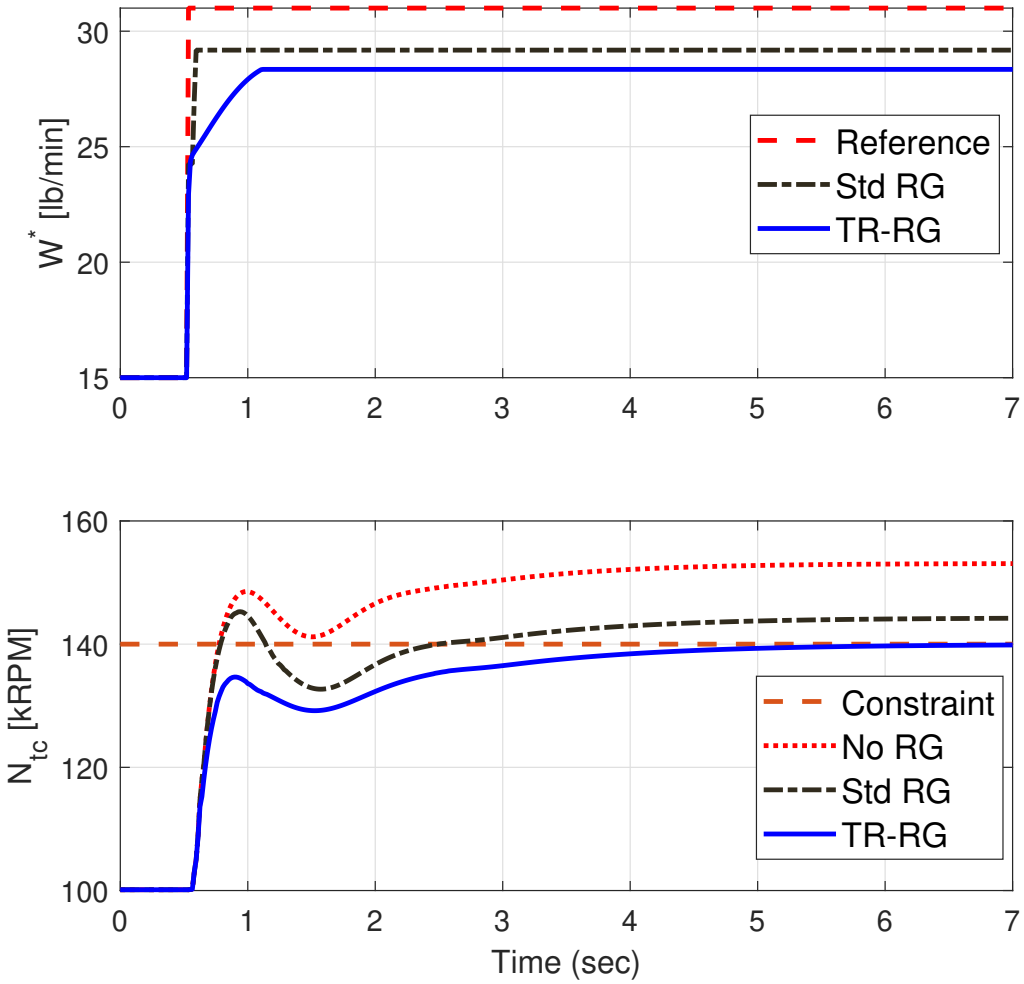

Figure 5.6: Nonlinear turbocharged engine simulation with RG and TR-RG. Top plot shows the desired air mass and bottom plot shows turbo speed.

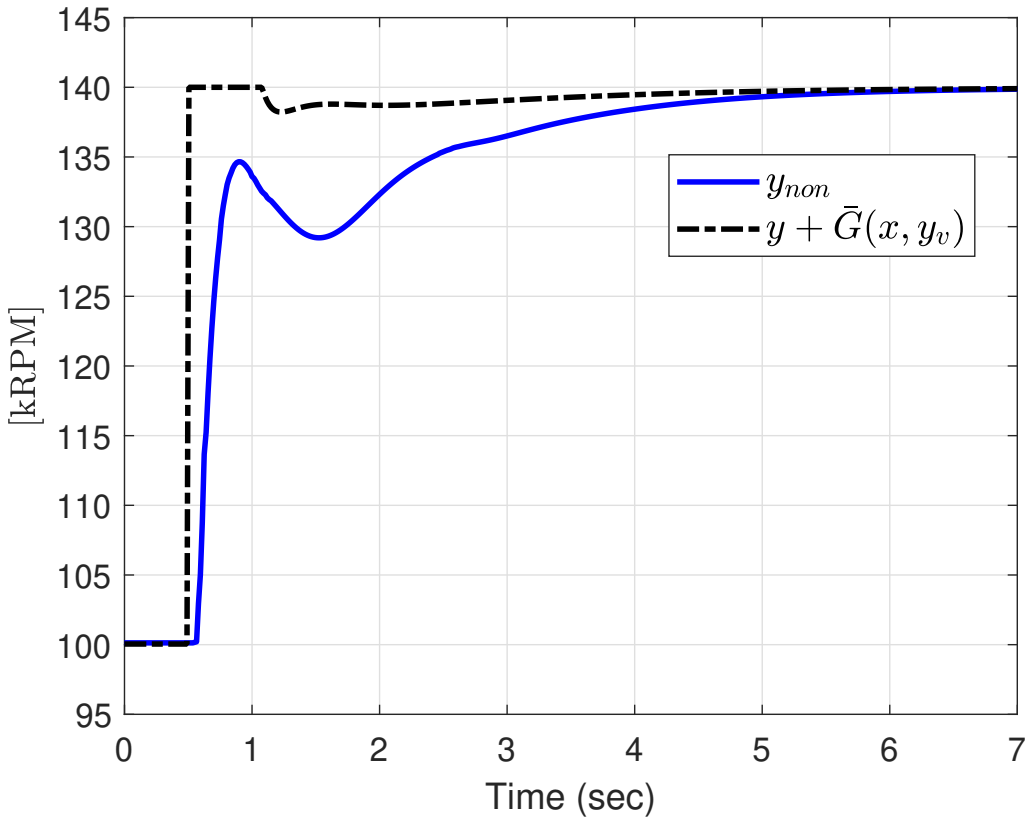

Figure 5.7: y_{non} *vs.* $y+\bar{G}(x, y_v)$*, where* y_{non} *and* y *are the turbo speed output of the nonlinear and linear systems respectively.*

with a 2.3L Eco-Boost Turbocharged Gasoline engine. Similar to the example in Section 5.4, the constraint is imposed on the turbo speed (N_{tc}) and the governed input is the desired air mass entering the engine (W^*). Since the actual turbo speed limit is typically not reachable at sea-level, we lower the constraint artificially for proof of concept. As discussed in Section 5.3.2, engine speed is a slow-varying parameter that affects the steady-state mapping of this system. We thus use the existing dynamometer data to find the steady-state map (ζ) from the governed input to the constrained output at different engine speed conditions. To tune the ROAS, we excite the vehicle with step inputs applied to the accelerator pedal, which translate into torque requests, and finally desired air mass requests. The desired air mass requests are also used as the input of a linear model of the engine that is linearized near the constraint. The collected data is used to calibrate γ_i using Algorithm 2.

To evaluate the scheme, we use two different maneuvers. In the first maneuver, we apply a single step to the accelerator pedal and then release it, in order to emulate a condition of an abrupt change in the engine torque request. The engine speed variation is presented in Fig. 5.8, and the turbo speed response and desired air mass request are presented in Fig. 5.9, where the dotted blue signal is the base condition with no turbo speed constraint (No RG in the loop), and the magenta solid line is the TR-RG results. The constraint on the turbo speed is enforced at all times for this maneuver. Note that we have concealed the axes labels and the value of the constraint due to the confidential nature of the data.

The second maneuver consists of variable pedal steps starting from both zero and non-zero initial turbo speeds. The variation of the engine speed is presented in Fig. 5.10 and the results are presented in Fig. 5.11. As can be seen, the scheme enforces the constraints for all times.

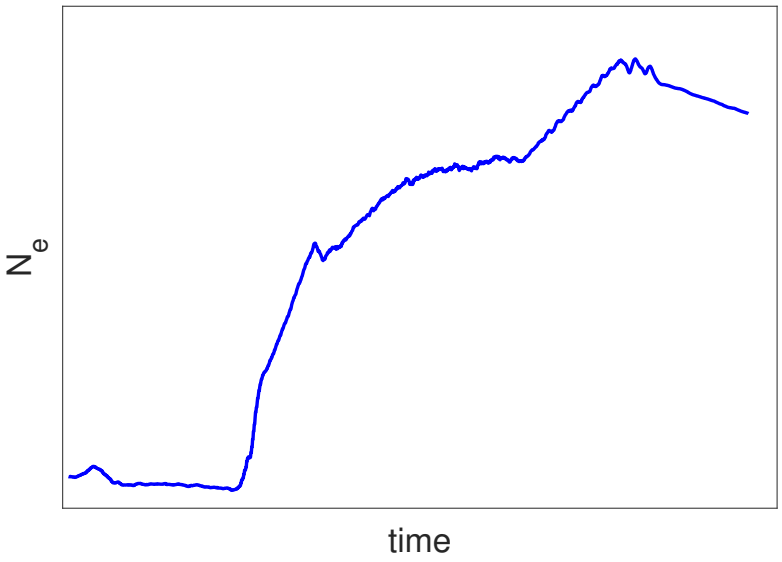

Figure 5.8: Engine speed for single step maneuver.

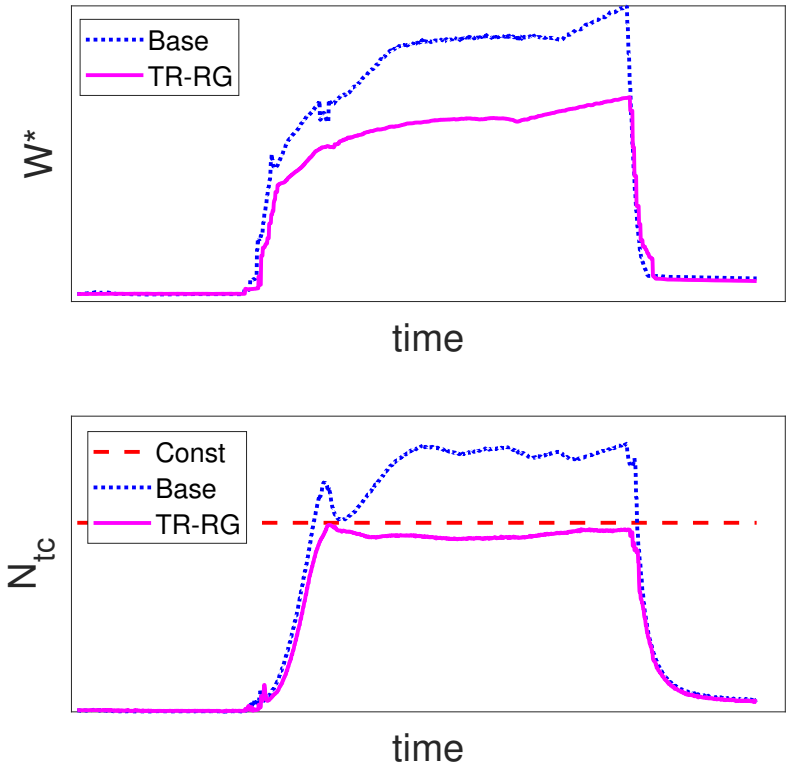

Figure 5.9: Vehicle results for a single step. Bottom plot is the turbo speed (constrained output) with the dashed line being the constraint, and top plot is the desired air mass (governed reference).

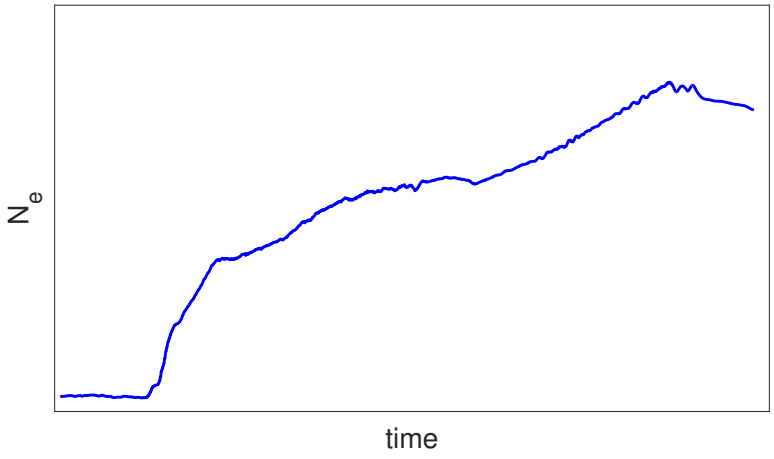

Figure 5.10: Engine speed for variable steps.

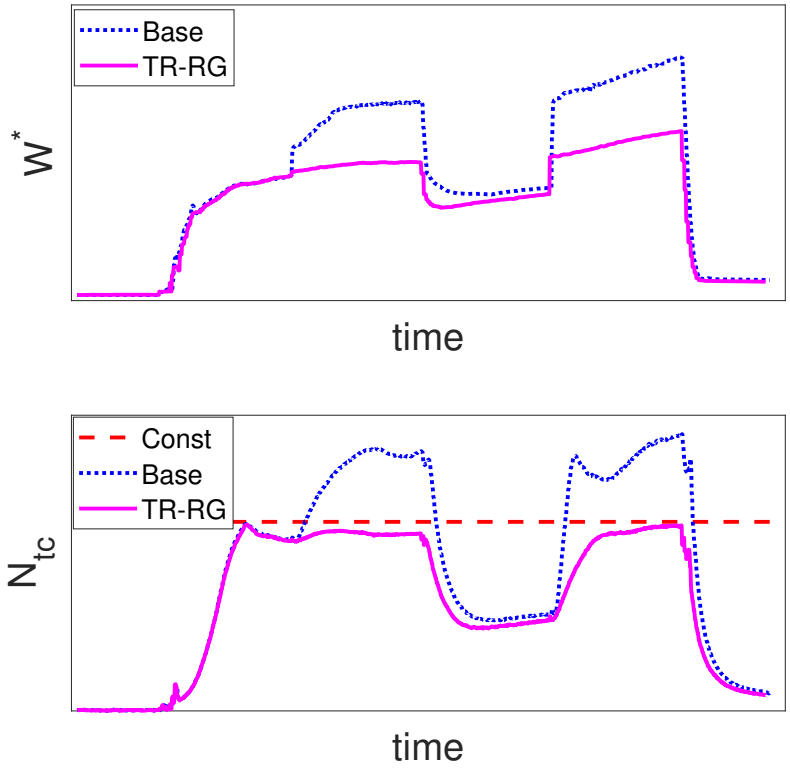

Figure 5.11: Vehicle results for variable steps. Bottom plot turbo speed and top plot is desired air mass.

Chapter 6

Implementations

This chapter presents a set of simulations with TR-RG for different applications. The idea is to study the extension of TR-RG to Single-Input Multiple Outputs (SIMO) systems and explore the effects not satisfying assumption A.5.1.2. By doing this, we can have a better understanding of the robustness and versatility of TR-RG.

6.1 Rollover Prevention: TR-RG

In automotive applications, rollover happens when a vehicle's roll angle increases abnormally, which is normally the result of an abrupt steering wheel change or lost of control. To address this problem, we consider that given the vehicle dynamics, a driver in the loop, and a set of rollover avoidance constraints, we can implement TR-RG in order to find a control law for the steering angle, such that the constraints are always enforced.

6.1.1 Vehicle model

The nonlinear vehicle dynamics model is developed based on [3, 213]. The model includes a nonlinear model for the tire forces and the suspension. The vehicle forces diagram is presented in Fig. 6.1. Before getting into the model details, the notation description is listed below:

- v_x: longitudinal vehicle speed component.
- v_y: lateral vehicle speed component.
- ϕ: vehicle roll angle.
- r: yaw angular speed.
- p: roll angular speed.

- $F_{x,i}$: longitudinal tire force, with $i = FL, FR, RL, RR$ (i.e., Front left, Front rear, Rear left, and Rear right).
- $F_{z,i}$: lateral tire force.
- $F_{z,i}$: normal tire force.
- N_T: total yaw moment.
- α: tire slip angle.
- λ: tire slip ratio.
- v_i: linear velocities at wheel hubs.
- δ_f: steering angle at the front wheels.
- δ_{sw}: steering wheel angle.

The equations of motion are given by:

$$\begin{aligned}
\dot{v}_x &= \frac{F_{x,T} - m_{sM} h_{sM} p \cos(\phi)}{m} + v_y r, \\
\dot{v}_y &= \frac{F_{y,T} + m_{sM} h_{sM} (\dot{p} \cos(\phi) - p^2 \sin(\phi))}{m} - v_x r, \\
\dot{r} &= \frac{N_T}{I_{zz}}, \\
\dot{p} &= \frac{m_{sM} g h_{sM} - K_s}{I_e} \phi - \frac{D_s}{I_e} p + \frac{m_{sM} h_{sM}}{m I_e} F_{y,T},
\end{aligned} \tag{6.1}$$

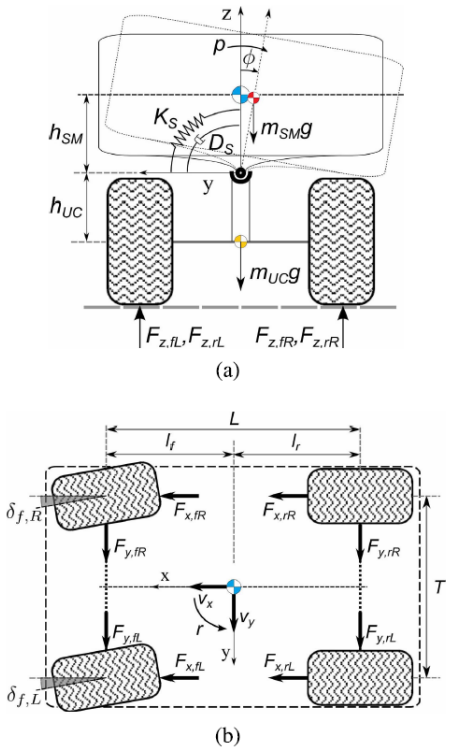

Figure 6.1: Vehicle forces diagram. (a) Rear view. (b) Top view. Image from [3]

where,

$$
\begin{aligned}
F_{x,T} =& F_{x,FL} + F_{x,FR} - (F_{y,FL} + F_{y,FR})\delta_f + F_{x,RL} + F_{x,RR}, \\
F_{y,T} =& (F_{x,FL} + F_{x,FR})\delta_f + F_{y,FL} + F_{y,FR} + F_{y,RL} + F_{y,RR}, \\
N_T =& (F_{x,FL} + F_{x,FR})l_f\delta_f + (F_{y,FL} + F_{y,FR})l_f - (F_{y,RL} + F_{y,RR})l_r + \\
& \frac{T}{2}[F_{x,FR} - F_{x,FL} - (F_{y,FR} - F_{y,FL})\delta_f + F_{x,RR} - F_{x,RL}], \\
I_e =& I_{xx} - \frac{m_{SM}^2 h_{sM}^2}{m}
\end{aligned}
\tag{6.2}
$$

where the corresponding lateral and longitudinal forces are given by the Magic Tire equation [3], this is:

$$
\begin{aligned}
&\begin{bmatrix} F_{x,i} \\ F_{y,i} \end{bmatrix} = F_p P(s_c, C, E)\hat{\mathbf{s}}, \\
&P(s_c, C, E) = \sin\left(C\tan^{-1}\left[\frac{s_c}{C}(1-E) + E\tan^{-1}(\frac{s_c}{C})\right]\right), \\
&s_c = \frac{C_\alpha ||\mathbf{s}||}{F_p}, \quad C_\alpha = c_1 mg\left(1 - e^{-\frac{c_2 F_z}{mg}}\right), \quad c_1 = \frac{BCD}{4\left(1 - e - \frac{c_2}{4}\right)}, \\
&F_p = \frac{F_z 1.0527D}{1 + \left(\frac{1.5F_z}{mg}\right)^3}, \quad \mathbf{s} = \begin{bmatrix} \lambda \\ \tan(\alpha) \end{bmatrix}, \quad \hat{\mathbf{s}} = \frac{\mathbf{s}}{||\mathbf{s}||}.
\end{aligned}
\tag{6.3}
$$

The main source of nonlinearities in this model comes from (6.3). The slip ratio (λ), slip angle, and normal force are discussed next. The slip ratio is defined as:

$$
\lambda_i = \frac{r_w \omega_i}{v_i} - 1, \tag{6.4}
$$

where w_i is the wheel angular speed and the linear velocities at wheel hubs are given by:

$$
\begin{aligned}
v_{FL} &= \left(v_x - \frac{T}{2}r\right)\cos(\delta_f) + (v_y + l_f r)\sin(\delta_f), \\
v_{FR} &= \left(v_x + \frac{T}{2}r\right)\cos(\delta_f) + (v_y + l_f r)\sin(\delta_f), \\
v_{RL} &= v_x - \frac{T}{2}r, v_{RR} = v_x + \frac{T}{2}r.
\end{aligned}
\tag{6.5}
$$

The steering angle at the front wheels is given by, $\delta_f = K * \delta_{sw}$. The slip angle for each tire is given by:

$$
\begin{aligned}
&\alpha_{FL} = \delta_f - \tan^{-1}\left(\frac{v_y + l_f r}{v_x - \frac{T}{2}r}\right), \alpha_{FR} = \delta_f - \tan^{-1}\left(\frac{v_y + l_f r}{v_x + \frac{T}{2}r}\right) \\
&\alpha_{RL} = -\tan^{-1}\left(\frac{v_y - l_r r}{v_x - \frac{T}{2}r}\right), \alpha_{RR} = -\tan^{-1}\left(\frac{v_y - l_r r}{v_x + \frac{T}{2}r}\right)
\end{aligned}
\tag{6.6}
$$

Table 6.1: Vehicle parameters. Data from [3]

Parameter	Description	Value	Unit
m_{sM}	Rolling mass	1700	kg
m	Total vehicle mass	2000	kg
h_{sM}	Distance from sprung mass CG to the roll axis	0.8580	m
l_r	Distance from rear axle to vehicle CG	1.750	m
l_f	Distance from front axle to vehicle CG	1.750	m
T	Lateral distance between wheels	1.26	m
I_{xx}	Sprung mass roll moment of inertia about the roll axis	1280	kg/m^2
I_{zz}	Vehicle yaw moment of inertia about the z axis	2800	kg/m^2
K_s	Total suspension roll stiffness	95707	N · m
D_s	Total suspension roll damping	7471	N · m · s/rad
g	gravity acceleration	9,8	m/s^2
r_w	wheel ratio	0.3	m
K	Steering wheel/steering front wheels angle ratio	0.2	-

Finally the wheel normal load are given by:

$$\begin{aligned} F_{z,FL} &= \frac{mgl_r}{2L} - K_{RSF} \cdot mg\frac{m_{sM}h_{sM}}{mT}\sin(\phi), \\ F_{z,FR} &= \frac{mgl_r}{2L} + K_{RSF} \cdot mg\frac{m_{sM}h_{sM}}{mT}\sin(\phi), \\ F_{z,RL} &= \frac{mgl_f}{2L} - (1 - K_{RSF}) \cdot mg\frac{m_{sM}h_{sM}}{mT}\sin(\phi), \\ F_{z,RR} &= \frac{mgl_f}{2L} + (1 - K_{RSF}) \cdot mg\frac{m_{sM}h_{sM}}{mT}\sin(\phi), \end{aligned} \tag{6.7}$$

where $K_{RSF} = K_s/(m_{s_M} \cdot g \cdot L)$, is the front proportion of total roll constant. For this model the vehicle speed is assumed to be constant, and hence the acceleration is not considered in (6.7). The parameters for the model are presented in Table 6.1.

For this nonlinear system the constrained output is the lateral Load Transfer Ratio (LTR), which is defined as:

$$LTR = -\frac{2}{mgT}(D_s p + K_s \phi) \tag{6.8}$$

Fig. 6.2 presents the response of the nonlinear system to an abrupt steering wheel change, this shows how the LTR goes above 100% and below −100%, meaning that the vehicle rolls over.

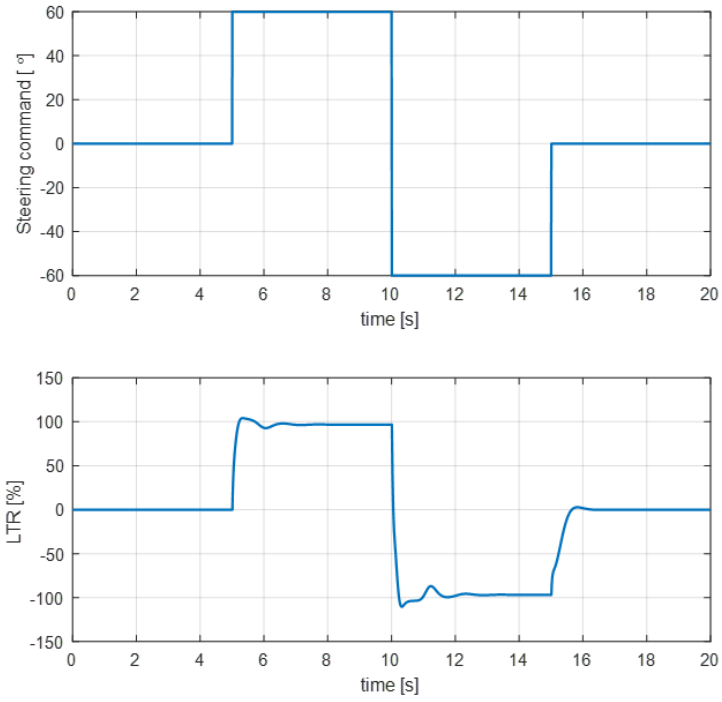

Figure 6.2: Step responses of the nonlinear system

Next, we implement TR-RG for this model.

6.1.2 Rollover Avoidance with TR-RG

The constraint imposed on the LTR is defined by the compact set $Y = [-100\%, 100\%]$, and the governed reference is the steering wheel command (δ_{sw}). The first step to implement TR-RG is to evaluate the steady-state characterization of the nonlinear plant, which is presented in Fig. 6.3.

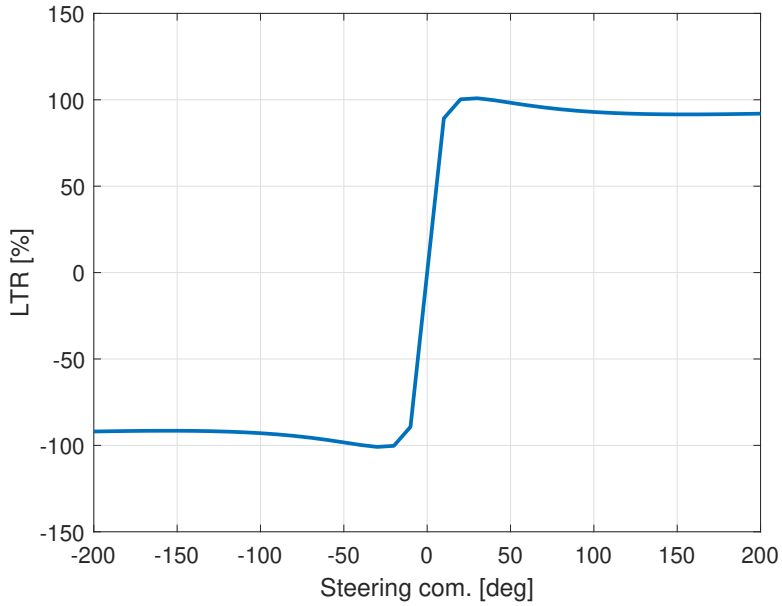

Figure 6.3: Steady-state map of nonlinear plant for initial speed of 120 km/h

As can be seen from Fig. 6.3, the nonlinear plant steady-state map is not invertible, hence the forward and inverse mapping cannot be implemented. Nevertheless, note that at the initial vehicle speed of 120 km/h, there are just a few violations at steady-state, so even without the steady-state map in the loop, the mismatch between the linear and nonlinear system can be taken into account by the tuning process. Specifically γ_2 and γ_4.

Next, the nonlinear plant must be linearized around an operating point. Note from Figure 6.3 that if the plant is linearized around the upper constraint (i.e., 100%), there will be a significant mismatch when the system operates around the lower constraint. For this reason, two linear models are considered, one from linearizing at $\delta_{sw} = 10°$ (i.e., close to the upper constraint) and at $\delta_{sw} = -10°$ (i.e., close to the lower). With this, TR-RG for multiple outputs (Section 5.3.1) is implemented. The state vector for these linear model is:

$$x = \begin{bmatrix} v_x & v_y & r & p & \phi \end{bmatrix}$$

In order to compare TR-RG with standard RG, the latter is implemented based on linear model obtained around $\delta_{sw} = 0$. The function $g(\cdot)$ is selected to be the squared norm, i.e., $||x - x_{ss}||_P^2$. From the tuning process we obtain the following parameters for $\gamma_j, j = 1, \ldots, 4$. These are:

$$\gamma_1 = 7.2168,$$

$$\gamma_2 = 73.6039,$$

$$\gamma_3 = 6.9583,$$

$$\gamma_4 = 73.7008.$$

A comparison of the ROAS and MAS is presented in Figure 6.4, which shows a projection onto the input (δ_{sw}) and four state ($x_4 = p$). This shows how the ROAS shrinks the solution space in order to account for the plant model mismatch and hence enforces the constraints. Also, note from the tuning process, that γ_2 and γ_4 are larger than γ_1 and γ_3. The reason is the violations at steady-state that were observed in Fig. 6.3. Through the optimization process the ROAS is robustified, and hence the TR-RG can enforce the constraints at steady-state even in the absence of the steady-state characterization of the nonlinear system.

The results of implementing TR-RG based on multiple outputs and standard RG for the nonlinear vehicle with an initial vehicle speed of $120km/h$ are presented in Figure 6.5 and Figure 6.6.

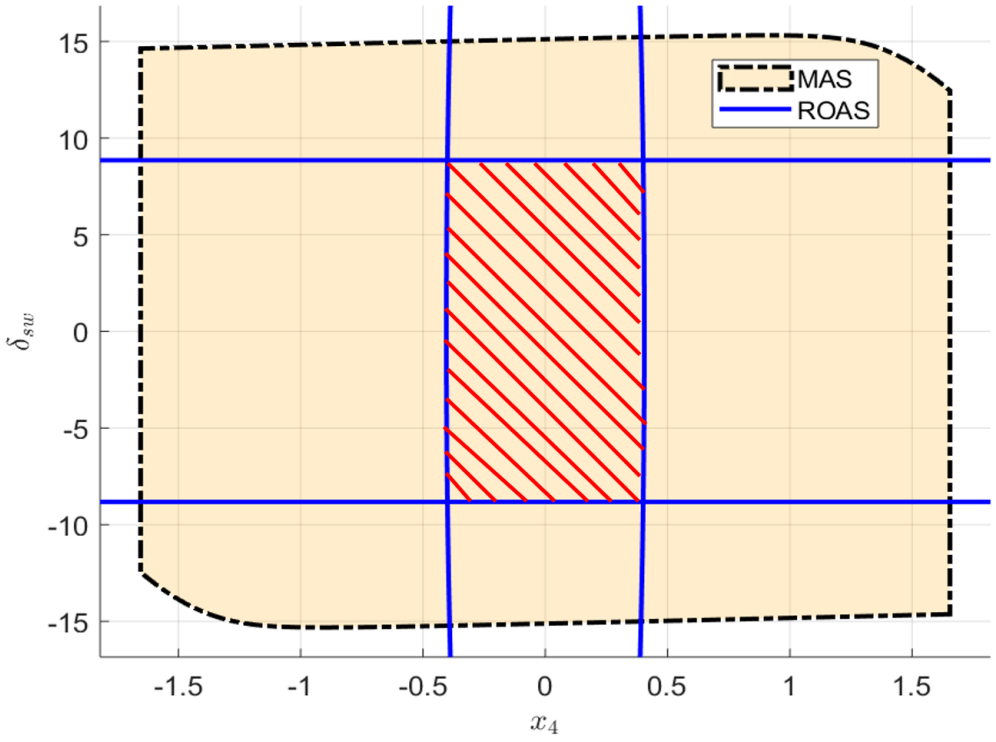

Figure 6.4: ROAS vs. MAS for rollover avoidance. The x_4 corresponds to roll angular speed, p.

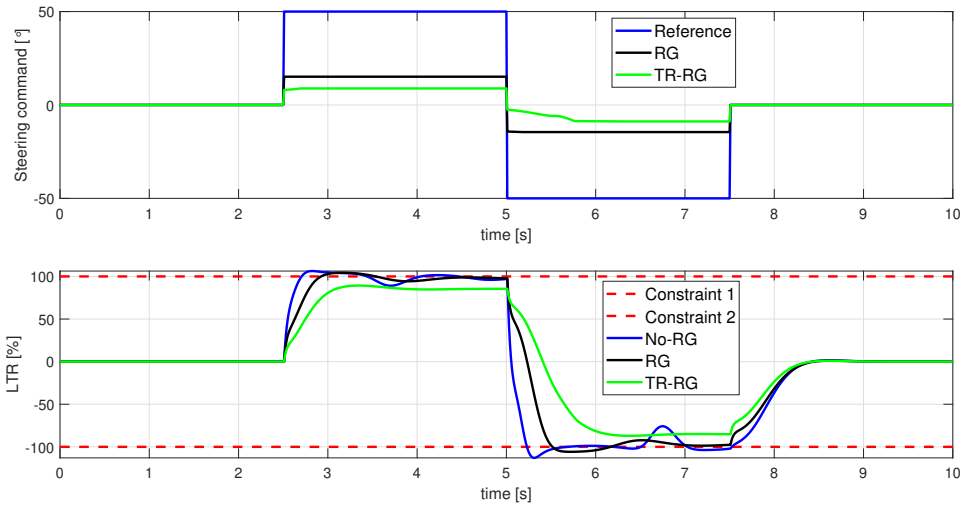

Figure 6.5: TR-RG for rollover avoidance, step responses. Initial vehicle speed $120km/h$.

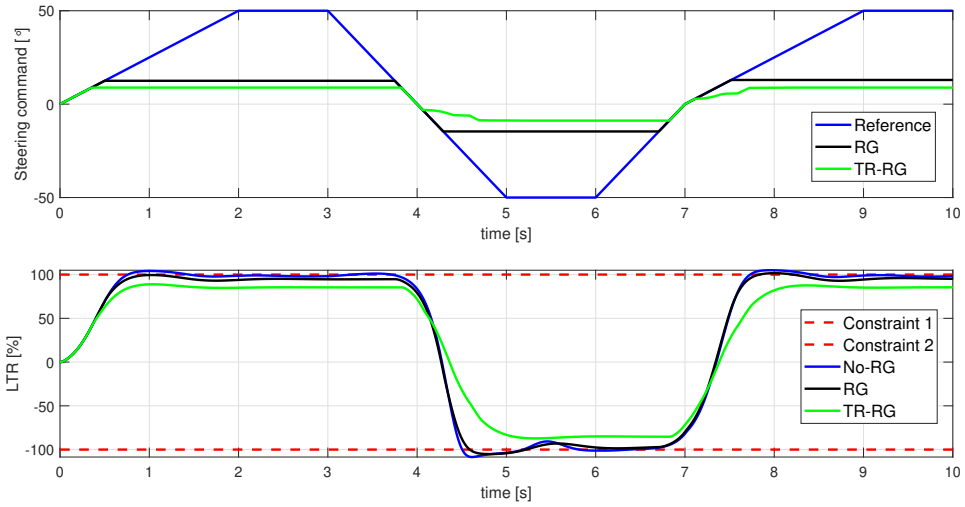

Figure 6.6: TR-RG for rollover avoidance, ramp responses. Initial vehicle speed $120km/h$.

In order to test the robustness of TR-RG. The vehicle initial speed condition is changed to $140km/h$, the results for this condition are presented in Figure 6.7 and Figure 6.8.

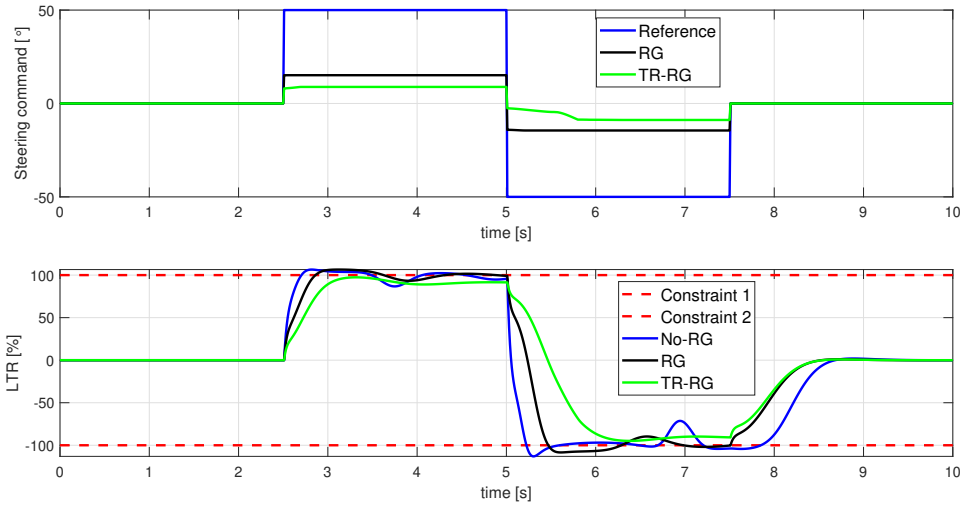

Figure 6.7: TR-RG for rollover avoidance, step responses. Initial vehicle speed $140km/h$.

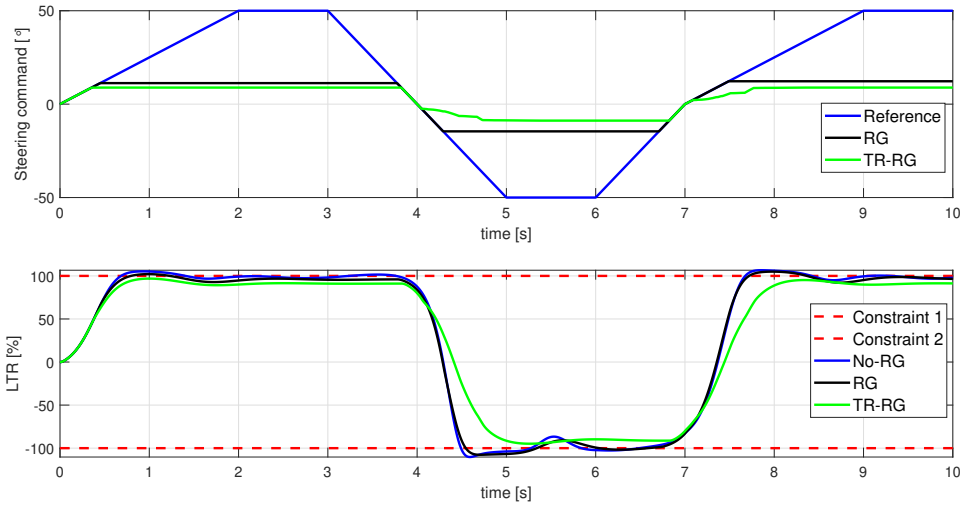

Figure 6.8: TR-RG for rollover avoidance, ramp responses. Initial vehicle speed $140km/h$.

The results show how TR-RG avoid drastic steering wheel changes in order to avoid rollover at high vehicle speed.

6.2 Inverted Pendulum: TR-RG

The idea with the inverted pendulum is to stabilize the pendulum in the upright position. However, it is well known that this is not an easy task. In this section, we analyzed how TR-RG is implemented to enforce constraints in this type of applications.

6.2.1 DC-motor and Inverted Pendulum Nonlinear Model

An schematic of the inverted pendulum model is shown in Fig. 6.9, which shows the pendulum position, α, and rotatory arm position, θ. Their first and second derivatives

represent the speed and acceleration respectively.

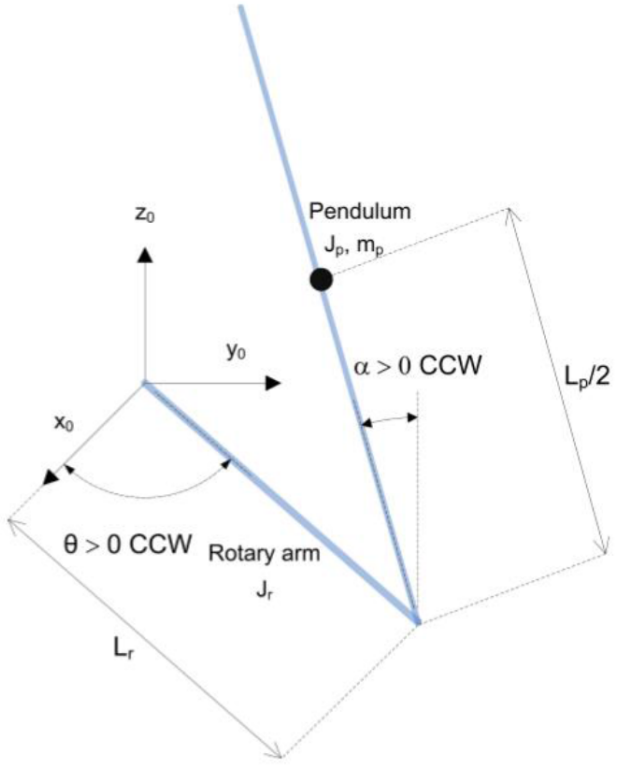

Figure 6.9: Rotatory inverted pendulum model [4]

The nonlinear dynamic equations are given by:

$$\begin{aligned}
&\left(m_p L_r^2 + \frac{1}{4} m_p L_p^2 - \frac{1}{4} m_p L_p^2 \cos(\alpha)^2 + J_r\right) \ddot{\theta} - \left(\frac{1}{2} m_p L_p L_r \cos(\alpha)\right) \ddot{\alpha} \\
&+ \left(\frac{1}{2} m_p L_p \sin(\alpha)\cos(\alpha)\right) \dot{\theta}\dot{\alpha} + \left(\frac{1}{2} m_p L_p L_r \sin(\alpha)\right) \dot{\alpha}^2 = \frac{K_t}{R_m} V, \\
&- \frac{1}{2} m_p L_p L_r \cos(\alpha) \ddot{\theta} + \left(J_p + \frac{1}{4} m_p L_p^2\right) \ddot{\alpha} - \frac{1}{4} m_p L_p \cos(\alpha)\sin(\alpha) \dot{\theta}^2 \\
&- \frac{1}{2} m_p L_p g \sin(\alpha) = 0.
\end{aligned} \tag{6.9}$$

The parameters of this nonlinear model are listed in Table 6.2. The nonlinear model (6.9) is linearized on the upright position, and two-loop controller are designed for the inverted

Table 6.2: Inverted pendulum parameters

Parameter	Description	Value	Unit
R_m	Terminal resistance	8.4	Ω
K_t	Torque constant	0.042	Nm/A
L_r	Total arm length	0.085	m
J_r	Arm moment of inertia	5.7198×10^{-5}	$kg \cdot m^2$
m_p	Mass of pendulum	0.024	kg
L_p	Total pendulum length	0.129	m
J_p	Arm moment of inertia	3.3282×10^{-5}	$kg \cdot m^2$

pendulum. The first loop is closed around the pendulum angle, α, with a lead compensator and is intended to stabilize the pole in the RHP. The second loop is closed around the arm angle, θ, with a PV (Proportional-Velocity) controller. The control command is the voltage that is injected to the DC motor, which is defined as:

$$V = V_1 + V_2$$

where

$$V_1 = C_1(R_\alpha - \alpha),\ C_1(s) = \frac{182.62(s + 10.584)}{s + 50}$$

is the lead compensator and the PV controller is:

$$V_2 = -2(R_\theta - \theta) + \frac{-75s}{s + 50}$$

where R_θ is the desired arm position, and R_α is the desired pendulum position. The latter is set to zero ($R_\alpha = 0$), since this is the desired pendulum position (i.e., upright). Using Simulink/Matlab to obtain the closed-loop linear model, the state vector $x \in \mathrm{R}^6$ is:

$$x = \begin{bmatrix} \theta & \dot{\theta} & \alpha & \dot{\alpha} & x_c \end{bmatrix},$$

where x_c represents the control states, which is composed by the lead and PV controllers. The constraint is imposed on the speed of the rotatory arm, $\dot{\theta}$, with a constraint value of $\Big[60, -60\Big]$. The closed loop system is excited with steps and the results are presented in Fig. 6.10.

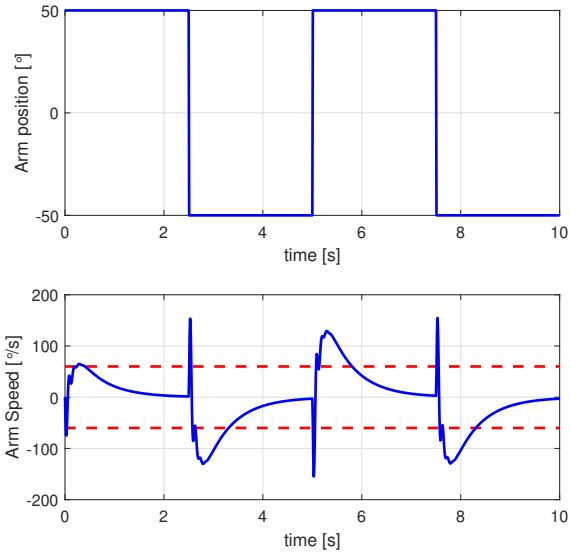

Figure 6.10: Nonlinear inverted pendulum response

Next, the TR-RG is implemented to enforce the constraints on the nonlinear model.

6.2.2 Nonlinear Inverted Pendulum with TR-RG

To implement TR-RG, first, we analyzed the steady-state characterization of the plant from the governed reference, R_θ, to the constrained output $\dot{\theta}$. By simple inspection is possible to note that the steady-state map is not invertible, the reason is that for all input commands R_θ, the arm speed $\dot{\theta}$ converges to zero at steady-state. Therefore, the TR-RG is implemented with no steady-state maps in the loop.

The function $g(\cdot)$ is selected to be the squared norm, i.e., $||x - x_{ss}||_P^2$, the matrix P is tuned based on the states of the linear model as explained in Section 5.4. The next step is to tune the parameters of the TR-RG, to do so, the linear model and nonlinear system are excited by the same inputs, the results are:

$$\gamma_1 = 0.6463,$$

$$\gamma_2 = 0,$$

$$\gamma_3 = 0.6463,$$

$$\gamma_4 = 0.$$

These results are consistent with the previous observation, i.e., the steady-state value for the arm speed is zero for all references. A comparison with standard RG and the results of TR-RG are presented in Fig. 6.11.

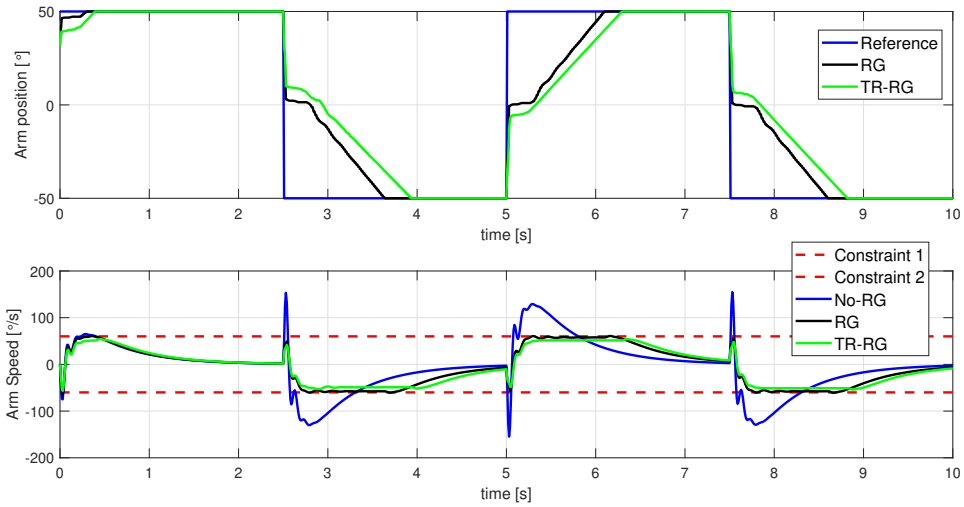

Figure 6.11: Nonlinear inverted pendulum response with RG and TR-RG

As expected the standard RG is not able to enforce the constraint for all times, whereas TR-RG maintains the arm speed constraint satisfied by slowing the reference when needed.

Chapter 7

Conclusions and Future Works

In this dissertation, new RG schemes were presented for different types of systems, namely: linear, stochastic, and nonlinear systems. The schemes presented in this work were supported by a systematical analysis of their main properties and implementation characteristics. Hence, the development of this work made both theoretical and practical contributions to the control community. This chapter presents the final observations and remarks about the contributions of this work.

7.1 Review of Constraint Management Schemes

Chapter 2 provides a more technical overview of the different constraint management techniques for linear and nonlinear systems, with a focus around reference governors. This chapter showed main technical characteristics, advantages and disadvantages of different schemes based on RG approaches, MPC, Lyapunov-based, and machine learning. This helps to provide a more complete idea of the different schemes and areas of opportunities for constraint management.

7.2 Stochastic Reference Governor

A chance constraint approach to maximal output admissible sets (MAS) and reference governors (RG) was studied in Chapter 3. An analysis of MAS was presented, and showed that the removal of the first halfspace leads to constraint violation for one timestep, followed by recovery for all future times. The idea presented in [24] was extended to Lyapunov stable systems with output constrains, in order to build a stochastic robustly invariant MAS (SR-MAS). Important properties such as positive invariance and finite determinism of SR-MAS

were then presented. Added to this, an algorithm to compute the SR-MAS in finite time was provided. Minor modifications to this algorithm were shown, for systems without disturbance feedthrough. Finally, a numerical validation, with a mass-spring-damper model, was used to compare standard and stochastic RG. The latter showed a less conservative response, which was possible thanks to the expansion of the admissible set introduced by the SR-MAS.

In Stochastic RG, the structure of the set set $Z_{t,\beta}$ and the properties of W_β are important to ensure properties such as positive invariance and recursive feasibility. These must be satisfied for the optimal performance of Stochastic RG.

7.3 Recovery Reference Governor

A new method for a Reference Governor (RG), referred to as the Recovery RG (RRG), was presented in Chapter 4. This method uses a set-theoretic approach to find a feasible governed input upon constraint violation. The violation is assumed to be caused by unknown and unmodeled external disturbances. The RRG scheme preserves desirable characteristics of the RG and introduces additional logic to switch between standard operation and recovery operation. Recursive feasibility of the proposed solution was discussed, and guarantees of recovering from constraint violation were presented. This paper considers the case when a disturbance model is not available and extra considerations need to be introduced in order to treat plant/model mismatch due to parametric uncertainties. A numerical simulation with a turbocharged gasoline engine illustrated the benefits of applying the RRG and how disturbance estimation plays an important role in the RRG scheme.

This scheme highly depends on an accurate estimation of the disturbance to recover from violations, this estimation is mainly based on assumption A.4.1.1. Thus, a limitation of RRG is when assumption A.4.1.1 is not satisfied, for instance, in a case when a system

is affected by a disturbance with a significant amplitude of variation.

7.4 Transient Robust-Reference Governor

A novel scheme, referred to as Transient Robust Reference Governor (TR-RG), to impose pointwise-in-time constraints on nonlinear systems was presented in Chapter 5. The idea is to employ a Reference Governor (RG) that uses predictions obtained from a linearized model of the nonlinear plant, but incorporate mechanisms to handle the mismatch between the linear model and the nonlinear plant at steady-state and during transients. The former is achieved by using the steady-state characterization of the nonlinear system inside the TR-RG. For the latter, the idea of a Robust Output Admissible Set (ROAS) was introduced, which is obtained using a data-driven approach. The theoretical properties and the computational aspects of the ROAS were studied and algorithms for its tuning were presented.

Two extensions of TR-RG were presented to make the scheme applicable to a broader class of systems. The first was extension to multi-output systems and the second was extension to systems whose steady-state characterization depends on a slowly-varying parameter. Experimental results on a turbocharged engine were presented in order to validate the performance of TR-RG. It was shown that TR-RG can enforce the constraints in realistic settings.

This scheme has two main limitations, one is that the scheme can enforce the constraints for all time only if the system operates in the region of interest (i.e., operating conditions that were used to collect the data for tuning). Two, if assumption A.5.1.2 is satisfied, then constraint enforcement at steady-state is guaranteed. However, even if initially the

steady-state map is correct, as the systems operates over a period of time their steady-state mapping will vary due to physical deterioration of the internal components. Therefore, steady-state enforcement may not be guaranteed. The latter may be resolved by recreating some predictable changes in the systems and characterizing their steady-state maps. Hence, a set membership condition can be defined to identify the corresponding steady-state map and enforce constraints at steady-state.

7.5 Other Practical Implementations of TR-RG

In Chapter 6 two practical applications for TR-RG were presented. The first was related to rollover stability, and the second was an inverted pendulum application. The main idea was to validate the extensions of TR-RG presented in Chapter 5 Section 5.3.1 for multiple outputs applications, and to test the robustness of TR-RG when the steady-state characterization is not invertible, i.e., Assumption A.5.1.2 does not hold. These two aspects were successfully tested with these applications and constraint enforcements were achieved. Added to this, different initial conditions cases were studied for the rollover avoidance application with successful results as reported in Chapter 6 Section 6.1.2.

7.6 Future Work

For each one of the novel RG schemes proposed in this dissertation, we list the future works below:

- Stochastic RG: Future work will explore the extension of SR-MAS for systems with Gaussian disturbance models, which definitely amplify the applicability of the stochas-

tic RG to a broader type of systems. Investigation of RG schemes for systems with additive Gaussian or Uniform disturbances and multiplicative disturbances due to plant/model mismatch. For this case, some approaches like [64] studied the deterministic case, where the worst case disturbance realization is considered at each time step, and hence more conservative results may be obtained. Therefore, for constraint management of stochastic systems, there is still a room for innovation and improvement that can be filled by extending stochastic RG.

- Recovery RG: Future work will explore a relaxation of the assumptions used in this work. That is, relaxing assumption A.4.1.1, and study the case when a varying disturbance affects the system. In alignment with this relaxation, it is necessary to re-evaluate assumption A.4.1.2 since allowing a disturbance big enough such that the equilibrium is not admissible may produce undesirable abrupt jumps in the governed signal. All these composed a challenging problem that definitely motivates the extension of RRG to a broader type of applications. Also, in terms of practical implementation, it is desired to study the extension of RRG to nonlinear plants.

- TR-RG: Future work will explore the extension of this work to systems with structured uncertainties and additive unknown disturbances to reduce the transient margin of the ROAS. Richer structures will be considered for the function $g(\cdot)$ to reduce the transient margin of the ROAS and improve the tuning process. Also, it is our interest to study systems with polytopic constraints. Finally, we will study the extension of TR-RG to systems with multiple inputs.

Bibliography

[1] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: theory and practice-a survey. *Automatica*, 25(3):335–348, 1989.

[2] H. R. Ossareh, J. Buckland, and M. Jankovic. Continuous compressor recirculation to improve boost response and mitigate compressor surge in turbocharged gasoline engines. In *2016 American Control Conference (ACC)*, pages 5093–5098, 2016.

[3] R. Bencatel, R. Tian, A. R. Girard, and I. Kolmanovsky. Reference governor strategies for vehicle rollover avoidance. *IEEE Transactions on Control Systems Technology*, 26(6):1954–1969, 2018.

[4] Quanser. Quanser rotatory inverted pendulum user manual. *User Manual*, 2016.

[5] David Q Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[6] Milan Korda and Colin N Jones. Certification of fixed computation time first-order optimization-based controllers for a class of nonlinear dynamical systems. In *American Control Conference (ACC), 2014*, pages 3602–3608. Ieee, 2014.

[7] M. Farina, L. Giulioni, L. Magni, and R. Scattolini. A probabilistic approach to model predictive control. In *52nd IEEE Conference on Decision and Control*, pages 7734–7739, Dec 2013.

[8] Georg Schildbach, Lorenzo Fagiano, Christoph Frei, and Manfred Morari. The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *Automatica*, 50(12):3009 – 3018, 2014.

[9] I. Kolmanovsky, E. Garone, and S. Di Cairano. Reference and command governors: A tutorial on their theory and automotive applications. In *2014 American Control Conference*, pages 226–241, June 2014.

[10] P. Kapasouris, M. Athans, and G. Stein. Design of feedback control systems for stable plants with saturating actuators. pages 469–479 vol.1, Dec 1988.

[11] Elmer G Gilbert and Ilya Kolmanovsky. Discrete-time reference governors for systems with state and control constraints and disturbance inputs. In *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, volume 2, pages 1189–1194. IEEE, 1995.

[12] Elmer G. Gilbert and K. T. Tan. Linear systems with state and control constraints: the theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, Sep 1991.

[13] Jing Sun and Ilya V Kolmanovsky. Load governor for fuel cell oxygen starvation protection: A robust nonlinear reference governor approach. *IEEE Transactions on Control Systems Technology*, 13(6):911–920, 2005.

[14] JL Guzman, T Alamo, M Berenguel, S Dormido, and EF Camacho. A robust constrained reference governor approach using linear matrix inequalities. *Journal of Process Control*, 19(5):773–784, 2009.

[15] Ilya Kolmanovsky and Elmer G Gilbert. Maximal output admissible sets for discrete-time systems with disturbance inputs. In *American Control Conference, Proceedings of the 1995*, volume 3. IEEE, 1995.

[16] Ilya Kolmanovsky and Elmer G Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical problems in engineering*, 4(4):317–367, 1998.

[17] Elmer G. Gilbert and Ilya Kolmanovsky. Fast reference governors for systems with state and control constraints and disturbance inputs. *International Journal of Robust and Nonlinear Control*, 9(15):1117–1141, 1999.

[18] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems*, 36(6):30–44, 2016.

[19] James A Primbs. A soft constraint approach to stochastic receding horizon control. In *Decision and Control, 2007 46th IEEE Conference on*, pages 4797–4802. IEEE, 2007.

[20] Marcello Farina, Luca Giulioni, and Riccardo Scattolini. Stochastic linear model predictive control with chance constraints–a review. *Journal of Process Control*, 44:53–67, 2016.

[21] Pu Li, Moritz Wendt, and GüNter Wozny. A probabilistically constrained model predictive controller. *Automatica*, 38(7):1171–1176, 2002.

[22] M Amini and M Almassalkhi. Corrective dispatch of uncertain energy resources using chance-constrained receding horizon control. *arXiv preprint arXiv:1810.08685*, 2018.

[23] Frauke Oldewurtel, Colin N Jones, and Manfred Morari. A tractable approximation of chance constrained stochastic mpc based on affine disturbance feedback. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4731–4736. IEEE, 2008.

[24] Chen Wang, Chong-Jin Ong, and Melvyn Sim. Linear systems with chance constraints: Constraint-admissible set and applications in predictive control. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 2875–2880. IEEE, 2009.

[25] Ngoc Anh Nguyen. Stochastic control based on convex lifting. In *American Control Conference (ACC), 2017*, pages 412–417. IEEE, 2017.

[26] Stefano Di Cairano Uros V. Kalabic, Abraham Goldsmith. Extended command governors for constraint enforcement in dual-stage processing machines. In *American Control Conference (ACC), 2017 IEEE International Conference on*. IEEE, 2017.

[27] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[28] James Blake Rawlings and David Q Mayne. Model predictive control: Theory and design, 2009.

[29] Munoz de la Pena, A. Bemporad, and C. Filippi. Robust explicit mpc based on approximate multi-parametric convex programming. In *Proceedings of Decision and Control (CDC)*. IEEE, 2004.

[30] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[31] A. Alessio and A. Bemporad. A survey on explicit model predictive control. In *Nonlinear model predictive control*, pages 345–369. Springer, 2009.

[32] Luca Zaccarian and Andrew R Teel. *Modern anti-windup synthesis: control augmentation for actuator saturation*, volume 36. Princeton University Press, 2011.

[33] K. P. Tee, S. S. Ge, and E. H. Tay. Barrier lyapunov functions for the control of output-constrained nonlinear systems. *Automatica*, 45(4):918–927, 2009.

[34] Emanuele Garone, Stefano Di Cairano, and Ilya Kolmanovsky. Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica*, 75:306–328, 1 2017.

[35] A. Vahidi, I. Kolmanovsky, and A. Stefanopoulou. Constraint handling in a fuel cell system: A fast reference governor approach. *IEEE Transactions on Control Systems Technology*, 15(1):86–98, Jan 2007.

[36] Neil Watson and Marian Janota. *Turbocharging the internal combustion engine*. Macmillan International Higher Education, 1982.

[37] S. Di Cairano and I. V. Kolmanovsky. Real-time optimization and model predictive control for aerospace and automotive applications. In *2018 Annual American Control Conference (ACC)*, pages 2392–2409, June 2018.

[38] Rolf Findeisen and Frank Allgöwer. An introduction to nonlinear model predictive control. In *21st Benelux meeting on systems and control*, volume 11, pages 119–141. Technische Universiteit Eindhoven Veldhoven Eindhoven, The Netherlands, 2002.

[39] Rodolphe Sepulchre, Mrdjan Jankovic, and Petar V Kokotovic. *Constructive nonlinear control*. Springer Science & Business Media, 2012.

[40] M. Jankovic. Combining control lyapunov and barrier functions for constrained stabilization of nonlinear systems. In *2017 American Control Conference (ACC)*, pages 1916–1922, May 2017.

[41] Alberto Bemporad, Daniele Bernardini, Ruixing Long, and Julian Verdejo. Model predictive control of turbocharged gasoline engines for mass production. Technical report, SAE Technical Paper, 2018.

[42] U. Kalabic, I. Kolmanovsky, J. Buckland, and E. Gilbert. Reference and extended command governors for control of turbocharged gasoline engines based on linear models. In *2011 IEEE International Conference on Control Applications (CCA)*, pages 319–325, Sept 2011.

[43] Hayato NAKADA, Peter MARTIN, Gareth MILTON, Akiyuki IEMURA, and Akira OHATA. An application study of online reference governor to boost pressure control for automotive diesel engines. *Transactions of the Society of Instrument and Control Engineers*, 50(3):197–202, 2014.

[44] N. Li, I. V. Kolmanovsky, and A. Girard. A reference governor for nonlinear systems with disturbance inputs based on logarithmic norms and quadratic programming. *IEEE Transactions on Automatic Control*, pages 1–1, 2019.

[45] Joycer Osorio, Mario Santillo, Julia Buckland Seeds, Mrdjan Jankovic, and Hamid R Ossareh. A reference governor approach towards recovery from constraint violation. In *2019 American Control Conference (ACC)*, pages 1779–1785. IEEE, 2019.

[46] A. Bemporad. Reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 43(3):415–419, Mar 1998.

[47] Elmer Gilbert and Ilya Kolmanovsky. Nonlinear tracking control in the presence of state and control constraints: a generalized reference governor. *Automatica*, 38(12):2063 – 2073, 2002.

[48] Marco Nicotra. *Constrained Control of Nonlinear Systems: The Explicit Reference Governor and its Application to Unmanned Aerial Vehicles*. PhD thesis, 2016. Doctorat en Sciences de l'ingénieur et technologie.

[49] E. Garone and M. M. Nicotra. Explicit reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 61(5):1379–1384, May 2016.

[50] M. M. Nicotra and E. Garone. The explicit reference governor: A general framework for the closed-form control of constrained nonlinear systems. *IEEE Control Systems Magazine*, 38(4):89–107, Aug 2018.

[51] K. Liu, N. Li, D. Rizzo, E. Garone, I. Kolmanovsky, and A. Girard. Model-free learning to avoid constraint violations: An explicit reference governor approach. In *2019 American Control Conference (ACC)*, pages 934–940, July 2019.

[52] L. Burlion, M. M. Nicotra, and I. V. Kolmanovsky. A fast reference governor for the constrained control of linear discrete-time systems with parametric uncertainties. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6289–6294, Dec 2018.

[53] Joycer Osorio, Mario Santillo, Julia Buckland Seeds, Mrdjan Jankovic, and Hamid R Ossareh. A novel reference governor approach for constraint management of nonlinear systems. *Automatica, under revision*, 2020.

[54] Joycer Osorio and Hamid R. Ossareh. A stochastic approach to maximal output admissible sets and reference governors. In *IEEE Conference on Control Technology and Applications, CCTA 2018, Copenhagen, Denmark, August 21-24, 2018*, pages 704–709, 2018.

[55] Yudan Liu, Joycer Osorio, and Hamid R Ossareh. Decoupled reference governors: A constraint management technique for mimo systems. *Automatica, under revision*, 2020.

[56] Y. Liu, J. Osorio, and H. Ossareh. Decoupled reference governors for multi-input multi-output systems. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 1839–1846, Dec 2018.

[57] Yudan Liu, Joycer Osorio, and Hamid R Ossareh. Preview reference governors. In *2020 IEEE Conference on Decision and Control (CDC)*, 2020.

[58] Petros. Kapasouris, Michael. Athans, Günter. Stein, and United States. *Design of feedback control systems for unstable plants with saturating actuators [microform] / by Petros Kapasouris, Michael Athans, Gunter Stein*. National Aeronautics and Space Administration Washington, DC, 1988.

[59] Elmer G Gilbert, Ilya Kolmanovsky, and Kok Tin Tan. Discrete-time reference governors and the nonlinear control of systems with state and control constraints. *International Journal of robust and nonlinear control*, 5(5):487–504, 1995.

[60] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: the theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, Sep 1991.

[61] E. G. Gilbert, I. Kolmanovsky, and Kok Tin Tan. Nonlinear control of discrete-time linear systems with state and control constraints: a reference governor with global convergence properties. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 1, pages 144–149 vol.1, Dec 1994.

[62] Alberto Bemporad and Edoardo Mosca. Nonlinear predictive reference filtering for constrained tracking. In *Proc. European Control Conf.*, pages 1720–1725, 1995.

[63] A. Bemporad, A. Casavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transactions on Automatic Control*, 42(3):340–349, Mar 1997.

[64] A. Casavola, E. Mosca, and D. Angeli. Robust command governors for constrained linear systems. *IEEE Transactions on Automatic Control*, 45(11):2071–2077, Nov 2000.

[65] Elmer G. Gilbert and Chong-Jin Ong. Constrained linear systems with hard constraints and disturbances: An extended command governor with large domain of attraction. *Automatica*, 47(2):334 – 340, 2011.

[66] Uroš Kalabic. *Reference governors: Theoretical extensions and practical applications*. PhD thesis, University of Michigan, 2015.

[67] U. KalabiÄ, I. Kolmanovsky, and E. Gilbert. Reference governors for linear systems with nonlinear constraints. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 2680–2686, Dec 2011.

[68] L. Chisci, P. Falugi, and G. Zappa. Predictive tracking control of constrained nonlinear systems. *IEE Proceedings - Control Theory and Applications*, 152(3):309–316, May 2005.

[69] David Angeli, Alessandro Casavola, and Edoardo Mosca. Command governors for constrained nonlinear systems: direct nonlinear vs. linearization-based strategies. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 9(10):677–699, 1999.

[70] M. M. Nicotra and E. Garone. Explicit reference governor for continuous time nonlinear systems subject to convex constraints. In *2015 American Control Conference (ACC)*, pages 4561–4566, July 2015.

[71] M. M. Nicotra, R. Naldi, and E. Garone. A robust explicit reference governor for constrained control of unmanned aerial vehicles. In *2016 American Control Conference (ACC)*, pages 6284–6289, July 2016.

[72] Jing Sun and I. V. Kolmanovsky. Load governor for fuel cell oxygen starvation protection: a robust nonlinear reference governor approach. *IEEE Transactions on Control Systems Technology*, 13(6):911–920, Nov 2005.

[73] M. M. Nicotra and E. Garone. Control of euler-lagrange systems subject to constraints: An explicit reference governor approach. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 1154–1159, Dec 2015.

[74] Marco M Nicotra, Mihovil Bartulovic, Emanuele Garone, and Bruno Sinopoli. A distributed explicit reference governor for constrained control of multiple uavsâ. *IFAC-PapersOnLine*, 48(22):156–161, 2015.

[75] I. Kolmanovsky and Jing Sun. Parameter governors for discrete-time nonlinear systems with pointwise-in-time state and control constraints. In *Proceedings of the 2004 American Control Conference*, volume 4, pages 3075–3081 vol.4, June 2004.

[76] Ilya V. Kolmanovsky and Jing Sun. Parameter governors for discrete-time nonlinear systems with pointwise-in-time state and control constraints. *Automatica*, 42(5):841 – 848, 2006.

[77] T. Hatanaka and K. Takaba. Output feedback reference governor for nonlinear systems. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 7558–7563, Dec 2005.

[78] Eelco Scholte and Mark E Campbell. A nonlinear set-membership filter for on-line applications. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 13(15):1337–1358, 2003.

[79] V. Tsourapas, J. Sun, and A. Stefanopoulou. Incremental step reference governor for load conditioning of hybrid fuel cell and gas turbine power plants. *IEEE Transactions on Control Systems Technology*, 17(4):756–767, July 2009.

[80] B. Pluymers, J. A. Rossiter, J. A. K. Suykens, and B. De Moor. The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 804–809 vol. 2, June 2005.

[81] Eric Colin Kerrigan. *Robust constraint satisfaction: Invariant sets and predictive control*. PhD thesis, University of Cambridge, 2001.

[82] E. Klintberg, M. Nilsson, A. Gupta, L. J. MÄLrdh, and P. Falcone. Tree-structured polyhedral invariant set calculations. *IEEE Control Systems Letters*, 4(2):426–431, April 2020.

[83] M Herceg, M Kvasnica, C Jones, and M Morari. Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, 2013.

[84] S. Miani and C. Savorgnan. Maxis-g: a software package for computing polyhedral invariant sets for constrained lpv systems. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 7609–7614, Dec 2005.

[85] N. Athanasopoulos and G. Bitsoris. Invariant set computation for constrained uncertain discrete-time linear systems. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5227–5232, Dec 2010.

[86] Morten Hovd, Sorin Olaru, and George Bitsoris. Low complexity constraint control using contractive sets. *IFAC Proceedings Volumes*, 47(3):2933–2938, 2014.

[87] S. Sheer and P. Gutman. A novel approach to the computation of polyhedral invariant sets for constrained systems. In *2016 IEEE Conference on Computer Aided Control System Design (CACSD)*, pages 1428–1433, Sep. 2016.

[88] Pengyuan Zheng, Dewei Li, Yugeng Xi, and Jun Zhang. Improved model prediction and rmpc design for lpv systems with bounded parameter changes. *Automatica*, 49(12):3695–3699, 2013.

[89] B Pluymers, JAK Suykens, and B De Moor. Construction of reduced complexity polyhedral invariant sets for lpv systems using linear programming. *Automatica,Submitted for publication*, 2005.

[90] Mark Cannon, Shuang Li, and Qifeng Cheng Basil Kouvaritakis. Efficient robust output feedback mpc. *IFAC Proceedings Volumes*, 44(1):7957–7962, 2011.

[91] F. Scibilia, R. R. Bitmead, S. Olaru, and M. Hovd. Maximal robust feasible sets for constrained linear systems controlled by piecewise affine feedback laws. In *2009 IEEE International Conference on Control and Automation*, pages 104–109, Dec 2009.

[92] Y. Chen, H. Peng, J. Grizzle, and N. Ozay. Data-driven computation of minimal robust control invariant set. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4052–4058, Dec 2018.

[93] A. Chakrabarty, A. Raghunathan, S. Di Cairano, and C. Danielson. Data-driven estimation of backward reachable and invariant sets for unmodeled systems via active learning. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 372–377, Dec 2018.

[94] R. E. Allen, A. A. Clark, J. A. Starek, and M. Pavone. A machine learning approach for real-time reachability analysis. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2202–2208, Sep. 2014.

[95] Chong Jin Ong, Dan Sui, and Elmer G Gilbert. Enlarging the terminal region of nonlinear model predictive control using the support vector machine method. *Automatica*, 42(6):1011–1016, 2006.

[96] Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[97] Lin Xie, Serge Shishkin, and Minyue Fu. Piecewise lyapunov functions for robust stability of linear time-varying systems. *Systems & Control Letters*, 31(3):165–171, 1997.

[98] AL Zelentsovsky. Nonquadratic lyapunov functions for robust stability analysis of linear uncertain systems. *IEEE Transactions on Automatic Control*, 39(1):135–138, 1994.

[99] Franco Blanchini. Ultimate boundedness control for uncertain discrete-time systems via set-induced lyapunov functions. *IEEE Transactions on automatic control*, 39(2):428–433, 1994.

[100] Jamal Daafouz and Jacques Bernussou. Parameter dependent lyapunov functions for discrete time systems with time varying parametric uncertainties. *Systems & control letters*, 43(5):355–359, 2001.

[101] D. Y. Rubin, H. Nguyen, and P. Gutman. Yet another algorithm for the computation of polyhedral positive invariant sets. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pages 698–703, Aug 2018.

[102] B. Pluymers, M. V. Kothare, J. A. K. Suykens, and B. De Moor. Robust synthesis of constrained linear state feedback using lmis and polyhedral invariant sets. In *2006 American Control Conference*, pages 6 pp.–, June 2006.

[103] M. V. Kothare, V. Balakrishnan, and M. Morai. Robust constrained model predictive control using linear matrix inequalities. In *Proceedings of 1994 American Control Conference - ACC '94*, volume 1, pages 440–444 vol.1, June 1994.

[104] B Pluymers, JA Rossiter, JAK Suykens, and Bart De Moor. A simple algorithm for robust mpc. In *Proceedings of the IFAC World Congress*, 2005.

[105] UroĹĄ V. KalabiÄ, Ilya V. Kolmanovsky, and Elmer G. Gilbert. Reduced order extended command governor. *Automatica*, 50(5):1466 – 1472, 2014.

[106] A. Casavola, E. Mosca, and M. Papini. Predictive teleoperation of constrained dynamic systems via internet-like channels. *IEEE Transactions on Control Systems Technology*, 14(4):681–694, July 2006.

[107] A. Casavola, M. Papini, and G. Franze. Supervision of networked dynamical systems under coordination constraints. *IEEE Transactions on Automatic Control*, 51(3):421–437, March 2006.

[108] K. Kogiso and K. Hirata. Experimental validations of a remote control technique for constrained linear systems using reference governor plus switching control strategy. In *2006 SICE-ICASE International Joint Conference*, pages 5668–5672, Oct 2006.

[109] Stefano Di Cairano, UroĹĄ V. KalabiÄ, and Ilya V. Kolmanovsky. Reference governor for network control systems subject to variable time-delay. *Automatica*, 62:77 – 86, 2015.

[110] S. Di Cairano and I. V. Kolmanovsky. Rate limited reference governor for network controlled systems. In *Proceedings of the 2010 American Control Conference*, pages 3704–3709, June 2010.

[111] S. Di Cairano and I. V. Kolmanovsky. Further developments and applications of network reference governor for constrained systems. In *2012 American Control Conference (ACC)*, pages 3907–3912, June 2012.

[112] Kiminao Kogiso and Kenji Hirata. Reference governor for constrained systems with time-varying references. *Robotics and Autonomous Systems*, 57(3):289 – 295, 2009. Selected papers from 2006 {IEEE} International Conference on Multisensor Fusion and Integration (MFI 2006)2006 {IEEE} International Conference on Multisensor Fusion and Integration.

[113] S. Di Cairano and I. V. Kolmanovsky. Constrained actuator coordination by virtual state governing. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 5491–5496, Dec 2011.

[114] A. Casavola, G. Franze, and M. Sorbara. A supervisory control strategy for fault tolerance enhancement in networked power systems. In *2007 American Control Conference*, pages 4762–4767, July 2007.

[115] A. Casavola, G. FranzÄ¨, and M. Sorbara. Reference-offset governor approach for the supervision of constrained networked dynamical systems. In *2007 European Control Conference (ECC)*, pages 7–14, July 2007.

[116] G. FranzÄ¨ and F. Tedesco. Constrained load/frequency control problems in networked multi-area power systems. *Journal of the Franklin Institute*, 348(5):832 – 852, 2011.

[117] Emanuele Garone, Francesco Tedesco, and Alessandro Casavola. Distributed coordination-by-constraint strategies for networked control systems. *IFAC Proceedings Volumes*, 42(20):144 – 149, 2009.

[118] A. Casavola, E. Garone, and F. Tedesco. A liveliness analysis of a distributed constrained coordination strategy for multi-agent linear systems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 8133–8138, Dec 2011.

[119] Alessandro Casavola, Emanuele Garone, and Francesco Tedesco. Scalability and performance improvement of distributed sequential command governor strategies via graph colorability theory. *IFAC Proceedings Volumes*, 47(3):9400 – 9405, 2014.

[120] K. McDonough and I. Kolmanovsky. Controller state and reference governors for discrete-time linear systems with pointwise-in-time state and control constraints. In *2015 American Control Conference (ACC)*, pages 3607–3612, July 2015.

[121] G. FranzÄ¨ and W. Lucia. A set-theoretic control architecture for constrained switching systems. In *2016 American Control Conference (ACC)*, pages 685–690, July 2016.

[122] G. FranzÄ¨, W. Lucia, and F. Tedesco. A dwell-time based command governor approach for constrained switched systems. In *2015 American Control Conference (ACC)*, pages 1077–1082, July 2015.

[123] T. Taguchi and Y. Ohta. A dual mode reference governor for discrete time systems with state and control constraints. In *2007 IEEE 22nd International Symposium on Intelligent Control*, pages 202–207, Oct 2007.

[124] U. KalabiÄ, Y. Chitalia, J. Buckland, and I. Kolmanovsky. Prioritization schemes for reference and command governors. In *2013 European Control Conference (ECC)*, pages 2734–2739, July 2013.

[125] Emanuele Garone, Francesco Tedesco, and Alessandro Casavola. Sensorless supervision of linear dynamical systems: The feed-forward command governor approach. *Automatica*, 47(7):1294 – 1303, 2011.

[126] Manuel Lanchares, Ilya Kolmanovsky, Anouck Girard, and Denise Rizzo. Reference governors based on online learning of maximal output admissible set. In *ASME 2019 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers Digital Collection, 2019.

[127] Ying Tan, Jian-Xin Xu, Mikael Norrlöf, and Christopher Freeman. On reference governor in iterative learning control for dynamic systems with input saturation. *Automatica*, 47(11):2412–2419, 2011.

[128] Chong Jin Ong, S Sathiya Keerthi, Elmer G Gilbert, and ZH Zhang. Stability regions for constrained nonlinear systems and their functional characterization via support-vector-machine learning. *Automatica*, 40(11):1955–1964, 2004.

[129] Zhaojian Li, Uroš Kalabić, and Tianshu Chu. Safe reinforcement learning: Learning with supervision using a constraint-admissible set. In *2018 Annual American Control Conference (ACC)*, pages 6390–6395. IEEE, 2018.

[130] Adam Nathan Banker, Julia Helen Buckland, Joseph Norman Ulrey, Uros Vojko Kalabic, Matthew John Gerhart, Tobias John Pallett, Ilya Kolmanovsky, and Suzanne Kay Wait. Methods and systems for torque control, November 3 2015. US Patent 9,174,637.

[131] U. V. KalabiÄ, J. H. Buckland, S. L. Cooper, S. K. Wait, and I. V. Kolmanovsky. Reference governors for enforcing compressor surge constraints. *IEEE Transactions on Control Systems Technology*, 24(5):1729–1739, Sept 2016.

[132] Hayato Nakada, Gareth Milton, Peter Martin, Akiyuki Iemura, and Akira Ohata. Application of reference governor using soft constraints and steepest descent method to diesel engine aftertreatment temperature control. *SAE Int. J. Engines*, 6:257–266, 04 2013.

[133] H. Nakada, P. Martin, G. Milton, A. Iemura, and A. Ohata. An application study of online reference governor to boost pressure control for automotive diesel engines. In *2014 American Control Conference*, pages 3135–3140, June 2014.

[134] I. V. Kolmanovsky, E. G. Gilbert, and J. A. Cook. Reference governors for supplemental torque source control in turbocharged diesel engines. In *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, volume 1, pages 652–656 vol.1, Jun 1997.

[135] S. Jade, E. HellstrÄśm, J. Larimore, A. G. Stefanopoulou, and L. Jiang. Reference governor for load control in a multicylinder recompression hcci engine. *IEEE Transactions on Control Systems Technology*, 22(4):1408–1421, July 2014.

[136] K. Zaseck, M. Brusstar, and I. Kolmanovsky. Constraint enforcement of piston motion in a free-piston engine. In *2014 American Control Conference*, pages 1487–1492, June 2014.

[137] L. Albertoni, A. Balluchi, A. Casavola, C. Gambelli, E. Mosca, and A. L. Sangiovanni-Vincentelli. Hybrid command governors for idle speed control in gasoline direct injection engines. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 1, pages 773–778 vol.1, June 2003.

[138] Robert Gaynor, Fabian Mueller, Faryar Jabbari, and Jacob Brouwer. On control concepts to prevent fuel starvation in solid oxide fuel cells. *Journal of Power Sources*, 180(1):330 – 342, 2008.

[139] Salah Laghrouche, Imad Matraji, Fayez Shakil Ahmed, Samir Jemei, and Maxime Wack. Load governor based on constrained extremum seeking for {PEM} fuel cell oxygen starvation and compressor surge protection. *International Journal of Hydrogen Energy*, 38(33):14314 – 14322, 2013.

[140] I. V. Kolmanovsky, E. G. Gilbert, and H. E. Tseng. Constrained control of vehicle steering. In *2009 IEEE Control Applications, (CCA) Intelligent Control, (ISIC)*, pages 576–581, July 2009.

[141] Nazli E. Kahveci and Ilya V. Kolmanovsky. Control design for electromagnetic actuators based on backstepping and landing reference governor. *IFAC Proceedings Volumes*, 43(18):393 – 398, 2010. 5th IFAC Symposium on Mechatronic Systems.

[142] I. Kolmanovsky and E. G. Gilbert. Landing reference governor. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, volume 1, pages 374–375 vol.1, 2001.

[143] James Wilborn and John Foster. Defining commercial transport loss-of-control: A quantitative approach. In *AIAA atmospheric flight mechanics conference and exhibit*, page 4811, 2004.

[144] M. Pachter and R. B. Miller. Manual flight control with saturating actuators. *IEEE Control Systems*, 18(1):10–20, Feb 1998.

[145] Domenico Famularo, Davide Martino, and Massimiliano Mattei. Constrained control strategies to improve safety and comfort on aircraft. *Journal of guidance, control, and dynamics*, 31(6):1782–1792, 2008.

[146] G. FranzĂ¨, M. Mattei, L. Ollio, V. Scordamaglia, and F. Tedesco. A reconfigurable aircraft control scheme based on an hybrid command governor supervisory approach. In *2014 American Control Conference*, pages 1273–1278, June 2014.

[147] Nazli E Kahveci and Ilya V Kolmanovsky. Constrained control of uavs using adaptive anti-windup compensation and reference governors. Technical report, SAE Technical Paper, 2009.

[148] Victor Carlsson and Oskar Sunesson. Reference governor for flight envelope protection in an autonomous helicopter using model predictive control, 2014.

[149] Christopher Petersen, Morgan Baldwin, and Ilya Kolmanovsky. Model predictive control guidance with extended command governor inner-loop flight control for hypersonic vehicles. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, page 5028, 2013.

[150] Alessandro Casavola, Domenico Famularo, Giuseppe Franzé, and Emanuele Garone. Set-points reconfiguration in networked multi-area electrical power systems. *International journal of adaptive control and signal processing*, 23(8):808–832, 2009.

[151] A. Casavola, G. Franze, F. Tedesco, and E. Garone. Distributed coordination-by-constraint strategies in networked multi-area power systems. In *2011 IEEE International Symposium on Industrial Electronics*, pages 1697–1702, June 2011.

[152] Francesco Tedesco and Alessandro Casavola. Fault-tolerant distributed load/frequency supervisory strategies for networked multi-area microgrids. *International Journal of Robust and Nonlinear Control*, 24(8-9):1380–1402, 2014.

[153] Francesco Tedesco and Alessandro Casavola. Fault-tolerant distributed load/frequency coordination strategies for multi-area power microgrids. *IFAC-PapersOnLine*, 48(21):54–59, 2015.

[154] Ilya V Kolmanovsky and Mohammad Haghgooie. Control algorithm for soft-landing in electromechanical actuators, February 17 2004. US Patent 6,693,787.

[155] R. H. Miller, I. Kolmanovsky, E. G. Gilbert, and P. D. Washabaugh. Control of constrained nonlinear systems: a case study. *IEEE Control Systems*, 20(1):23–32, Feb 2000.

[156] Stefano Di Cairano, Abraham Goldsmith, and Scott Bortoff. Mpc and spatial governor for multistage processing machines in precision manufacturing. In *IFAC nonlinear model predictive control conference*, 2015.

[157] Takahiro Iwasa, Kazuhiko Terashima, Nyioh Yong Jian, and Yoshiyuki Noda. Operator assistance system of rotary crane by gain-scheduled h infinity controller with reference governor. In *Control Applications (CCA), 2010 IEEE International Conference on*, pages 1325–1330. IEEE, 2010.

[158] Lars Grüne and Jürgen Pannek. Nonlinear model predictive control. In *Nonlinear Model Predictive Control*, pages 43–66. Springer, 2011.

[159] Wook Hyun Kwon and Soo Hee Han. *Receding horizon control: model predictive control for state models*. Springer Science & Business Media, 2006.

[160] Jan Marian Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.

[161] DQ Mayne and JB Rawlings. Model predictive control: theory and design. *Madison, WI: Nob Hill Publishing, LCC*, 2009.

[162] JA Rachael, A Rault, JL Testud, and J Papon. Model predictive heuristic control: application to an industrial process. *Automatica*, 14(5):413–428, 1978.

[163] Charles R Cutler and Brian L Ramaker. Dynamic matrix control?? a computer control algorithm. In *Joint automatic control conference*, number 17, page 72, 1980.

[164] David M Prett and RD Gillette. Optimization and constrained multivariable control of a catalytic cracking unit. In *Joint Automatic Control Conference*, number 17, page 73, 1980.

[165] David W Clarke, C Mohtadi, and PS Tuffs. Generalized predictive controlâpart i. the basic algorithm. *Automatica*, 23(2):137–148, 1987.

[166] DW Clarke, C Mohtadi, and PS Tuffs. Generalized predictive controlâpart ii. extension and interpretations. *Automatica*, 23(2):149–160, 1987.

[167] Davor Hrovat, Stefano Di Cairano, H Eric Tseng, and Ilya V Kolmanovsky. The development of model predictive control in automotive industry: A survey. In *Control Applications (CCA), 2012 IEEE International Conference on*, pages 295–302. IEEE, 2012.

[168] Daniel Limón, JM Gomesda Silva, Teodoro Alamo, and Eduardo F Camacho. Improved mpc design based on saturating control laws. *European journal of control*, 11(2):112–122, 2005.

[169] Alberto Bemporad, Manfred Morari, Vivek Dua, and Efstratios N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3 – 20, 2002.

[170] Alberto Bemporad, Francesco Borrelli, and Manfred Morari. On the optimal control law for linear discrete time hybrid systems. In *International workshop on hybrid systems: computation and control*, pages 105–119. Springer, 2002.

[171] E.F. Camacho, D.R. Ramirez, D. Limon, D. MuĂąoz de la PeĂąa, and T. Alamo. Model predictive control techniques for hybrid systems. *Annual Reviews in Control*, 34(1):21 – 31, 2010.

[172] Alberto Bemporad and Manfred Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407 – 427, 1999.

[173] R. Goebel, R. G. Sanfelice, and A. R. Teel. Hybrid dynamical systems. *IEEE Control Systems*, 29(2):28–93, April 2009.

[174] Mircea Lazar. Model predictive control of hybrid systems: Stability and robustness, 2006.

[175] Riccardo Scattolini. Architectures for distributed and hierarchical model predictive control–a review. *Journal of process control*, 19(5):723–731, 2009.

[176] Panagiotis D Christofides, Riccardo Scattolini, David Munoz de la Pena, and Jinfeng Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21–41, 2013.

[177] Tamás Keviczky, Francesco Borrelli, and Gary J Balas. A study on decentralized receding horizon control for decoupled systems. In *American Control Conference, 2004. Proceedings of the 2004*, volume 6, pages 4921–4926. IEEE, 2004.

[178] A. Richards and J. P. How. Robust distributed model predictive control. *International Journal of Control*, 80(9):1517–1531, 2007.

[179] A. Bemporad, A. Oliveri, T. Poggi, and M. Storace. Ultra-fast stabilizing model predictive control via canonical piecewise affine approximations. *IEEE Transactions on Automatic Control*, 56(12):2883–2897, Dec 2011.

[180] Boris Houska, Hans Joachim Ferreau, and Moritz Diehl. Acado toolkitâan open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[181] Alexander Domahidi, Eric Chu, and Stephen Boyd. Ecos: An socp solver for embedded systems. In *Control Conference (ECC), 2013 European*, pages 3071–3076. IEEE, 2013.

[182] D Limon Marruedo, T Alamo, and EF Camacho. Input-to-state stable mpc for constrained discrete-time nonlinear systems with bounded additive uncertainties. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 4, pages 4619–4624. IEEE, 2002.

[183] G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel. Nominally robust model predictive control with state constraints. *IEEE Transactions on Automatic Control*, 52(10):1856–1870, Oct 2007.

[184] G. Pin, D. M. Raimondo, L. Magni, and T. Parisini. Robust model predictive control of nonlinear systems with bounded and state-dependent uncertainties. *IEEE Transactions on Automatic Control*, 54(7):1681–1687, July 2009.

[185] J. A. Primbs and C. H. Sung. Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise. *IEEE Transactions on Automatic Control*, 54(2):221–230, Feb 2009.

[186] Jun Yan and Robert R. Bitmead. Incorporating state estimation into model predictive control and its application to network traffic control. *Automatica*, 41(4):595 – 604, 2005.

[187] Basil Kouvaritakis, Mark Cannon, SaĹĄa V. RakoviÄ, and Qifeng Cheng. Explicit use of probabilistic distributions in linear predictive control. *Automatica*, 46(10):1719 – 1724, 2010.

[188] D. Chatterjee, P. Hokayem, and J. Lygeros. Stochastic receding horizon control with bounded control inputs: A vector space approach. *IEEE Transactions on Automatic Control*, 56(11):2704–2710, Nov 2011.

[189] Veronica Adetola, Darryl DeHaan, and Martin Guay. Adaptive model predictive control for constrained nonlinear systems. *Systems & Control Letters*, 58(5):320–326, 2009.

[190] Giancarlo Marafioti, Robert R Bitmead, and Morten Hovd. Persistently exciting model predictive control. *International Journal of Adaptive Control and Signal Processing*, 28(6):536–552, 2014.

[191] Alberto Bemporad. Explicit model predictive control. *Encyclopedia of Systems and Control*, pages 405–411, 2015.

[192] Alessandro Alessio and Alberto Bemporad. A survey on explicit model predictive control. In *Nonlinear model predictive control*, pages 345–369. Springer, 2009.

[193] Chunxiao Wang, Yuqiang Wu, and Jiangbo Yu. Barrier lyapunov functions-based dynamic surface control for pure-feedback systems with full state constraints. *IET Control Theory & Applications*, 2016.

[194] Keng Peng Tee, Beibei Ren, and Shuzhi Sam Ge. Control of nonlinear systems with time-varying output constraints. *Automatica*, 47(11):2511–2516, 2011.

[195] Keng Peng Tee, Shuzhi Sam Ge, and Francis Eng Hock Tay. Adaptive control of electrostatic microactuators with bidirectional drive. *IEEE transactions on control systems technology*, 17(2):340–352, 2009.

[196] Yan-Jun Liu and Shaocheng Tong. Barrier lyapunov functions-based adaptive control for a class of nonlinear pure-feedback systems with full state constraints. *Automatica*, 64:70–75, 2016.

[197] Khoi B Ngo, Robert Mahony, and Zhong-Ping Jiang. Integrator backstepping using barrier functions for systems with multiple state constraints. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 8306–8312. IEEE, 2005.

[198] Ben Niu and Jun Zhao. Barrier lyapunov functions for the output tracking control of constrained nonlinear switched systems. *Systems & Control Letters*, 62(10):963–971, 2013.

[199] Wei He, Zhao Yin, and Changyin Sun. Adaptive neural network control of a marine vessel with constraints using the asymmetric barrier lyapunov function. *IEEE transactions on cybernetics*, 2016.

[200] Mohammad Ali Javaheri Koopaee and Vahid Johari Majd. Controller design with constraint on control input and output for ts fuzzy systems with nonlinear local model. In *Electrical Engineering (ICEE), 2015 23rd Iranian Conference on*, pages 876–881. IEEE, 2015.

[201] Tianping Zhang, Meizhen Xia, and Yang Yi. Adaptive neural dynamic surface control of strict-feedback nonlinear systems with full state constraints and unmodeled dynamics. *Automatica*, 81:232–239, 2017.

[202] Rong Li, Mou Chen, and Qingxian Wu. Adaptive neural tracking control for uncertain nonlinear systems with input and output constraints using disturbance observer. *Neurocomputing*, 235:27–37, 2017.

[203] Glen Chou, Dmitry Berenson, and Necmiye Ozay. Learning constraints from demonstrations, 2018.

[204] Glen Chou, Necmiye Ozay, and Dmitry Berenson. Learning parametric constraints in high dimensions from demonstrations, 2019.

[205] Glen Chou, Necmiye Ozay, and Dmitry Berenson. Learning constraints from locally-optimal demonstrations under cost function uncertainty, 2020.

[206] G. Pannocchia and A. Bemporad. Combined design of disturbance model and observer for offset-free model predictive control. *IEEE Transactions on Automatic Control*, 52(6):1048–1053, June 2007.

[207] Elmer G Gilbert, Ilya Kolmanovsky, and Kok Tin Tan. Discrete-time reference governors and the nonlinear control of systems with state and control constraints. *International Journal of robust and nonlinear control*, 5(5):487–504, 1995.

[208] J. H. Buckland. *Estimation methods for turbocharged spark ignition engines.* PhD thesis, 2009.

[209] Willem Esterhuizen, Tim Aschenbruck, and Stefan Streif. On maximal robust positively invariant sets in constrained nonlinear systems. *ArXiv*, abs/1904.01985, 2019.

[210] Mirko Fiacchini, Teodoro Alamo, and Eduardo F Camacho. On the computation of convex robust control invariant sets for nonlinear systems. *Automatica*, 46(8):1334–1338, 2010.

[211] Manfred Morari and Evanghelos Zafiriou. *Robust process control.* Morari, 1989.

[212] H. R. Ossareh, S. Wisotzki, J. B. Seeds, and M. Jankovic. An internal model control-based approach for characterization and controller tuning of turbocharged gasoline engines. *IEEE Transactions on Control Systems Technology*, pages 1–10, 2019.

[213] Jing Zhou. *Active Safety Measures for Vehicles Involved in Light Vehicle-to-Vehicle Impacts.* PhD thesis, 2009.