Centro Federal de Educação Tecnológica de Minas Gerais
Universidade Federal de São João del-Rei
Graduate Program in Electrical Engineering
Master's Thesis

Álan Crístoffer e Sousa

Command Governor Supervisor Scheme Based On Region Of Attraction

Divinópolis
2020

Álan Crístoffer e Sousa

Command Governor Supervisor Scheme Based On Region Of Attraction

Master's thesis presented to the Federal Centre of Technological Education of Minas Gerais's (Centro Federal de Educação Tecnológica de Minas Gerais) Graduate Program in Electrical Engineering (PPGEL - Programa de Pós-Graduação em Engenharia Elétrica), as a requisite to obtain the title of Master in Electrical Engineering.

Supervisor: Prof. Dr. Valter Júnior de Souza Leite
Federal Centre of Technological Education of Minas Gerais
Co-supervisor: Prof. Dr. Walter Lucia
Concordia University

Divinópolis
2020

(Catalogação - Biblioteca Universitária – Campus Divinópolis – CEFET-MG)

S725c Sousa, Álan Cristoffer e.

Command governor supervisor scheme based on region of attraction. / Álan Cristoffer e Sousa. –Divinópolis, 2020.

62f. : il.

Orientador: Prof. Dr. Valter Júnior de Souza Leite.

Coorientador: Prof. Dr. Walter Lucia.

Área de Concentração: Modelagem e Controle de Sistemas.

Linha de Pesquisa: Sistema de Controle.

Dissertação (Mestrado) – Centro Federal de Educação Tecnológica de Minas Gerais/Universidade Federal de São João del-Rei. Programa de Pós-Graduação em Engenharia Elétrica.

1. Command governor. 2. Discrete-time systems. 3. Switching systems. 4. Lyapunov stability. 5. Region of attraction. I. Leite, Valter Júnior de Souza. II. Lucia, Walter. III. Centro Federal de Educação Tecnológica de Minas Gerais. IV. Universidade Federal de São João del-Rei. V. Título.

CDU: 004.75: 004.414.23

Bibliotecária Maria Inês Passos Pereira Bueno - CRB-6/2805.

Centro Federal de Educação Tecnológica de Minas Gerais
Universidade Federal São João del-Rei
Graduate Program in Electrical Engineering

Master's thesis entitled "Command Governor Supervisor Scheme Based On Region Of Attraction", authored by Álan Crístoffer e Sousa, approved by the examination board consisting of the following professors:

Prof. Dr. Valter Júnior de Souza Leite

Prof. Dr. Márcio Fantini Miranda

Divinópolis

December 2020

To my family, who always supported me in this journey.

Acknowledgments

Thanks,

to my parents, for all their support and trust in me;

to Prof. Valter for all the patience and great guidance;

to Prof. Luis for the conversations and help;

to my fellow Lucas Arantes, who shared this journey with me.

The world always seems brighter when you’ve just made something that wasn’t there before.

Neil Gaiman

Abstract

Physical systems are subject to constraints, like input saturation or physical limits to the states, which also appears on each switched system mode. One way of dealing with them is to apply the Command Governor scheme, which changes the controller's reference to enforce constraints. When the system is a switched one, the problem of stability arises. Switched systems are a common kind of system used to describe non-linear systems by dividing them into linear sections or different modes of operations of the same system, like the different phases an airplane goes through during take-off or landing. However, arbitrarily switching the modes of a switched system can cause instability, requiring a switching rule design. The most commonly used rule is the dwell-time switch, in which the system waits for a dwell-time to elapse after the references changes to switch modes. Seeing the possibility of speeding up such systems' convergence, we propose a new rule based on the controller's Region of Attraction, which requires the system's state to be inside the mode's controller's Region of Attraction to switch, guaranteeing stability after a mode switch. With this technique, we also propose a hybrid switch technique, which can further speed up convergence and generate lower actuator effort in some cases. We present some simulations to illustrate the proposal's potential and compare it with a scheme exploiting a dwell-time approach. The results suggest that our approach adds new CG and supervisor design possibilities, reducing the transition time between system modes and improving the closed-loop performance indexes.

Keywords: Command Governor, Discrete-time systems, Switching systems, Lyapunov stability, Region of attraction

Contents

List of Figures

Notations

CG	Command Governor
$A^\top$	Matrix transpose
$\|x\|_\Psi^2$	$x^\top \Psi x \mid x \in \mathbb{R}^n,\ \Psi \in \mathbb{R}^{n\times n}$
$g(t)$	Continuous time signal
$g(k)$	Discrete time signal
$\geq$	Real number inequality
$\succ$	Definite positive matrix inequality
$\mathbb{R}^{n\times m}$	Matrix of n rows and m columns and real elements
$\mathbb{R}^n$	Vector of n real elements
$\mathbb{R}$	Set of real numbers
$\mathcal{C}$	A set

Chapter 1

Introduction

Physical systems have constraints on their inputs, outputs, and states, which are usually ignored to facilitate the controller's design. Such constraints can result from physical limitations, such as a tank's capacity or a turbine's maximum power output, or can be mathematically imposed to achieve some goal, like keeping a process' pH bounded. In the literature, different solutions exists to address actuator saturation constraints, see e.g. (Klug et al., 2015; Tarbouriech et al., 2011). However, those techniques do not enforce constraints but design controllers that avoid the borders of constraints or even allow the system to violate them for a short period.

By taking advantage of optimization techniques, Model Predictive Control uses an online optimization procedure to predict a system's state evolution on a prediction horizon and compute a system's optimal control trajectory and allows constraints to be imposed to states, inputs, outputs, and their variations (Wang, 2009; Zhang, 2016). As the optimization procedure is online, constraints are enforced, and the controller will actively avoid violations while allowing for less conservative control paths.

Besides MPC, other techniques were developed to keep systems constrained, but instead of computing the optimal control path that keeps the system constrained, it uses model prediction to change the reference given to existing controllers to keep the system constrained. The first of such techniques were reference filters, which imposed only soft-constraints (Vahidi et al., 2007), i.e. constraints which penalize the objective function when constraints are violated instead of rendering the problem infeasible.

This idea then evolved into Reference Governors (RG), which divides the problem into two parts: tracking and constraint enforcement. The controller solves the tracking prob-

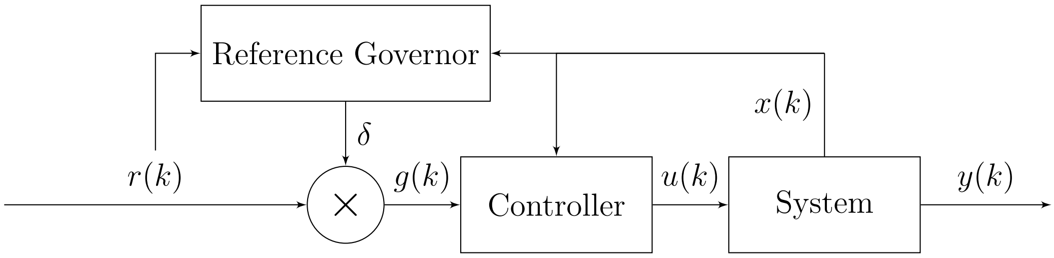


Figure 1.1 – Reference Governor's block diagram for a linear system. The inner loop's controller guarantees reference tracking and the outer loop guarantees constraints satisfaction.

lem in the inner loop without taking constraints into account, and the Reference Governor solves the constraint enforcement problem in the outer loop by using the reference and system's output to change the inner loop's reference. Figure 1.1 shows a block diagram for this technique. The optimization problem finds the best δ that minimizes the distance between $g(k)$ and $r(k)$ without violating constraints. Because of the numerical simplicity of the optimization problem, this approach has an easy implementation but suffers from loss of dimensions. Such a loss comes from the fact that $\delta \in \mathcal{R}$ while $r(k) \in \mathcal{R}^n$ (Gilbert & Kolmanovsky, 1999).

Building on this idea and on the work of Kapasouris et al. (1988), which explores the ideas of the Lyapunov Theorem and Invariant Sets Theorem (Blanchini & Miani, 2008), Bemporad et al. (1997) and Casavola et al. (2000) developed what is known today as the Command Governor (CG) approach. The difference is that Reference Governors optimize a number δ, which multiplies the reference $r(k)$ to create the virtual reference $g(k)$, whereas Command Governors optimize the the vector $g(k)$ directly, requiring more computational processing power but yielding better closed-loop performance.

Reference and Command Governors are still subjects of studies, and used in conjunction with other techniques. It has been of particular interest on the adaptive control field (Arabi et al., 2020; Dogan et al., 2020; Gruenwald et al., 2020; Makavita et al., 2019; Ristevski et al., 2019; Wilcher et al., 2020) as a mean to add constraints to the system. It has also been used to constrain switching networks (Ong et al., 2020), networks with delays (Shen et al., 2019) and interconnected systems (Tedesco & Casavola, 2020). Peng et al. (2019) also used a Command Governor to develop an anti-disturbance controller for an uncertain, constrained system and Schwerdtner et al. (2019) developed an anti-windup controller for systems with saturated inputs. Seeber et al. (2019) provide a real-time implementation of reference shaping for a biomass grate boiler, based on

Command Governor, to avoid actuator saturation and to constrain mass-flow.

The Command Governor can be used in so many different scenarios because it is a add-on technique, not requiring adaptation from the controller or system. This is one main advantage of the Command Governor, as well as the fact that it enforces hard-constraints. The main drawback is that it requires an online optimization procedure, which might not be doable on systems with fast dynamics. Another concern is that the optimization problem might not be feasible in the presence of model uncertainties, as, for example, an observer might put the model's state on an invalid state for the optimization problem. A common technique is to re-apply the last calculated reference value when the optimization problem is unfeasible, but it can lead to instability if the situation persists for extended periods of time (Garone et al., 2017).

This work applies the *CG* technique to switched systems. Switched systems are composed of many subsystems, called modes, which switch according to a switching rule (Liberzon & Morse, 1999; Liberzon, 2003). Only one subsystem can be active at a given time. The switching can cause instability even when all subsystems are stable. Techniques exist to guarantee stability under abirtrary switching, such as using polytopic linear parameter varying representations (Deaecto et al., 2014), but they are extremely conservative. Development leads to the notion of a dwell-time: how long a subsystem must remain active before switching to avoid instability (Liberzon & Morse, 1999). Different approaches have been proposed to compute the minimum dwell time (see (Chesi et al., 2010) and reference therein) and stabilizing controller (see (Lin & Antsaklis, 2009) for switched linear systems). Fewer solutions exist to deal with constrained switching systems, see e.g. (Franzè et al., 2017; Lucia & Franzè, 2017) and references therein.

In (Franzè et al., 2017; Lucia & Franzè, 2017), the *CG* framework is used to supervise the system mode switches and to assure both stability and constraint satisfaction.

1.1 Objectives

In this work, considering a class of constrained switched systems, we propose a novel switching rule based on the controllers's region of attraction. There are other techniques for stabilizing switched systems, but for constrained switched systems, specially when using the Command Governor structure, the dwell-time seems to be the only one applied.

Without employing the Command Governor structure, however, the Model Predictive Control technique accomplishes the same goal. For the proposed technique two rules can be stated:

- the system's state is inside the next controller's region of attraction.
- the system's state is inside the command governors' constraint's intersection.

This allows two switching rules. In the first scenario, the first rule triggers a partial switch, in which only the controller is changed. The second rule is used to complete the switch, swaping the active command governor unit. We call this a hybrid switch. The second scenario makes a complete switch using only the second rule. Both scenarios need the same stability condition, the only difference is that the second avoids the first set membership check, at the cost of possibly missing on convergence's speed gains opportunities.

1.2 Thesis organization

The main concepts involved in this work are explained in Chapter 2 - Theoretical Foundations. The proposed technique is explained in Chapter 3 - Switching Rules. In Chapter 4 - Results we show two experiments that illustrate the advantages of the proposed technique.

Chapter 2

Theoretical Foundations

This work presents a Command Governor-based control strategy, based on the concept of region of attraction, for switched systems. Thus, those are the main concepts involved in this work and will be further explained in the following sections.

2.1 Switched Systems

A switched system is a system composed of many subsystems, called modes, and a mode-transition rule. It is often used to divide non-linear systems into many linear systems around different operation points, creating linear approximations, which are active one at a time. Differently from switching systems, in which mode transition can happen arbitrarily, in the switched system, a supervisor orchestrates them (Liberzon & Morse, 1999; Lucia & Franzè, 2017).

A linear switched system can be described as follows

$$\dot{x}(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t), \tag{2.1}$$

$$y(t) = C_{\sigma(t)}x(t) + D_{\sigma(t)}u(t), \tag{2.2}$$

where $\sigma(t) : t > 0 \to \mathbb{K} = \{1, \ldots, N\}$ is the switching function. A_i, B_i, C_i and D_i are real matrices and represent linearizations around different operation points. On the other hand, there are also nonlinear switched systems with the form

$$\dot{x}(t) = f_{\sigma(t)}(t, x(t), u(t)), \tag{2.3}$$

$$y(t) = g_{\sigma(t)}(t, x(t)), \tag{2.4}$$

but those will not be discussed here. In both cases, $t_k : k \in \mathbb{N}$ is the switching instant.

There are two classes of switching functions: pertubation and control. For the former, given $J(\sigma)$ a switching criterion and $\mathcal{D}_T = \sigma(\cdot) : t_{k+1} - t_k \geq T \; \forall \; k \in \mathbb{N}$, we solve

$$\sigma = \sup_{\sigma \in \mathcal{D}_T} J(\sigma), \tag{2.5}$$

and for the later, which is the one of interest for us, given $\mathcal{S}$ the set of all stabilizing switching functions, we solve

$$\sigma = \inf_{\sigma \in \mathcal{S}} J(\sigma). \tag{2.6}$$

This is, however, a very broad definition. In plain english, the pertubation switching function finds the worse-case scenario (the admissible pertubation depends on the magnitude of T) and the control the best-case. To define stabilizing switching functions, let us analyse the stability of switched systems.

The linear switched system is not guaranteed to be stable even if all subsystems are stable (Liberzon & Morse, 1999). This happens because the switching signal itself is a source of instability, as the subsystems might behave differently and diverge or start cycling under constant switching. This can also be seen if you consider the set $\{A_1, A_2, \ldots, A_N\}$ to be the vertices of a polytope, as the system will be instantaneously jumping between the vertices (Geromel & Colaneri, 2005).

If we consider the system to be a polytope and can find $P \succ 0$ such that

$$A_i^\top P + PA_i \prec 0, \tag{2.7}$$

then the system is stable under arbitrary switching (Geromel & Deaecto, 2014). However, this is a conservative solution and might yield very small regions of attraction. A more desirable result is to have a mode dependent Lyapunov cadidate function leading to

$$A_i^\top P_i + P_i A_i \prec 0, \tag{2.8}$$

which is, however, only stable under arbitrary switching, from any mode i to any mode j, if

$$\begin{align}
V(x(t_{k+1})) &= x^\top(t_{k+1}) P_j x(t_{k+1}) \tag{2.9}\\
&= x^\top(t_k) e^{A_i^\top T_k} P_j x e^{A_i T_k}(t_k) \tag{2.10}\\
&< x^\top(t_k) e^{A_i^\top (T_k - T)} P_i x e^{A_i (T_k - T)}(t_k) \tag{2.11}\\
&< x^\top(t_k) P_i x(t_k) \tag{2.12}\\
&= V(x(t_k)), \tag{2.13}
\end{align}$$

implying that all subsystems must be stable (Geromel & Colaneri, 2005).

The solutions discussed so far allow arbitrary switching, but a large class of systems will not satisfy the required condition. If the switching is assumed to not be arbitrary, one can create a switching rule that will guarantee stability. There is a class of such rules called slow-switching. The dwell-time is a rule of this class requiring $t_{k+1} - t_k > T$, that is, the system must remain on a mode for T seconds before switching again. This time is counted from the moment the reference changes. The goal is to minimize T, minimizing the time the system has to wait (Chesi et al., 2010; Franzè et al., 2017; Liberzon & Morse, 1999). Note that there is one dwell-time T for each mode.

Other ways of guaranteeing switching stability exists. See Geromel and Colaneri (2005), Geromel and Deaecto (2014), and Liberzon and Morse (1999).

2.2 Command Governor

2.2.1 System Description

A linear, switched, discrete-time system given in the state-space representation can be described as follows

$$
\begin{aligned}
x(k+1) &= A_i x(k) + B_i u(k),\\
y(k) &= C_i x(k) + D_i u(k),\\
c(k) &= E_i x(k) + F_i u(k),
\end{aligned}
\tag{2.14}
$$

where $x(k) \in \mathbb{R}^n$ is the state vector, $y(k) \in \mathbb{R}^p$ is the output, $c(k) \in \mathbb{R}^{n_c}$ is the constrained or weight output, the matrices $A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{n \times n_u}$, $C_i \in \mathbb{R}^{n_y \times n}$, and $D_i \in \mathbb{R}^{n_y \times n_u}$ concern the system's dynamic and output, matrices $E_i \in \mathbb{R}^{n_c \times n}$ and $F_i \in \mathbb{R}^{n_c \times n_u}$ concern the constrained output and are chosen by the designer to take into account the constraints associate with the state and control signal. The sub-index $i = 1, \ldots, N$ refers to the active model.

2.2.2 Command Governor

The Command Governor (CG) is an add-on technique that extends existing controllers with constraint enforcement. It uses the system model to predict states given a reference $r(k)$ and computes the virtual reference $g(k)$ closest to $r(k)$ that keeps the system constrained. Figure 2.1 presents a block diagram containing a single CG unit.

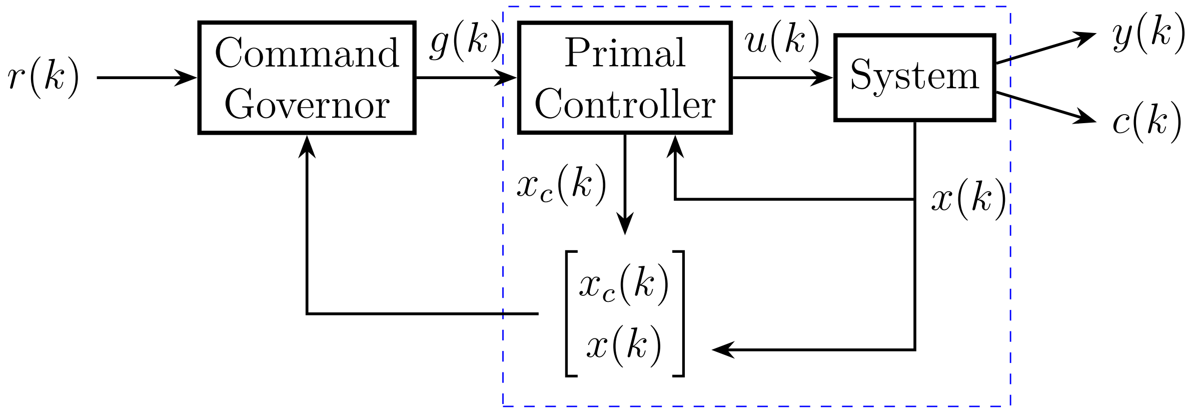


Figure 2.1 – Command Governor Block Diagram. The system's and controller's states are fedback to constraint the output $c(k)$.

The process diagram in Figure 4.1 is simplified here as the dashed blue square. The reference is not an input of the process, but of the *CG* and the output of the *CG* is the input of the process. The *CG* block has two inputs: the reference and the augmented state vector, a concatenation of the controller's state $x_c(k)$ and the system's state $x(k)$.

An optmization procedure, described below, calculates the reference $g(k)$ which keeps the system output $c(k)$ within a predefined region, called the constraint. In the system defined in Equation (2.14) the constraint is applied to a linear combination of the states and inputs of the system. $c(k)$ is not fed back to the *CG* because the information needed to calculate it is already used during the design of the *CG* logic, so it can be reconstructed from the augmented vector.

In what follows we use the sets $\mathcal{C}$, $\mathcal{W}$, and $\mathcal{V}(x(k))$ defined as follows:

1. $\mathcal{C}$ is the set of restrictions, which contains all allowed values of $c(k)$.

2. $\mathcal{W}$ is the set of all constant references ω that keep $c(k)$ constrained on the steady-state.

$$\mathcal{W} = \{\omega \in \mathbb{R}^{n_y} : c(k) \in \mathcal{C}, k \to \infty\}.$$

3. $\mathcal{V}(x(k))$ is the set of ω values which ensures that the constrained output belongs to $\mathcal{C}$ during the transitory

$$\mathcal{V}(x(k)) = \{\omega \in \mathcal{W} : c(k) \in \mathcal{C}, 0 < k \leq k_0\},$$

We write the set $\mathcal{V}(x(k))$ in terms of convex constraints on the virtual reference, the k_0 future states, the minimization of the distance to the desired reference, and actuator saturation bound.

The output of the Command Governor is given by the following convex optimization procedure:

$$g(k) = \underset{\omega \in \mathcal{V}(x(k))}{\arg\min} \|\omega - r(k)\|_{\Psi}^{2} \tag{2.15}$$

where Ψ is a semi-definite positive weighing matrix and $g(k)$ is the virtual reference closest to the desired reference, not allowing the system to violate the constraints. Observe that the set $\mathcal{V}(x(k))$ is defined by evolving the system in closed-loop, without disturbances, a number k_0 of samples ahead, and testing if the state sequence do not violate constraints.

2.2.3 Supervisor

The supervisor is responsible for selecting which i-th CG is currently active. Figure 2.2 shows its block diagram, in which we see that all CGs are continuously updated, but only CG_3 is active, sending control signals to the system. The policy for switching CGs is problem-dependent. It might be another controller giving better performance, according to some metric, or being able to access a state-space region. Note that each CG unit has its $\mathcal{C}_i$, $\mathcal{W}_i$ and $\mathcal{V}(x(k))_i$ sets.

However, even if the policy allows the change, it will not be admissible if $x(k) \notin \mathcal{X}_i \cap \mathcal{X}_j$, where

$$\mathcal{X}_i = \left\{ x \in \mathbb{R}^n | \begin{bmatrix} \omega \\ x \end{bmatrix} \in \mathcal{Z}_i \text{ for at least one } \omega \in \mathbb{R}^{n_y} \right\} \tag{2.16}$$

is the set of all states that can be steered to an equilibrium point without constraint violation, with

$$\mathcal{Z}_i = \left\{ \begin{bmatrix} r \\ x \end{bmatrix} \in \mathbb{R}^{n_y+n} | c(k) \in \mathcal{C}_i, \forall k \in \mathbb{Z}^+ \right\} \tag{2.17}$$

being the admissible output set.

A switched system must have many CG units, each of them with its constraint region. A generic CG switch from CG_i to CG_j, denoted $CG_i \to CG_j$, is admissible (with respect to the plant constraints) if the two CG's domains have a nonempty intersection. If one wants to go from CG_1 to CG_3 and there is no intersection, one must find a path through other CGs with nonempty intersections. Therefore the union of all CG's domains cannot be a disconnected domain.

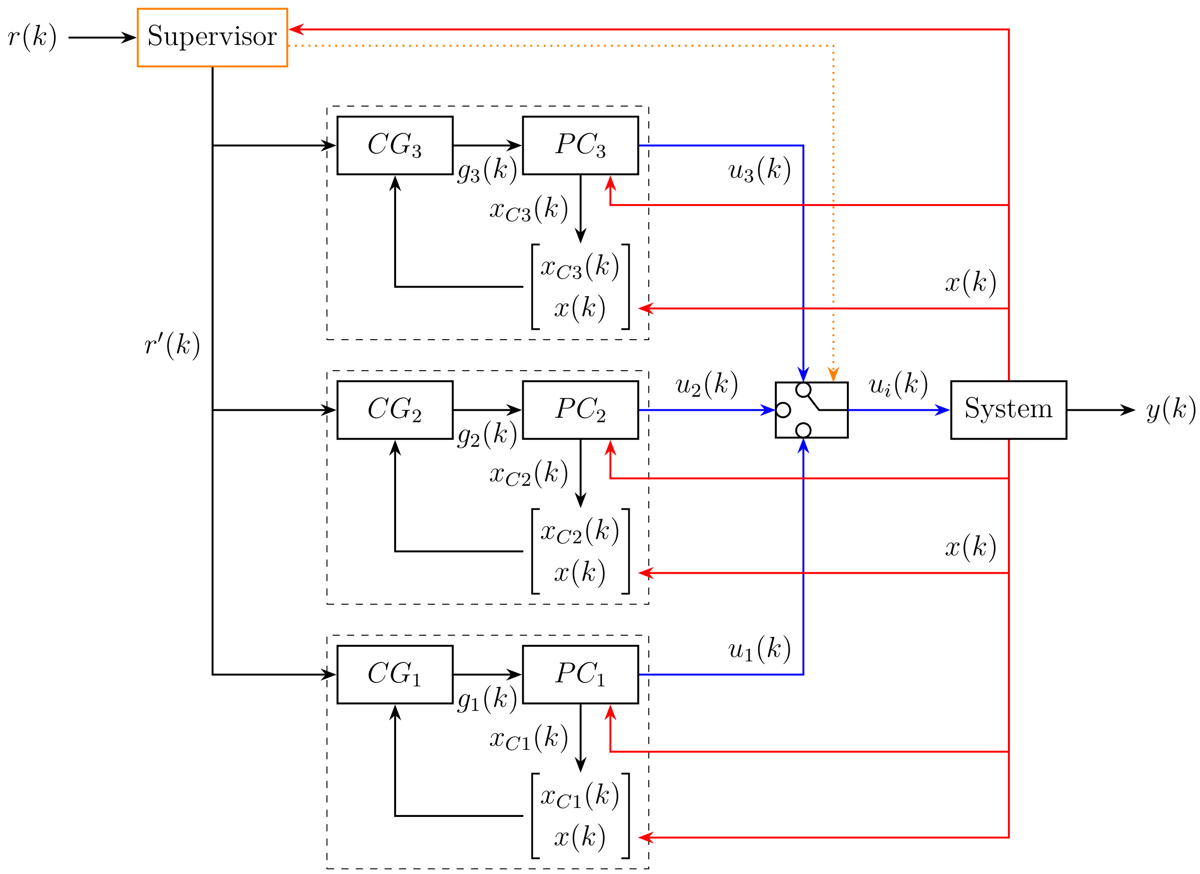


Figure 2.2 – Supervisor Block Diagram. Each mode has its corresponding Command Governor unit. The Supervisor uses the system's state and reference to generate an intermediary reference to the Command Governors, allowing the system to follow a path between modes. The intermediary references are called waypoints. The Command Governors generate references for their controllers and the Supervisor only allows the current active one to send its control signal to the system.

To change the active CG, one needs to go to a point interior to the CG' domains intersection, called way-point. Such a point can be chosen arbitrarily inside the intersection of regions defined by the constraints of the concerned CGs. However, it is better to use points in the central area, not so close to the borders, to avoid problems involving disturbances and controller sensitivity (Keel & Bhattacharyya, 1997). All the paths from one CG to another can be calculated offline, for example, using graph theory(Ahuja et al., 1990; Pettie, 2004). Such an aspect is now handled in this work and, then, the path is assumed to be known.

In (Franzè et al., 2017; Lucia & Franzè, 2017) it has been shown that any CG switch, e.g. $CG_i \rightarrow CG_j$ can be safely accomplished (not violating the constraints) from any point belonging to the intersection between the two CGs' domains. Therefore, a possible way to achieve a safe switch is to define a waypoint reference in the intersection set. Once the plant trajectory, under the action of CG_i is confined in the intersection region, then CG_j is activated.

This approach is a way of guaranteeing stability, as the dwell-time is calculated to allow the switching to occur neither too soon nor too often, giving enough time to the current controller to stabilize the system before switching again. It is, however, a very conservative approach that assumes a worst-case scenario. Although it might be necessary to stabilize some systems, it is not always needed and may lead the convergence time to take longer.

2.3 Region of Attraction

The Region of Attraction is a definition which initiated with the work of the French mathematician Henri Poincaré. He studied real, nonlinear differential equations' behaviour, finding that the states' evolution in time is stable, unstable or cyclic. However, he did not find proof, which was later given by the Swedish mathematician Ivar Bendixson (Bendixson, 1901).

Differential equations which describe real-world dynamics needs to satisfy a condition, which imposes a bound to its first derivative. In other words, the rate of change of the differential equation must be limited. This is a special form of function continuity called Lipschitz continuity. All differential equations that represent the dynamics of a real-world, physical system satisfy this condition. The condition is

$$\left\| \frac{\partial f(x,t)}{\partial x} \right\| \leq L, \tag{2.18}$$

where L is a real, positive constant, the rate of change. This condition is important as it guarantees that the differential equation $\dot{x} = f(x,t)$ with initial condition $x(t_0) = x_0$ has a unique solution for every δ in $[t_0, t_0 + \delta]$ (Donchev & Farkhi, 1998). In other words, the differential equation has only unique solutions.

For equations satisfying this condition, Bendixson (1901) presents the definition of a region with interesting properties. First, he defines the existence of a critical point for differential functions, which are points such that the differential function's derivative are zero and remain zero as the time goes to infinity. He then shows that there are two types of critical points: stable and unstable. A stable critical point is such that all states on a given neighbourhood converge to the point as time goes to infinity. For an unstable critical point, they diverge. He then proceeds to show the existence of limit-cycles, which are not critical points, but, for a differential equation $f(t, x, y)$, $f(t, x, y) = f(t - t_0, x, y)$, meaning that variables (x, y), the state, are repeating thenselves cyclic every $t - t_0$ time units. Cycles can be classified as stable or unstable by taking any of its points and applying the same analysis as for critical points, with the addition that it may be semi-stable, meaning that points inside the region convert and outside diverge, or vice-versa.

To verify the type of a critical point, one can take the derivative of the function and apply at the point. For a continuous-time differential equation, the resulting values need to be negative for the point to be a stable critical point, otherwise it is an unstable critical point. For a discrete-time system, their absolute values must be smaller than one for the point to be stable. Limit-cycles are only possible around critical points, so to test for their existence on a function $f(x)$, one must analyze the stability of the critical-point and then define a region $H(x)$, which contains the critical point, and calculate the gradient (∇) of $H(x)$ on $f(x)$. The following list sumarizes the possible outcomes:

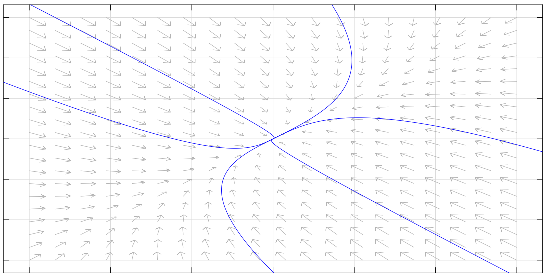
1. if the point is **unstable** and $\nabla H(x)f(x) < 0$, then there is a **stable cycle** inside the region $H(x)$;

2. if the point is **unstable** and $\nabla H(x)f(x) > 0$, then there is an **unstable cycle** inside the region $H(x)$ **or no cycles**;

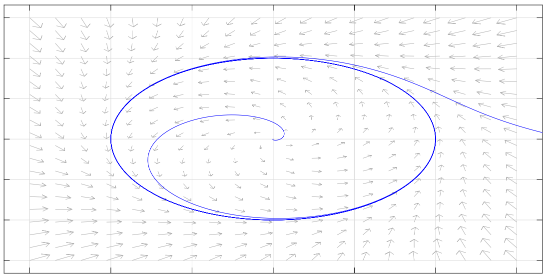
3. if the point is **stable** and $\nabla H(x)f(x) < 0$, then there is a **semi-stable cycle** inside the region $H(x)$ or **no cycles**;

4. if the point is **stable** and $\nabla H(x)f(x) > 0$, then there is a **semi-stable cycle** inside the region $H(x)$.

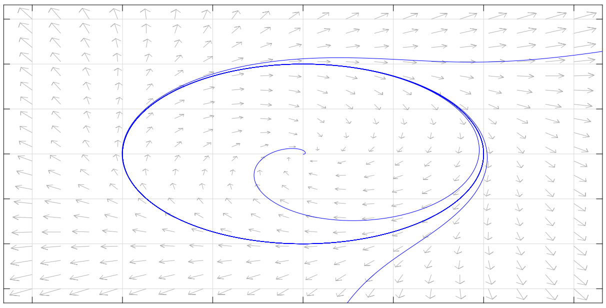
Figure 2.3 shows some possible gradients. In 2.3c there is a stable point and a semi-stable cycle. Notice how all points inside the cycle converge to the origin (center of the figure) and all points outside diverges. The cycle exists but is not stable, as the state might either converge or diverge after some time. In 2.3b the gradients are the opposite of the previous case, resulting on a sustained cycle. Every initial state will converge to the cycle and the state will keep looping forever. In 2.3a there is no cycle, but there could be one. This is the most common case, specially when working with linearizations. All states converge to the origin, regardless of the initial state. Note that we do not know what happens outside the plotted region. There could be a cycle hidden there, or the gradient might change completely, so we cannot make any affirmation about the system's behaviour outside the plotted region. You could call this region $H(x)$ and it would be a Region of Attraction.



(a) Stable point



(b) Stable cycle



(c) Semi-stable cycle

Figure 2.3 – Different cycles' gradient maps. In 2.3a there is a stable critical point and all values converge to it. In 2.3b there is an unstable critical point and stable limit-cycle, so all values converge to the limit-cycle and remain cycling on its border. The vector field in 2.3c is the contrary of that on the previous subfigure, and the values will only stay on the limit-cycle if they start there and there is no pertubation or noise, otherwise they will either converge to the critical point or diverge.

Definition 1 *A Region of Attraction is the neighbourhood of a critical point which is forward invariant under the flow generated by it (Milnor, 1985, p. 178).*

This definition allows limit-cycles to be inside the region of attraction, which is undesirable for control purposes, where we expect the system to converge, leading to a more restricted defition. Lyapunov's stability criteria states that for the system $\dot{x} = f(t, x)$, if there is a function $V(x)$ such that

$$V(x) = 0 \iff x = 0, \tag{2.19}$$

$$V(x) > 0 \iff x \neq 0, \tag{2.20}$$

$$V(x_1) > V(x_2) \iff x_1 > x_2 \tag{2.21}$$

$$\dot{V}(x) = \nabla V(x) f(x) \leq 0 \ \forall x \neq 0, \tag{2.22}$$

then the system is stable (Chen, 2012; Hespanha, 2018). Futhermore, if $\dot{V}(x) < 0$, the system is asymptotically stable. In this context, $V(x)$ is an energy function, but its derivative reminds us of Poincaré's theorem. In fact, Lyapunov's function can be seen as a more restricted form or Poincaré's region, which always contains stable points or stable cycles inside it. If the derivative is strictly negative, the region $V(x)$ is guarantee to not contain cycles (Chen, 2012). Because of this link between the two theorems, Lyapunov's stability criteria is often used to estimate the region of attraction.

In case of a linear system described by $\dot{x} = Ax$, it is necessary and sufficient to search for a function $V(x) = x^\top Px$. It is the most used candidate since it is easy to verify its positiviness: $V(x) \geq 0 \ \forall x$ if P is SDP (semidefinte positive) (Bochnak et al., 1998). It is also a simple region to describe and derive, making it easy to use with LMI tools. Note, however, that there are infinite possible regions, and $V(x) = x^\top Px$ is most certainly *not* the largest region, making it a conservative solution.

Chapter 3

Switching Rules

This chapter presents two switching rules: the dwell-time, the most employed and studied of such techniques in literature, and the proposed rule based on the region of attraction. Other techniques exist, like using robust control to design a controller that never allows the system to unstabilize; however, they are more conservative and not viable in many cases.

3.1 Dwell-time

As discussed in Section 2.1, the act of switching can cause instability, even if all modes are stable. Although it is possible to verify if the system is stable under arbitrary switching (and therefore to design controllers to do so), the procedure is not straightforward and challenging to apply to most real-world situations.

There are, however, other ways of verifying and guaranteeing the stability of switched systems. The dwell-time is one of such techniques that restricts the switching signal $\sigma(t)$ to the set

$$\mathcal{D}_T := \{\sigma(\cdot) : t_{k+1} - t_k \geq T\}, \tag{3.1}$$

where t_k and t_{k+1} are the switching instants, for all $k \in \mathbb{R}$, which forces the system to remain for T seconds on a mode before switching to next one (Colaneri, 2009). This falls in the class of slow-switching rules. The timer is restarted every time the reference changes and switching is only allowed after T seconds has passed. For large enough values of T, this rule guarantees the system's stability.

As the dwell-time certifies stability of the switch, it decouples the switching logic and the system stability, making it possible to analyse the system stability for each mode independently. One problem, however, is that computing the minimum dwell-time is not easy and is the focus of current research. An easier problem is to find an upper bound for it, which can be done efficiently using numerical algorithms (Colaneri, 2009).

Another technique that uses the concept of dwell-time is the average dwell-time, where the switching rule σ allows for a fixed number of discontinuities $N_\sigma(t,\tau)$ for $t \geq \tau \geq 0$ such that the set $\mathcal{D}_{T_D,N_0}$ satisfies

$$N_\sigma(t,\tau) \leq N_0 + \frac{t-\tau}{\tau_D}, \tag{3.2}$$

where τ_D is the average dwell-time and N_0 is the chatter bound (Hespanha & Morse, 1999). This set is larger than $\mathcal{D}_T$ and allows for signals with discontinuities separated by at most τ_D.

To illustrate how the dwell-time works, consider the state-space of a fictional two-state system shown in Figure 3.1. $\bullet$ marks the system's initial state, $\diamond$ are the waypoints and $\star$ is the reference. The collored ellipses are each mode's state's constraints, numbered 1, 2, 3 from left to right. Since there are three ellipses, we have three modes, meaning three command governors, controllers and linearized systems. The waypoints are intermediate references, chosen to allow the system to travel from the current state ($\bullet$) to the reference ($\star$) without violating any constraint.

Consider that the system is on steady-state at $\bullet$ when the reference switched to $\star$. The supervisor scheme used here is the one given in Figure 2.2. The supervisor will receive the new reference $r(k)$ and, because the system is currently on mode 1, will set $r'(k)$ to the first waypoint's coordinates, since it is not possible to gro from $\bullet$ to $\star$ directly without violating the constraints.

Because a reference change occurred, the supervisor will start a timer, counting down from the first mode's dwell-time, T_1. Even if $r(k)$ changes again, $r'(k)$ will not be changed until this timer expires. This gives the first mode's controller enough time to converge to the first waypoint, guaranteeing stability.

Because the waypoint is at mode 1's and mode 2's constraints intersection, the supervisor is allowed to change the the system to the second mode and change the reference $r'(k)$ to the second waypoint's coordinates, as soon as the dwell-time's timers expire (both changes happen simultaniously). Again, because the reference changed, a new timer will

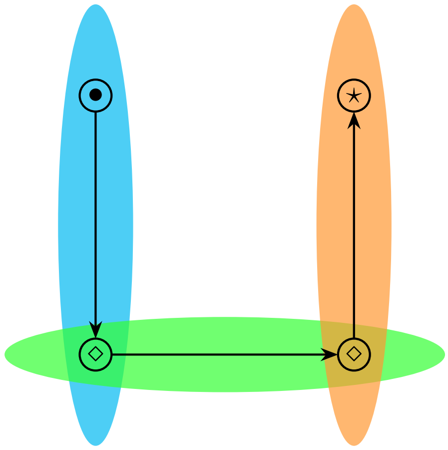


Figure 3.1 – Dwell-time illustrative example. The plane is the phase-plane of a second order system. Each colored region represents one mode's contraints, each symbol inside a circle represents a point of interest: $\bullet$ is the initial state, $\diamond$ are the waypoints and $\star$ the final reference, as set by the operator. The arrows show the path the system will take as the supervisor changes the references to the waypoints and then to $\star$.

start, but now counting down from mode 2's dwell-time, T_2. The same procedure will be repeated once the system's state reaches the second waypoint, when $r'(k)$ will finally be set to $r(k)$.

Algorithm 1 presents a generalized version of the algorith described in the given example.

Algorithm 1 dwell-time implementation

```
Input: CG_i ← current CG, CG_j ← target CG.
change r'(k) to next way-point
while dwell-time of CG_i not ellapsed do
    calculate g(k)
    execute controller
end while
change to CG_j
restart algorithm
```

Firstly, CG_i is set to the currently active CG and CG_j to the next CG in the path (briefly disscussed in Subsection 2.2.3). Then the reference $r'(k)$ is set to the next waypoint in the path. The timer is started and the supervisor will wait for it to ellapse before making any change to the system. While it is not ellapsed, the active CG and controller will drive the system to the reference $r'(k)$. As soon as the time ellapses, the system switches modes, activating CG_j. The procedure is then repeated.

3.2 Region of Attraction

The goal of this switching rule is to allow the system to converge faster to the final reference when going through a path of restricted mode switches. To achieve this goal, it is necessary to guarantee stability after switching modes and switch modes as soon as possible.

The main contribution of this work is to propose a switching rule based on the controller's region of attraction yielding a guaranteed stable closed-loop system. As described in Section 2.3, the controller's region of attraction is the forward invariant neighbourhood under the flow generated by a critical point. On a switched system, there is one controller and linearized system for each mode, therefore there is also one region of attraction for each mode.

Since the definition of a Region of Attraction undesirably allows the existence of limit-cycles within the region, we turn to Lyapunov's theorem to further restrict the region, eliminating them. The set which represents the region of attraction becomes *contractive*, becoming smaller as the system's state converges to the critical point. The Lyapunov function is seen as an energy function, and its value at some point in the state-space is an energy level. This leads to the so called level sets, which are all points that yield the same energy level, in Lyapunov's sense.

The level set, denoted $\mathcal{L}_V(P)$, where P is the matrix used in the function $V(x) = x^\top Px$, is an estimative of the region of attraction and, being contractive, implies that, once the system reaches a lower energy level, in Lyapunov's sense, it can not go to a state with higher energy and can only stay at the same level if it is zero, which guarantees convergence, as the only point with zero energy is the origin of the linear system.

The region of attraction is, therefore, a certificate of stability for a system. However, as stated in Section 2.1, having stable modes is not enough to guarantee the stability of the system after switching. Nonetheless, the following region of attraction based rule guarantees that the system will remain stable after switching:

$$\sigma_i = \begin{cases} 1 & \text{if } \xi_i(k) \in \mathcal{L}_V(P_i) \\ 0 & \text{otherwise} \end{cases}, \tag{3.3}$$

where $\xi_i(k)$ is the mode's state, and σ_i is the switching rule, meaning that the system can switch to the ith mode if $\sigma_i = 1$. The switching rule does not force the use of any particular region of attraction estimation. Any method can be used to estimate the region of attraction, and the use of Lyapunov's theorem is just an example.

Let us discuss the stability of the system when this rule is used. Suppose a system with two modes is currently in mode 1, and the supervisor has deemed that it needs to go mode 2. The switch will only occur when the system's state is inside the second mode's controller's region of attraction. Because it is inside the region of attraction, it guarantees that it will converge, and therefore the switch is stable.

Consider the same scenario, however the system has 10 modes and the switch from mode 1 to mode 2 follows the path $1 \to 3 \to 8 \to 2$, meaning that, to go from 1 to 2, it first needs to switch to 3, then to 8 and only then to 2. The worst-case would be if the system is in the middle of the intersection of all regions of attraction. In this case, the system will instantly switch modes, going from 1 to 2 in 3 sample times, in the case of a discrete system. However, since the switch can only happen when the state is inside the next mode's controller's region of attraction, it will simply cause the previous scenario to be applied recursivelly, and the system will remain stable after it reaches the last mode.

In another scenario, the software generating the references for the supervisor (i.e. generating $r(k)$) might malfunction or have a badly defined rule that will make the reference jump between many values, making the system try to switch modes seemly randomly. In this case the system will do the switches as long as it is allowed, but the state will always be inside some controller's region of attraction. It will then either never converge nor diverge (because of the ever-changing reference) or enter a state that is only inside one mode's controller's region of attraction, at which point it will not switch modes anymore.

The illustrative scenarios show that the state will always be inside some controller's region of attraction, and therefore it is not possible for the system to diverge. It can oscilate due to varying references, but that is a case of malfunction or badly desined referencing system, not normal operation. Furthermore, when used in conjunction with Command Governors, the reference is always guaranteed to be inside the region of attraction (since the constraint region is completely contained inside it), even if the supervisor's reference is not, eliminating the possible scenario where the system diverges because the reference is outside the region of attraction, dragging the system out of it.

To illustrate, we will use the same example of Section 3.1, but this time applying the region of attraction based rule. A difference, seen in Figure 3.2, is the presence of the dashed circles, representing each controller's region of attraction. When the reference $r'(k)$ changes to the first waypoint's coordinates, the supervisor will start checking whether the system's state is inside the region of attraction of the next mode's controller and, when it enters this region, the controller and linearized model will be changed to those of the system's mode 2, but the constraints will still be those of mode 1. We call this a hybrid switch. In Figure 3.2, the dashed arrows represent a path followed while in a hybrid mode.

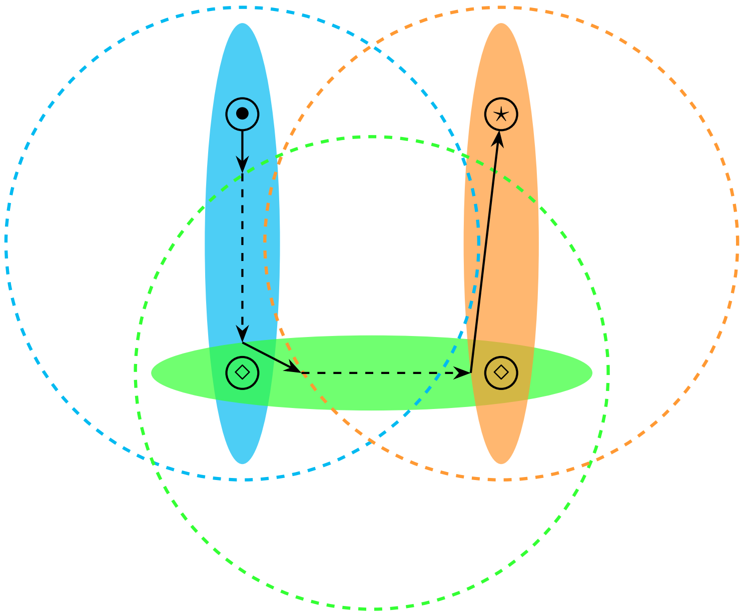


Figure 3.2 – Region of attraction illustrative example. The plane is the phase-plane of a second order system. Each colored region represents one mode's contraints, each symbol inside a circle represents a point of interest: $\bullet$ is the initial state, $\diamond$ are the waypoints and $\star$ the final reference, as set by the operator. The dashed circles represent each controller's Region of Attractoin. The arrows show the path the system will take as the supervisor changes the references to the waypoints and then to $\star$. Arrows in solid line mean the system is entirely on the active mode, dashed arrows mean the system is on a hybrid state.

The advantage of the hybrid switch is the possibility to speedup convergence. It is not required, but if the second controller is known to have a desired better performance index, it can be allowed. If it is known that it performs worse, this rule can be ignored. Here we assume that switching earlier always results in faster convergence to highlight in the figure when those changes would occur.

While in this hybrid mode, the system continues to converge to the waypoint. The supervisor checks at each instant whether the system's state is inside the current and next mode's constraint's intersection. Upon entering the intersection, a full mode switch is performed, activating mode 2's command governor and constraints. The supervisor sets the reference $r'(k)$ to the next waypoint in path, in this case the second one, and system is free to converge to it.

The same procedure will be repeated, with a hybrid switch occurring when the system's states enters the third mode's controller's region of attraction and another full switch when it enters modes 2 and 3's constraints intersection.

With this switching rule, there is no wait: the system will change mode as soon as possible, yielding faster overall convergence. The hybrid switching is an addition that may increase the performance of the system at the cost of computation resources, but is not necessary and can be safelly ignored. Algorithm 2 shows the proposed rule in both its forms, assuming the use of a Command Governor, which is always recommended for this rule as it helps to guarantee stability.

Algorithm 2 Switching rule based on region of attraction

Input: $CG_i \leftarrow$ current CG, $CG_j \leftarrow$ target CG.
let P_j be the P matrix of CG_j's Lyapunov function.
if should switch controllers earlier **then**
  while $\xi(k) \notin \mathcal{L}_V(P_j)$ **do**
    calculate $g(k)$
    execute controller
  end while
  change current controller and model to CG_j's ones
  reset integrators
end if
while $x(k) \notin \mathcal{X}_i \cap \mathcal{X}_j$ or $\hat{\xi}(k) \notin \mathcal{L}_V(P_j)$ **do**
  calculate $g(k)$
  execute controller
end while
change to CG_j
reset integrators if not already done
change $r(k)$ to next way-point
restart algorithm

CG_i is the currently active CG and CG_2 is the next CG in the path. P_j is next CG's Lyapunov function's P matrix. `should switch controllers earlier` tests whether a hybrid from CG_i to CG_j is allowed. If it is, the supervisor checks if the system's state is inside next mode's controller's region of attraction. While it is not, the current controller is executed, alogside its CG unit. Once the system enters the next mode's controller's region of attraction, the controller and internal linearized model are switched, taking the necessary measures to guarantee continuity of signals, specially control signal, such as resetting integrators. This controller is executed until the system's state enters the current and next mode's constraint's intersection, when the full mode switch occurs. The reference $r'(k)$ is set to the next waypoint's coordinates and the algorithm is restarted.

3.3 Practical Implementation Aspects

When implementing the techniques described in this thesis, some details need attention. For example, it is easy to say "minimize $f(x) \mid x \in \mathcal{X}$", however it is not so simple (and sometimes even impossible) to implement the set membership check $x \in \mathcal{X}$. Describing the set and finding a way to test membership numerically can be challenging. That is one reason why we choose convex sets: we know how to implement membership tests and use it in common optimization frameworks.

This chapter will discuss how to represent the constraints and region of attraction and provide some insight into how to implement the models of the command governor with a supervisor structure, and the problems faced when estimating the region of attraction.

3.3.1 Polytope representations

A convex polytope can approximate any convex region. A convex polytope is a compact convex set with a finite number of extreme points such that, for any two distinct points (a, b) belonging to the region, a closed segment with endpoints (a, b) is entirely contained within the region (Grünbaum, 1967). In other words, it is a convex hull of a finite number of points, called vertices. This representation is called the "V-representation" in $\mathcal{R}^2$ and is not useful to test membership. However, it is straightforward to sample points from the desired region's border to create a V-representation of a polytope approximating it and then convert it to the H-Representation, which can be used to easily test set membership. It is also useful for plotting. The Figure 3.3 illustrates a region in its V-representation, where the red dots are the vertices, and all blue dots are inside the region. The set of vertices is

$$\{(0.02,0.71),(0.31,0.02),(0.82,0.04),(0.87,0.13),(0.99,0.88),(0.60,0.98),(0.38,0.97),(0.04,0.88),(0.02,0.82)\} .$$

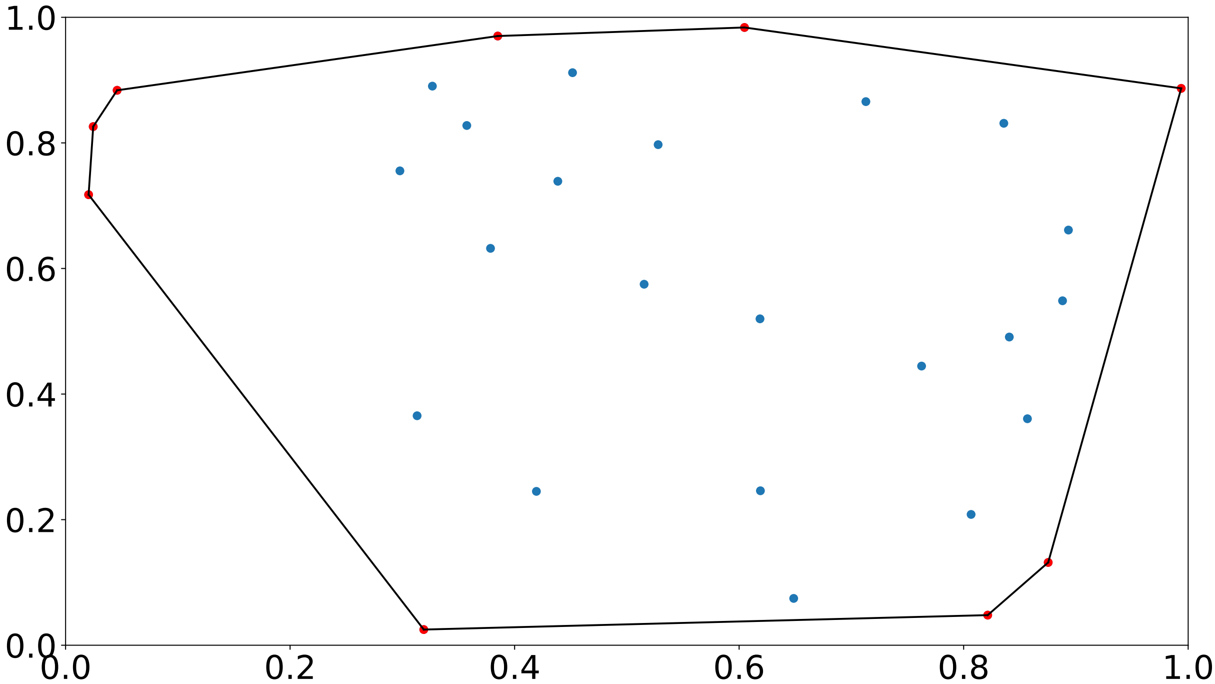


Figure 3.3 – Visualization of the V-representation of a set. The red dots are the vertices, which can be obtained by sampling the border of the set. The blue dots are considered to be inside the region, however it is not easy to test this with the V-representation.

A more useful representation of such regions is the "H-representation". In such a representation, the intersection of a finite number of half-spaces describes the region. A half-space is one of the two parts in which a hyper-plane divides an affine space. Since half-spaces are linear inequalities, the H-representation becomes the matrix inequality

$$Ax \leq b, \tag{3.4}$$

where A is a matrix of coefficients, x is a point in space and b is a vector of real numbers. The number of rows in A and b is the same as the number of half-spaces defining the region. All points satisfying the inequality are inside the region, making it easy to test set membership. Figure 3.4 shows the half-spaces that compose the H-representation of the same convex polytope presented in Figure 3.3. Equation 3.5 shows the inequation that describes the region. The selected side is not shown for each half-space but is the one in which intersections with the other selected halves make the convex region in the middle.

$$\begin{bmatrix} -0.91 & -0.39 \\ 0.04 & -0.99 \\ 0.98 & -0.15 \\ 0.84 & -0.54 \\ -0.24 & 0.96 \\ 0.24 & 0.97 \\ -0.06 & 0.99 \\ -0.99 & 0.03 \\ -0.93 & 0.34 \end{bmatrix} x \leq \begin{bmatrix} -0.30 \\ -0.01 \\ 0.84 \\ 0.66 \\ 0.84 \\ 1.10 \\ 0.94 \\ 0.00 \\ 0.26 \end{bmatrix} \tag{3.5}$$

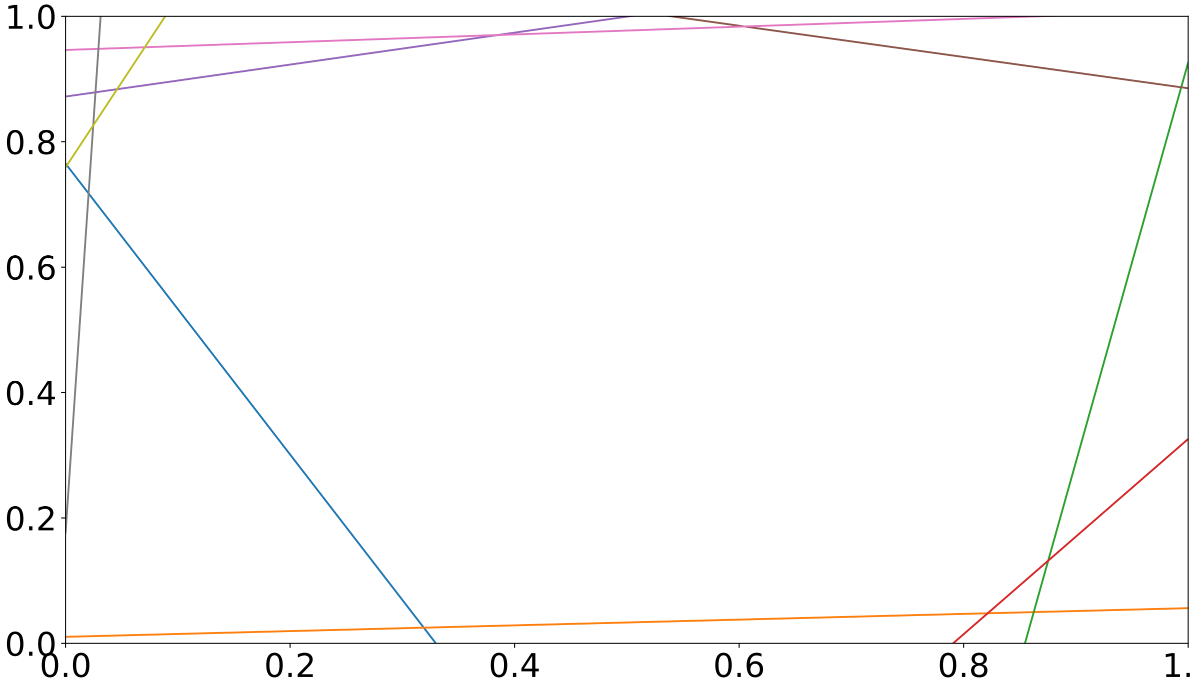


Figure 3.4 – Visualization of the H-representation of a set. The set is defined as the intersection of many half-hyperplanes. Only the lines that define the half-hyperplanes are represented, to facilitate visualization. The easiest way to get this representation is by transforming from the V-representation using one of the available algorithms. However, it is easy to test for membership using this representation.

Given a generic shape, it is easy to sample points and build a V-representation of the polytope, however we need the H-representation to easily test for set membership. Therefore, we need a way from transforming the V-representation into the H-representation.

Unfortunately, this is not straightforward, but some algorithms allow us to enumerate the facets given the vertices. Some problems of such algorithms are the time needed to find the facets of many vertices and how to find facets in higher-order spaces. However, good algorithms can be used for the conversion, especially if the computation is done offline, not presenting a time limitation to the computation (Avis et al., 1997; Graham & Frances Yao, 1983; Lee, 1983; McCallum & Avis, 1979).

Particular regions can be easily described in a way that it is easy to include in optimization problems. The ellipsoid is one such region, as it is described as $x^\top Px \leq 1$ so that any point x that satisfies the inequality is inside the region. Any region described by polynomials can be expressed in a matrix form and can quickly test membership. Such regions should use their inequation directly instead of a polytope approximation. We do that to the Region of Attraction, and it is how we implemented the check to verify if the state is inside it.

3.3.2 Internal Models

To switch modes and controllers of a system the way we propose, one must pay attention to the controller's internal states and the control signal's continuity. Also, since every *CG* unit has its controller and model, the destination *CG* must be updated to a valid state before switching. There are many ways to do so, and we will present some.

One way is to run all *CG*s in parallel. The optimization problem of the inactive units does not need to run with complete constraints. It can be relaxed only to contain the region constraint on the virtual reference; also, the observer can be ignored. It is necessary to find the virtual reference, which will be close to the constraint region's border, which is closest to the real system's state. Notice that, in this case, we are not looking for a virtual reference closest to the real reference but to the real system's state. By doing so, the internal model's and the controller's states will always be valid.

However, this approach can be resource-intensive, as it still computes the optimization problem at every sample. It is also possible to run the observer and update the internal model's states without checking for constraint violations, but this does not solve the controller's state problem. Another technique needs to be combined to find them.

Another way of guaranteeing valid states is to compute the states before changing mode. Two ways of computing the states are: by inverting the system and controller equations and by simulation. The first approach has a very low computational resource impact but may not be possible or yield approximations due to uncertainties. For systems with integrators, for example, there will be infinitely many results. Another problem is that even though the system has only one steady-state solution, it can have many transient ones.

The simulation becomes interesting, even though it is more computationally intensive, as it can calculate a state that better matches the current transitory characteristics of the real system. It can also find the model's and controller's states at once, guaranteeing that there is no mismatch. To use the simulation approach, execute the simulation right before changing modes, using the internal controller and model, and setting the reference to the real system's current state. This yields a valid steady-state model's and controller's state. However, this technique does not guarantee control signal continuity (which might even be impossible on some systems) without adjustments.

3.3.3 Region of Attraction Estimation

The Lyapunov approach presented in Section 2.3 can easily estimate the region of attraction. The presented approach makes use of a quadratic Lyapunov candidate, but it is not necessary, as it is possible to find functions of any form. The function should easily check if a point belongs to the region, as it will be done frequently. Quadratic forms, such as the simple one presented or those generated by Sum Of Squares techniques, are recommended since they can easily and computationally inexpensively check set membership.

The region of attraction needs to be larger than the basin of attraction of the CG and needs to intersect all mode's regions of attraction to and from which it can switch. Because of this, it is necessary to guarantee the intersection. However, LMIs usually try to optimize the size of the region of attraction by making it as big as possible or as small as possible. Since the exponential stability tries to minimize the size of the region of attraction, we need a way to ensure a minimum region size.

Consider the region of attraction described by

$$x^\top P x \leq 1, \tag{3.6}$$

where x is the point that should be inside the region of attraction. Note that, numerically, a number smaller than 1 may be necessary to make the problem feasible.

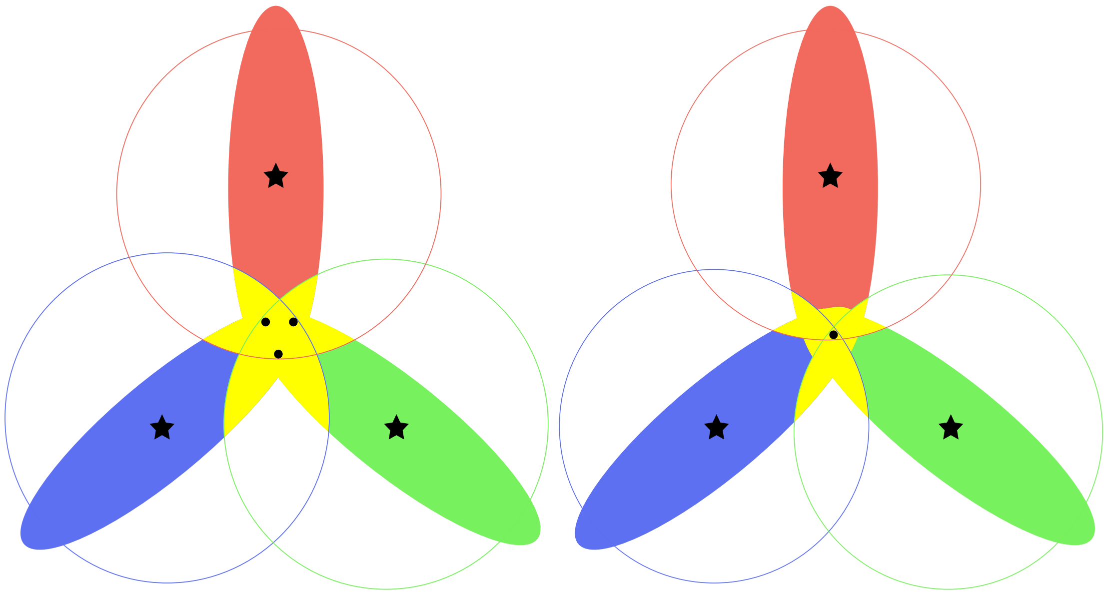
If the LMI is written in terms of P, it can be used directly to force the region to be big enough to contain the point x. If the LMI is written in terms of P^{-1}, simply applying Schur's complement yields a valid LMI:

$$\begin{bmatrix} P^{-1} & x \\ x^\top & 1 \end{bmatrix} \succ \mathbf{0}, \tag{3.7}$$

You need to add one of such LMIs to your optimization problem for each mode that is allowed to switch from or to the current mode. The point x does not need to be the same for two modes that intersect, as placing the points some distance apart creates a larger region of intersection, which gives a margin for errors.

To illustrate the choice of x, consider the switched system composed of three modes shown in Figure 3.5a. The stars represent each mode's linearization point, the filled ellipses its constraints regions, the unfilled circles the regions of attractions and the black dots the points used in the LMIs to ensure the RoA's size. Note the intersection region formed in the middle, displayed in yellow.

Now compare it with the same system, but only one point used for all systems, depicted in Figure 3.5b. See how smaller is the intersection of regions of attraction and constraints (in yellow). The proposed technique will be much more efficient in the first case, since it will result in an earlier switch.



(a) Choosing different x for each RoA (b) Choosing the same x for all RoA

Figure 3.5 – Different RoA's intesections. The intersection in 3.5a is larger, so a hybrid switch can be activated earlier.

Chapter 4

Results

This chapter presents both simulations and experimental results that show the technique working and compares it with dwell-time implementations.

For all simulations and experiments in this chapter, the controller is designed to ensure null steady-state error for piecewise constant references at each mode i, and thus the integral action is applied over the tracking error (Lopes et al., 2020)

$$e(k) = r(k) - y(k), \tag{4.1}$$

where $r(k)$ is the desired output of the system. The proportional action comes from the system's state deviation with respect to the equilibrium point. Figure 4.1 depicts the topology of the considered controller.

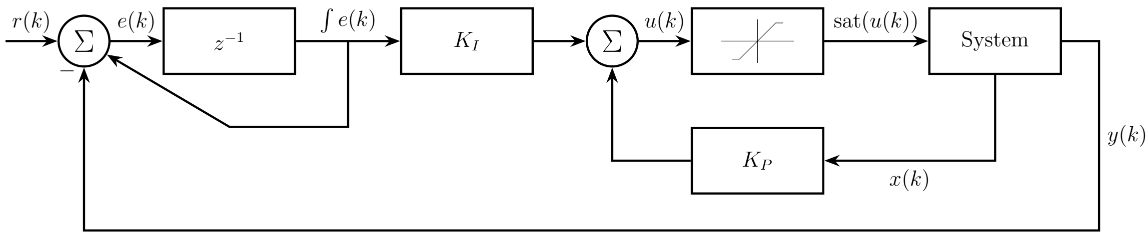


Figure 4.1 – PI-like controller. The states are fedback proportionally and the output error is integrated to obtain reference tracking.

By defining an augmented state vector

$$\xi(k) = \begin{bmatrix} x(k)^\top & v(k)^\top \end{bmatrix}^\top,$$

where $v(k) \in \mathbb{R}^{n_u}$ is the vector of added integrators, the closed-loop system shown in Figure 4.1 can be rewritten as

$$
\begin{aligned}
\xi(k+1) &= \mathcal{A}_i\xi(k) + \mathcal{B}_i u(k), \\
y_k &= \mathcal{C}_i\xi(k) + \mathcal{D}_i u(k),
\end{aligned}
\tag{4.2}
$$

where $\mathcal{A}_i = \begin{bmatrix} A_i & \mathbf{0} \\ -C_i & \mathbf{I} \end{bmatrix}$, $\mathcal{B}_i = \begin{bmatrix} B_i \\ D_i \end{bmatrix}$, $\mathcal{C}_i = \begin{bmatrix} C_i & \mathbf{0} \end{bmatrix}$, $\mathcal{D}_i = \begin{bmatrix} \mathcal{D}_i^\top & \mathbf{0} \end{bmatrix}^\top$.

The design of each controller gain $K_i \in \mathbb{R}^{n_u \times (n+p)}$ may use tandard LMI based techniques, such as, for instance, pole placement (Yu, 2013), LPV design (Briat, 2014), or robust control (Boyd et al., 1994). The exponential stability and saturation LMI's described below were used.

Note that the constrained output can also be expressed in terms of the augmented state $\xi(k)$, taking into account the state of the integrator, i.e., $c(k)$ in (2.14) can be given by

$$
c(k) = \mathcal{E}_i\xi(k) + \mathcal{F}_i u(k), \tag{4.3}
$$

with the mode-dependent matrices $\mathcal{E}_i$ and $\mathcal{F}_i$ with adequate dimensions.

There are various ways to measure stability, with the asymptotic stability being the most common. However, other definitions might yield better results. One of such stability criteria is exponential stability, which guarantees fast convergence to the origin. A system is said to be exponentially stable if its states decay is upper-bounded by an exponential (Hespanha, 2018):

$$
\|x(t)\| \leq Ce^{-\lambda(t-t_0)} \|x(0)\|. \tag{4.4}
$$

Squaring both sides of the equation and expanding the norms we get

$$
x(t)^T x(t) \leq C^2 e^{-2\lambda(t-t_0)} x(0)^T x(0). \tag{4.5}
$$

Using Taylor's series approximation to discretize the equation with $C = 1$ and letting the constant λ absorb the constant 2 yields

$$
x(k+1)^T x(k+1) \leq (1-\lambda) x(k)^T x(k). \tag{4.6}
$$

Inserting the matrix P between both vectors does not affect the innequality. Replacing $x(k+1) = (A+BK)x(k)$ results in

$$
x(k)^T[(A+BK)^T P(A+BK) - (I-\lambda I)P]x(k) \prec \mathbf{0}. \tag{4.7}
$$

Applying the Schur complement we obtain

$$\begin{bmatrix} (I-\lambda I)P & (A+BK)^T P \\ P(A+BK) & P \end{bmatrix} \succ \mathbf{0} \tag{4.8}$$

Pre- and post-multiplying by $\left[\begin{smallmatrix} P^{-1} & 0 \\ 0 & P^{-1} \end{smallmatrix}\right]$ and making $W = P^{-1}$ and $Z = KP^{-1}$, we obtain our LMI, which can be used with the following optimization procedure:

$$\begin{aligned} \min_{W,Z} \; & \mathtt{Trace}(W) \\ \text{s. t.} \; & \begin{bmatrix} W(I-\lambda I) & (AW+BZ)^T \\ AW+BZ & W \end{bmatrix} \succ \mathbf{0}. \end{aligned} \tag{4.9}$$

Considering (4.2) under input saturation, even though the input constraints are usually included in the set $\mathcal{C}$ and are handled by the optimization machinery, at this point, we explicitly consider them, by using a saturating allowance approach (Tarbouriech et al., 2011). In such a case, we assume decentralized saturation function, $\mathtt{sat}\,(u(k))$, instead of $u(k)$ as the input signal in (4.2), where $\bar{u}$ is the saturation value and

$$\mathtt{sat}\,(u) = \operatorname{sign}(U_\ell)\min\{|u_{(\ell)}|, \bar{u}_{(\ell)}\}, \quad \ell = 1, \ldots, m,$$

where $u_{(\ell)}$ means the ℓ-th entry of u. An estimate of the region of attraction can be computed by solving a suitable optimization procedure for each mode, such as (Klug et al., 2015):

$$\begin{aligned} \min_{P_i,G_i,S_i} \; & \text{Trace}(P_i) \\ \text{s. t.} \; & \begin{bmatrix} A_i^\top P_i A_i - P_i & -B_i S_i + G_i^\top \\ \star & -2S_i \end{bmatrix} < \mathbf{0} \\ & \begin{bmatrix} -P_i & (K_i - G_i)^\top \\ \star & -u^2 \end{bmatrix} < \mathbf{0} \end{aligned} \tag{4.10}$$

with $G_i \in \mathbb{R}^{n_u \times n_a}$, $S_i \in \mathbb{R}^{n_u \times n_u}$ is a positive diagonal matrix, for $\ell = 1, \ldots, m$.

4.1 Simulations

This section presents three MIMO-systems simulations using both region of attraction techniques described in Algorithm 2 (with and without early controller switching). The level-system presented models a physical system present in our laboratory, the unstable system is purely theoretical, and the Cessna 182 was taken from an article for comparison.

4.1.1 Level Control System

Consider an interactive tank system as indicated in Figure 4.2. It consists of two coupled tanks, namely $T1$ and $T2$, that are feed by two with controlled outflows u_1 and u_2, measured in $\mathrm{cm}^3\,\mathrm{s}^{-1}$. The levels of each tank, h_1 and h_2 (cm), are the control objective variables.

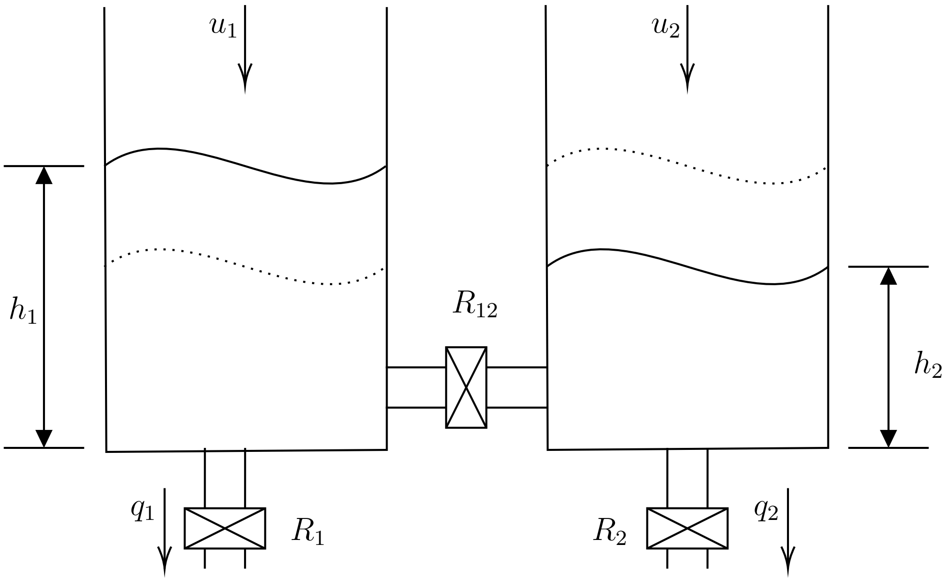


Figure 4.2 – System of Coupled Tanks. Two pumps input water into the system, and the water can flow out of the system through each tank. The water can also flow from one tank to another, making the state of one affect the state of the other. The control objective are the water levels of both tanks. The solid and dashed lines representing the water levels illustate two configurations, one with $h_1 > h_2$ and the contrary.

The output flow of the $T1$ and $T2$ are denoted by q_1 and q_2 ($\mathrm{cm}^3\,\mathrm{s}^{-1}$), respectively, and the flow between them is noted by q_{12} ($\mathrm{cm}^3\,\mathrm{s}^{-1}$). Both tanks have the same cross-section area, denoted as A (cm^2), as well as the cross-section areas of the restrictions in the outputs of the tanks, a (cm^2); g ($\mathrm{cm}\,\mathrm{s}^{-2}$) is the gravity acceleration. By using Bernoulli's equations, we have:

$$
\begin{aligned}
\dot{h}_1(t) &= \frac{u_1(t) - q_1(t) \pm q_{12}}{A}, \\
\dot{h}_2(t) &= \frac{u_2(t) - q_2(t) \mp q_{12}}{A},
\end{aligned}
\tag{4.11}
$$

where the flows are given by $q_1(t) = a\sqrt{2gh_1(t)}$, $q_2(t) = a\sqrt{2gh_2(t)}$, and $q_{12}(t) = a\sqrt{2g\,|h_2(t) - h_1(t)|}$.

This is a nonlinear and switching system, as the model changes depending on the height of the tanks' water column. At each mode, $h_1 > h_2$ or $h_1 \leq h_2$, equation (4.11) can be linearized around an equilibrium operational point (x_{eq}, u_{eq}) by using Jacobian matrices. In what follows, we use $a = 5.9\,\mathrm{cm}^2$, $A = 961\pi\,\mathrm{cm}^2$, and $g = 980.665\,\mathrm{cm\,s^{-2}}$.

Two operational conditions with $x(k) = \begin{bmatrix} h_1(k) & h_2(k) \end{bmatrix}^\top$, chosen so one tank is nearly full, are such that:

$$x^1_{eq} = \begin{bmatrix} 57.5 \\ 43.61 \end{bmatrix}, u^1_{eq} = \begin{bmatrix} 744 \\ 2960 \end{bmatrix}, x^2_{eq} = \begin{bmatrix} 43.61 \\ 57.5 \end{bmatrix}, u^2_{eq} = \begin{bmatrix} 2960 \\ 744 \end{bmatrix},$$

yielding two operational modes, each of them corresponding to a CG. After the linearization, we discretized the continuous-time model using a sample time of 5 s and Euler equations to get discrete-time model given by (2.14) with matrices:

$$\begin{aligned}
A_1 &= \begin{bmatrix} 0.92 & 0.053 \\ 0.053 & 0.91 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0.91 & 0.053 \\ 0.053 & 0.92 \end{bmatrix} \\
B_1 &= \begin{bmatrix} 0.0016 & 4.5 \times 10^{-5} \\ 4.5 \times 10^{-5} & 0.0016 \end{bmatrix}, B_2 = \begin{bmatrix} 0.0016 & 4.5 \times 10^{-5} \\ 4.5 \times 10^{-5} & 0.0016 \end{bmatrix}, \\
C_1 = C_2 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},
\end{aligned}$$

and number of modes $s = 2$. Using Lyapunov's approach (Chen, 2012; Hespanha, 2018), on the system augmented with 2 integrators (one for each state), we got the following controller:

$$\begin{aligned}
K_1 &= \begin{bmatrix} -875.384 & -9.217 & -297.447 & 7.982 \\ -8.505 & -849.853 & 8.514 & -279.434 \end{bmatrix}, \\
K_2 &= \begin{bmatrix} -849.853 & -8.505 & -279.434 & 8.514 \\ -9.217 & -875.384 & 7.982 & -297.447 \end{bmatrix},
\end{aligned}$$

for the operational modes 1 and 2.

We simulated our approaches described by Algorithm 2 to switch between the CGs 1 and 2. In both cases the system starts in $x(0) = \begin{bmatrix} 43.61 & 57.5 \end{bmatrix}^\top$. The references are $r = \begin{bmatrix} 57.5 & 43.61 \end{bmatrix}^\top$, for $0 \leq k \leq 10 \cup 50 \leq k \leq \infty$, and $r = \begin{bmatrix} 43.61 & 57.5 \end{bmatrix}^\top$, for $11 \leq k \leq 49$.

Therefore, the system must perform a closed path. Note that the references are the system's two equilibrium points, so the systems starts in mode 1, moves to mode 2 and goes back to mode 1.

The regions of attraction were estimated using the optimization procedure presented in Section 2.3. The following inequalities define each mode's regions of attraction:

$$\mathcal{L}_{V_1}(x) = x^\top \begin{bmatrix} 4.766 \times 10^{-3} & 1.431 \times 10^{-8} \\ 1.431 \times 10^{-8} & 4.766 \times 10^{-3} \end{bmatrix} x \leq 1 \tag{4.12}$$

$$\mathcal{L}_{V_2}(x) = x^\top \begin{bmatrix} 4.766 \times 10^{-3} & 1.185 \times 10^{-8} \\ 1.185 \times 10^{-8} & 4.766 \times 10^{-3} \end{bmatrix} x \leq 1 \tag{4.13}$$

The constraint regions are, in this case, also defined as ellipses, given by the following inequalities:

$$\mathcal{C}_1(x) = \frac{(x_1 - 50.55)^2}{8.681^2} + \frac{(x_2 - 43.61)^2}{1.085^2} \leq 1 \tag{4.14}$$

$$\mathcal{C}_2(x) = \frac{(x_1 - 43.61)^2}{1.085^2} + \frac{(x_2 - 50.55)^2}{8.681^2} \leq 1 \tag{4.15}$$

where $x = \left[\begin{smallmatrix} x_1 & x_2 \end{smallmatrix}\right]^\top$.

Since the system is linearized around two different operation points, there are three bases (the equilibrium points of each linearized model), and the states need to be converted between them. The first is the basis of the non-linear system, called "global". The second and third are those of the linearized systems. The constraints are written on the global basis and the regions of attraction in the respective mode's basis. Care must be taken to convert between the basis for set membership tests.

Fig. 4.4 shows the path taken by the closed-loop system over the space of the system's state. The trajectory marked with red-dashed line concerns the results achieved without early-switch, and the solid-blue line is related to ones with early-switch. The shadow regions concern the constraints of each mode. Also, Fig. 4.4 shows the borders of the estimates of the regions of attraction, RoA, for each mode, with dot-dashed lines.

It is clear that the path taken under each algorithm's variation (with and without early controller switch) are almost the same. The respective control signals are given in Fig. 4.3, showing faster convergence and smaller signal amplitude in the hybrid switch.

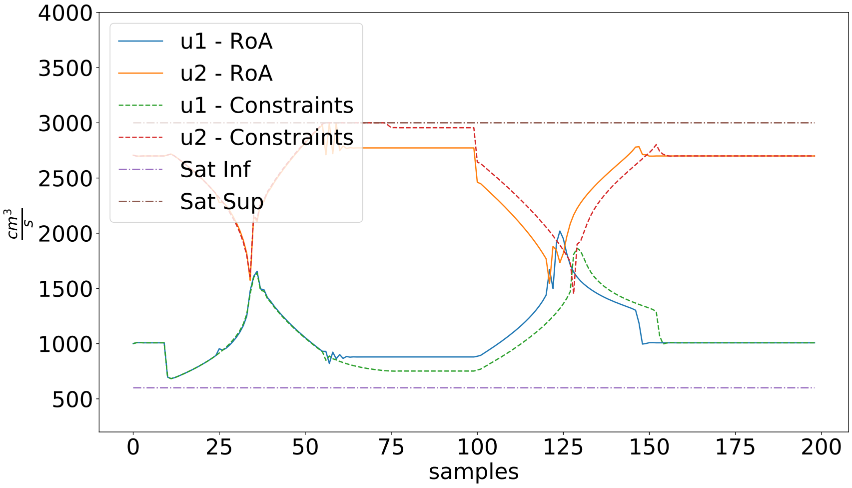


Figure 4.3 – Control signals for Example 1. Solid lines are the hybrid switch and dashed lines are the normal switch. Dashed-dotted lines are saturations.

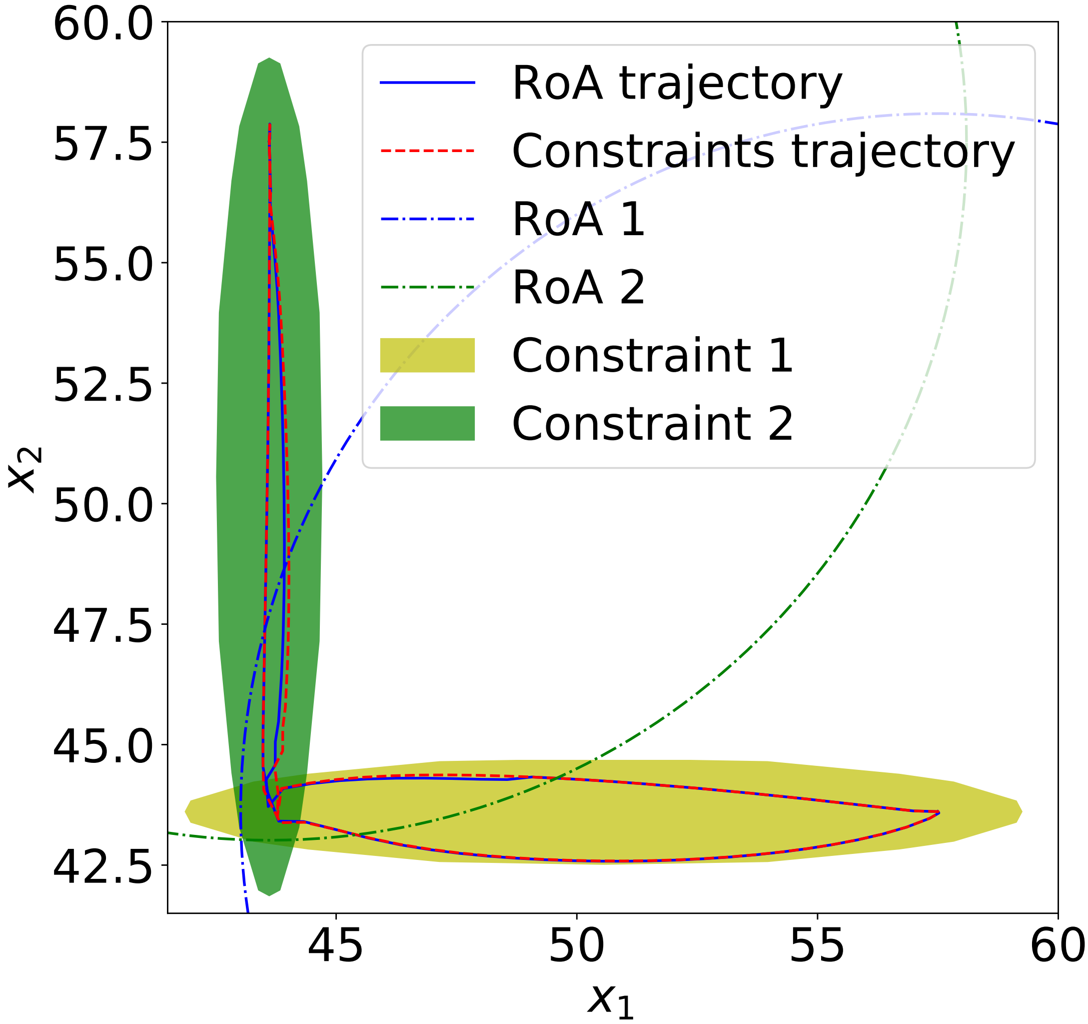


Figure 4.4 – States trajectory for Example 1. The filled ellipses are each mode's constraints. The dashed-dotted circles are each controller's Region of Attraction. The dashed blue red line is the state path of the normal switch and the solid blue line the state path of the hybrid switch.

4.1.2 Unstable System

Consider the unstable switching system with matrices

$$A_1 = \begin{bmatrix} 1 & 0.003 \\ 0 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & 0.0074 \\ 0 & 1.1 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 0.0005 & 1.2 \times 10^{-6} \\ 0 & 0.0008 \end{bmatrix}, B_2 = \begin{bmatrix} 0.0019 & 3.6 \times 10^{-5} \\ 0 & 0.011 \end{bmatrix},$$

$$C_1 = C_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and operational points

$$x_{eq}^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, u_{eq}^1 = \begin{bmatrix} -2 \\ \frac{-5}{4} \end{bmatrix}, x_{eq}^2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, u_{eq}^2 = \begin{bmatrix} \frac{-30}{19} \\ -10 \end{bmatrix}.$$

The sample-period was 0.1 s, and controller gains are given by:

$$K_1 = \begin{bmatrix} -2.669 \times 10^3 & -1.993 & -6.741 \times 10^2 & 1.010 \\ 3.582 \times 10^{-4} & -1.669 \times 10^3 & -3.103 \times 10^{-4} & -4.210 \times 10^2 \end{bmatrix},$$

$$K_2 = \begin{bmatrix} -7.034 \times 10^2 & -1.268 & -1.769 \times 10^2 & 6.097 \times 10^{-1} \\ -3.903 \times 10^{-6} & -1.370 \times 10^2 & -1.292 \times 10^{-5} & -3.202 \times 10^1 \end{bmatrix}.$$

With the same procedures and considerations of the previous example, including the used color codes, we simulated the unstable system. Figure 4.5 shows the same system trajectory for both methods.

The second method shows better performance under control signal restrictions. Figure 4.6 reveals a difference in the control signals, where the first method, displayed in red dashed-line and green dashed-line, have higher control signal outputs than the second method, shown by blue solid-line and orange solid-line. Both methods results in the same settling time, but with much lower control effort with the strategy of early switching.

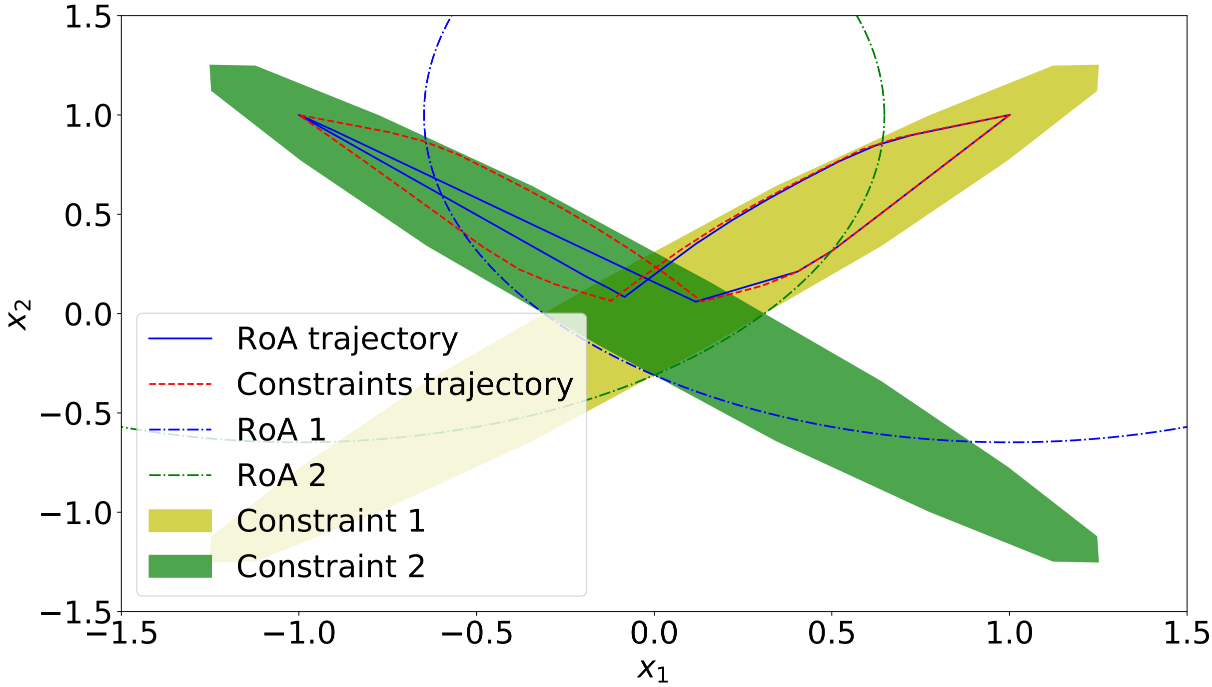


Figure 4.5 – States trajectory for Example 2. The filled ellipses are each mode's constraints. The dashed-dotted circles are each controller's Region of Attraction. The dashed blue red line is the state path of the normal switch and the solid blue line the state path of the hybrid switch.

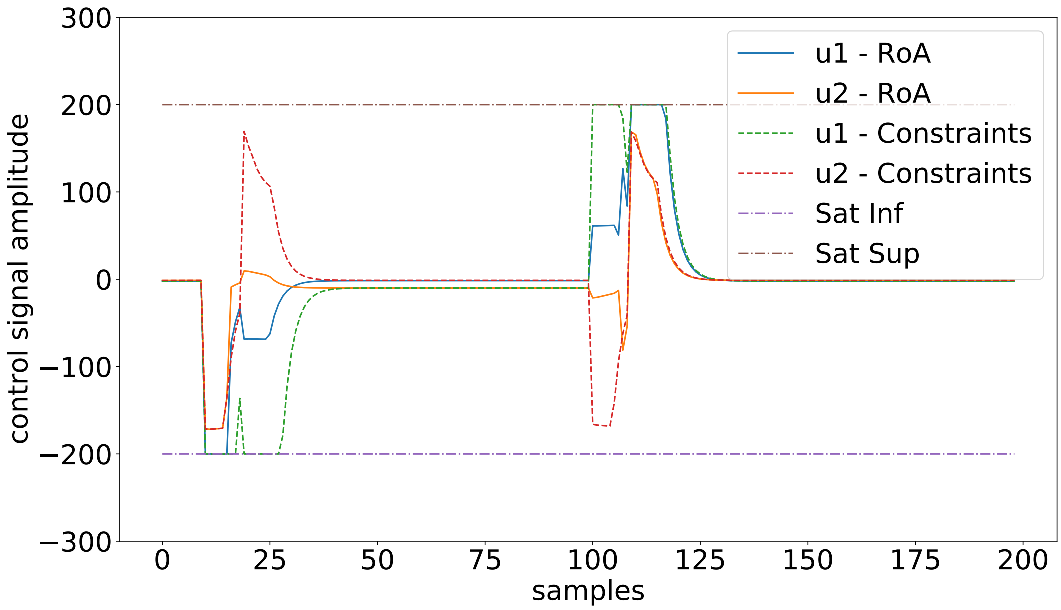


Figure 4.6 – Control signals for Example 2. Solid lines are the hybrid switch and dashed lines are the normal switch. Dashed-dotted lines are saturations.

4.1.3 Cessna 182

Franzè et al. (2017) present a Cessna 182 aircraft model, which they used to simulate a dwell-time. They describe the model, the operation points they chose, and the constraints applied to it. We used the same system changing only the switch technique to compare the performance. The proposed technique took 6 s to converge, whereas the dwell-time technique took over 20 s.

Figure 4.7 shows both actuators' control signals, which are kept mostly invariant, except at the mode transitions, where the integrator reset created a discontinuity. Figure 4.8 shows each state. In both figures, the solid black lines at the plot's top and bottom are the signal's constraints. The last state (x_4) is the output, and its plot also shows the reference and command governor's output.

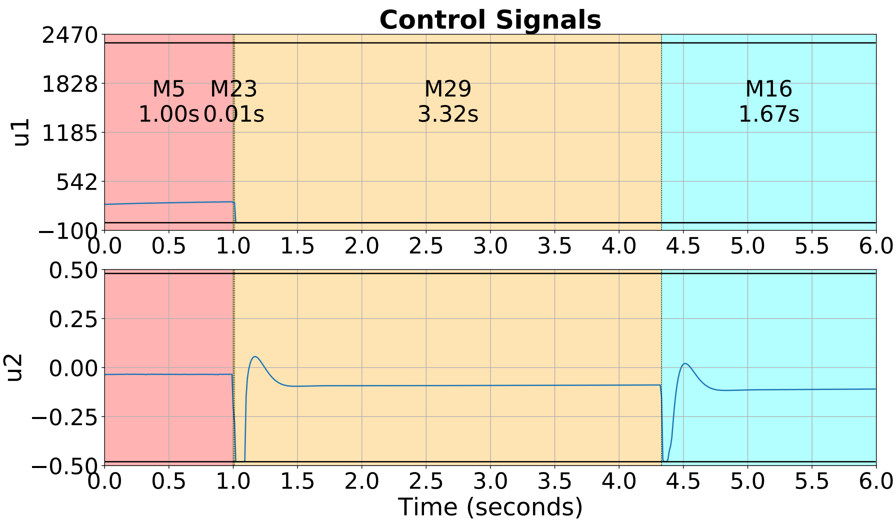


Figure 4.7 – Cessna 192 control signals.

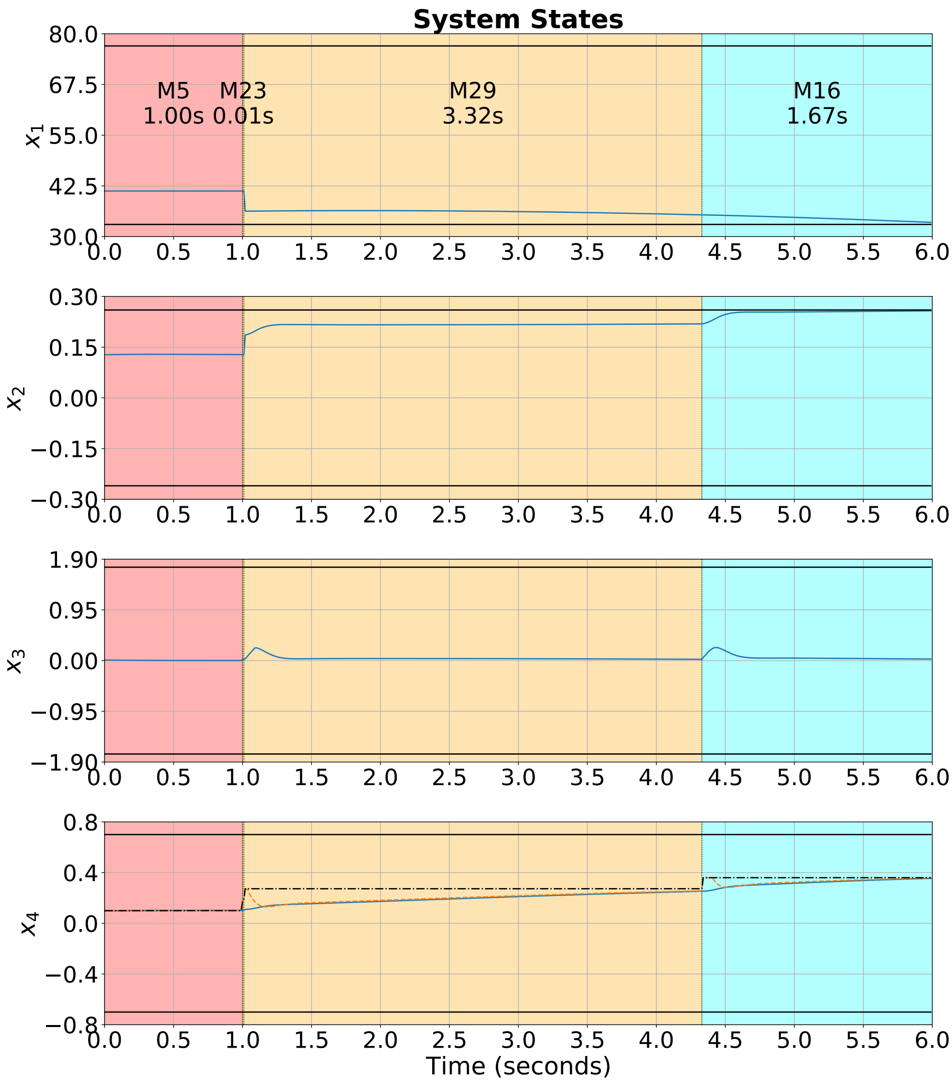


Figure 4.8 – Cessna 182 states. The colored backgrounds show the active mode. In the last plot, the dashed black line is the reference set by the supervisor, the dashed orange line is the reference generated by the Command Governor, the solid blue line is the state and the solid black lines are the saturation values.

4.2 Experimental Results

Consider an interactive tank system as indicated in Figure 4.10. It describes the physical system present at the System Analysis Laboratory of CEFET-MG campus V, shown in Figure 4.9. It consists of two coupled tanks, $T1$ and $T2$, that are fed by a pump with controlled flow $u(t)$, measured in $\mathrm{cm}^3\,\mathrm{s}^{-1}$. The levels of each tank, h_1 and h_2 (cm), are the control objective variables, and are measured directly (e Sousa et al., 2018; Franco et al., 2016; Lopes et al., 2020).

Both tanks have the same cross-section area, denoted as A (cm^2), however, there is a

Figure 4.9 – System of coupled tanks with four 200L tanks. There are two pumps that can be configured through the pipes to fill any of the tanks. Tank T3 has a rigid body with a non-linear shape inside it. The water flow between tanks is also configurable.

solid inside $T1$ that makes its area non-linear and adds uncertainties to its model. The cross-section area of $T1$ becomes

$$A_1(h_1(t)) = \frac{3r}{5}\left(2.7r - \frac{\cos(2.5\pi((h_1(t)-8)\times 10^{-2}-\mu))}{\sigma\sqrt{2\pi}} e^{-\frac{((h_1(t)-8)\times 10^{-2}-\mu^2)^2}{2\sigma^2}}\right), \quad (4.16)$$

where $\mu = 0.4$, $\sigma = 0.55$ and $r = 0.31$. The cross-section area of $T2$ is $0.31\,\mathrm{m}^2$.

By using Bernoulli's equations, the system's dynamics can be described by:

$$\begin{aligned} \dot{h}_1(t) &= \frac{R_{12}(h_1(t),h_2(t)) \times K_b \times u(t) - h_1(t) + h_2(t)}{A_1(h_1(t)) \times R_{12}(h_1(t),h_2(t))} \\ \dot{h}_2(t) &= \frac{h_1(t)-h_2(t)}{R_{12}(h_1(t),h_2(t)) \times A_2} - \frac{q_o(h_2(t))}{A_2}, \end{aligned} \quad (4.17)$$

where $R_{12}(h_1(t),h_2(t)) = (0.412(h_1(t-h_2(t))+11.488)\times 10^{-3}$ and $q_o(h_2(t)) = 11.941h_2(t)+787.586$. A frequency inverter controls the pump through the percentage of maximum flow, and this value can be converted to flow by $q_i = 13.201u_t + 220.085$. Applying this before

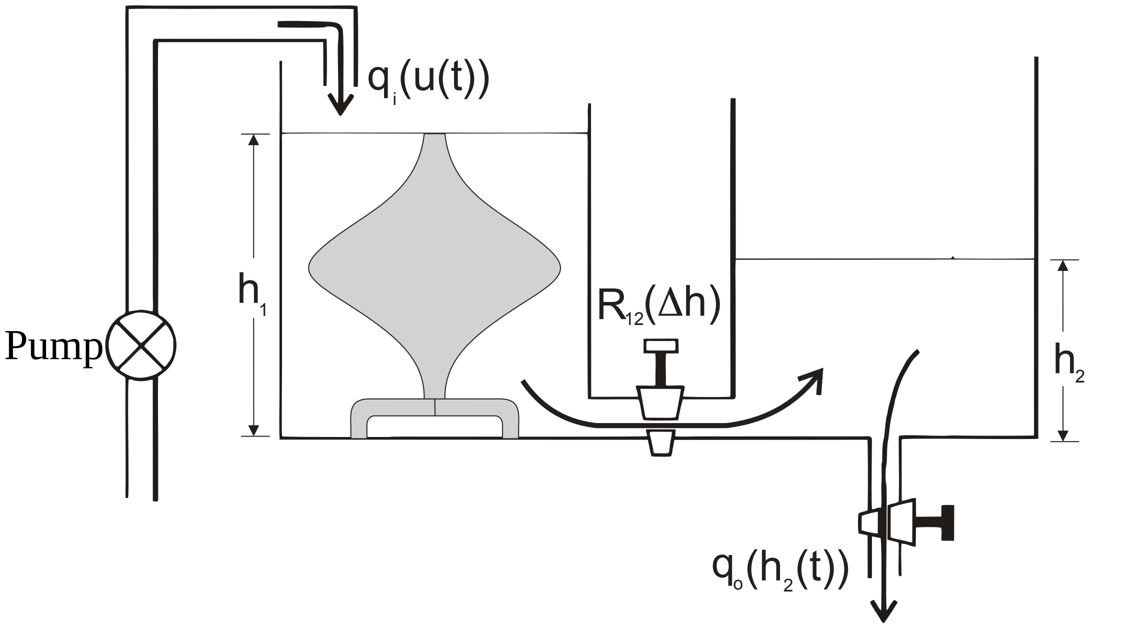


Figure 4.10 – Diagram of the third and fourth tanks, showing the non-linear body. The water flows from the pump into the tank three, then move to tank four and goes back to the reservoir.

inputing in Equation (4.17) results in $u(t)$ becoming this percentual value instead of the flow directly.

We chose four operation points to cover a significant portion of the available h_1 range (0 to 70 cm), which is the output of the system. The points and their respective linearized and discretized (with a sampling time of 5 s) systems are

$$\left[\begin{array}{c|c} x_{eq}^\top & u_{eq} \\ \hline A & B \end{array}\right]_1 = \left[\begin{array}{cc|c} 19.5 & 5 & 15 \\ \hline 0.91 & 0.14 & 0.028 \\ 0.085 & 0.9 & 0.0013 \end{array}\right], \quad \left[\begin{array}{c|c} x_{eq}^\top & u_{eq} \\ \hline A & B \end{array}\right]_2 = \left[\begin{array}{cc|c} 27.5 & 11.6 & 20 \\ \hline 0.94 & 0.19 & 0.04 \\ 0.084 & 0.9 & 0.0018 \end{array}\right],$$

$$\left[\begin{array}{c|c} x_{eq}^\top & u_{eq} \\ \hline A & B \end{array}\right]_3 = \left[\begin{array}{cc|c} 37.3 & 17.8 & 25 \\ \hline 1.1 & 0.47 & 0.1 \\ 0.086 & 0.92 & 0.0042 \end{array}\right], \quad \left[\begin{array}{c|c} x_{eq}^\top & u_{eq} \\ \hline A & B \end{array}\right]_4 = \left[\begin{array}{cc|c} 47.4 & 24.9 & 30 \\ \hline 0.49 & 0.96 & 0.22 \\ 0.053 & 0.95 & 0.0096 \end{array}\right].$$

To develop the controllers we applied the LMI described in the Section 2.3 - Region of Attraction to the integral-augmented system. The augmented state is the integral of the output, h_1. The points forced to be inside the region of attraction are the previous' and next's operation point. It creates an intersection that allows the early switching of controllers, which speeds up convergence. The following controllers were obtained:

$$K_1 = \begin{bmatrix} -12.884 & -97.540 & -13.975 \end{bmatrix}, \tag{4.18}$$

$$K_2 = \begin{bmatrix} -10.054 & -73.777 & -10.523 \end{bmatrix}, \tag{4.19}$$

$$K_3 = \begin{bmatrix} -5.840 & -31.622 & -4.148 \end{bmatrix}, \tag{4.20}$$

$$K_4 = \begin{bmatrix} -1.832 & -21.527 & -4.177 \end{bmatrix}. \tag{4.21}$$

To calculate each of the described modes' dwell-time, we applied the technique described by Franzè et al. (2017). We then ran both approaches — the dwell-time and the proposed one — on the physical system. Figures 4.11 and 4.12 show the result of the experiments, where the colored backgrounds represent the different modes, M1 to M4, the two states (x_1 and x_2) are plotted alongside their respective real ($r_{1,2}$) and virtual ($g_{1,2}$) references.

The proposed technique has a faster overall convergence, entering the final mode after 215 seconds while the dwell-time approach takes 340 seconds. This can easily be seen by comparing the width of the regions with different background colors, as the red, orange and yellow regions are much shorter on the proposed technique. The proposed approach also shows no overshoot, since it does not give the system enough time to converge to the waypoints and keeps it always moving towards the reference, as shown by the states' trajectories, which are always moving torwards the reference in the proposed technique, but starts to converge to the mode's operation point on the dwell-time technique. Because of that, it also has a smaller control effort, producing smaller control signal variation, seen in the last plot, where the control signal constantly saturates in the dwell-time approach, but does not saturate and has a smaller variation in the proposed technique.

The last mode should also have an overshoot on the proposed technique, since it is part of the closed-loop dynamics. However, the command governor's optimization problem generated virtual references that slowly guided the system to the real reference, making it take longer than necessary to converge. The behaviour was present on every test repetition, just like the dwell-time approach's overshoots.

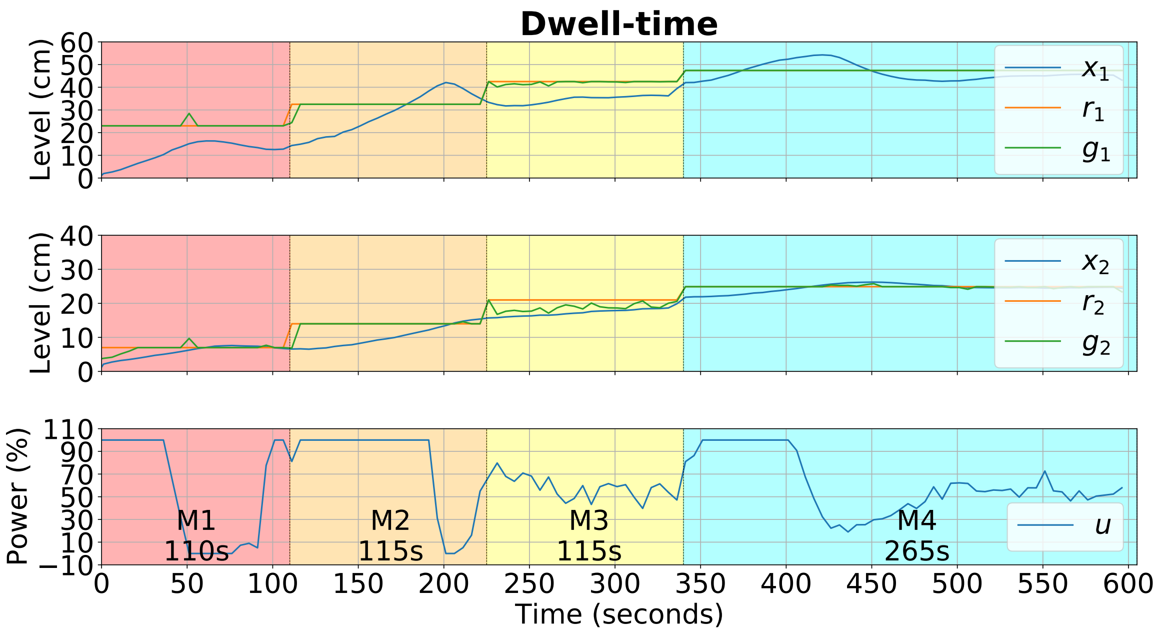


Figure 4.11 – This Figure shows the states and control signal of the system running with the dwell-time rule. The first two plots show the system states as well as the reference set by the supervisor and command governors and the third plot shows the control signal. All plots have colored backgrounds displaying the active mode, and the last plot also shows the time that each mode remained active. The states show overflows on first, second and fourth modes, which is expected from the system, and the control signal saturates four times and presents a large variation, going from 100% to 0%.

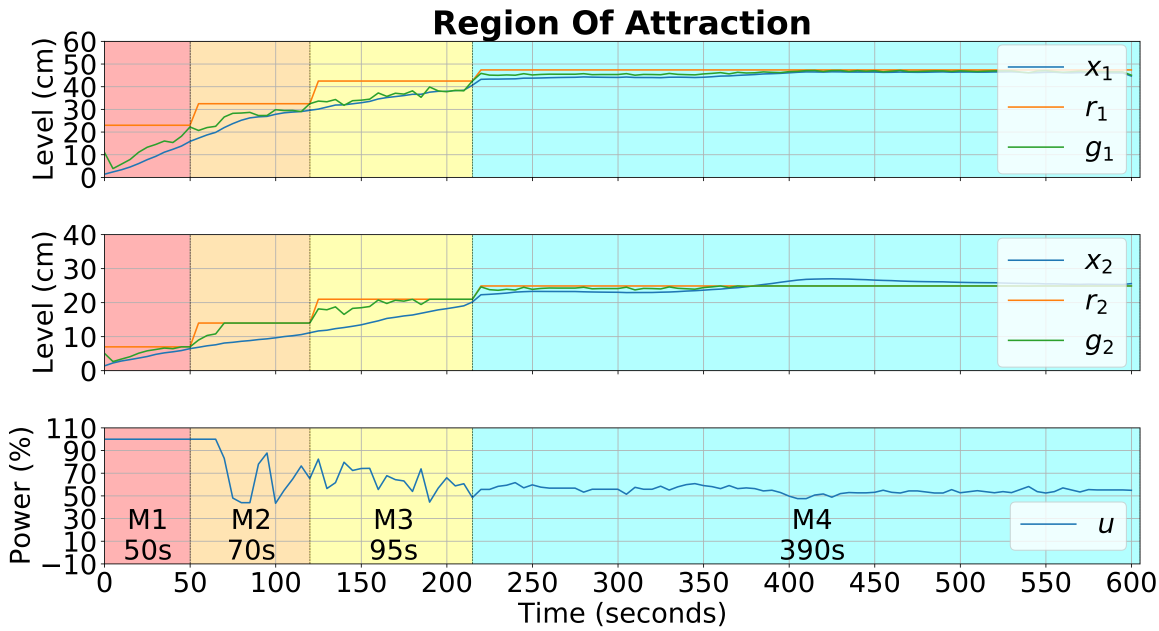


Figure 4.12 – This Figure show the states and control signals of the Region of Attraction-based switching rule. It has the same structure as Figure 4.11. The overall convergence is visibly faster than the dwell-time alternative. Also, the states do not present overshoots, since the system is not given enough time to converge to the waypoints. The control signal has a smaler variation in amplitude only saturates at the begining.

Chapter 5

Final Considerations

Switched systems usually lose stability under arbitrary switching, requiring the development of switching rules that keep the system stable after the switch. A class of switching rules denominated slow-switching guarantees stability by not switching between modes too fast. Most of such techniques try to determine the minimum amount of time the system needs to remain on a mode before switching to remain stable.

Command Governors enforces constraints on dynamic systems. When used with switched systems, it enforces constraints on each mode, aggravating the stability problem. Now the system is not allowed to switch from any state, only from a subset inside the constraints' intersection. The design of slow-switching rules must take this into account to avoid violating constraints.

In this work, we studied switched systems' constraints enforcement through Command Governors. The switched systems' stability problem, usually solved using dwell time, was reviewed under a new light, resulting in the proposed Region of Attraction-based switching rule, something not found in literature. The rule, shown in Algorithm 2, guarantees the system stability after mode switches by only allowing the switch if the system's states remain inside the controller's Region of Attraction and are inside constraints' intersection.

On the same algorithm, we can switch the controller earlier while keeping the constraints, which we call a hybrid change. It allows for faster convergence when the next controller performs better than the current one. However, if that is undesirable, the switch can happen at once on the constraints' intersection, still having all stability guarantees from the Region of Attraction.

The presented technique allows faster overall convergence of constrained switched systems while ensuring stability after switches. It is based on well-consolidated concepts and builds on them to provide stability to constrained switched systems. Although it was not designed to be used without a Command Governor, the rule can be further extended to guarantee stability in that case as well.

The major problem with the technique is that it is hard to apply it to existing controllers since their region of attraction might not satisfy the necessary criteria of size and intersections. Because of this, the main advantage of the Command Governor is lost and new controllers will most probably need to be synthetized.

On Chapter 4 we apply the technique on different simulated systems as well as a real one. The simulations illustrate how the proposed rule works and compares it with results from an article. The experimental result shows the convergence time gain when compared to the dwell-time technique as well as the better control signal trajectory. The simplicity of the algorithm does not put a burden in the hardware implementation and the design requires the estimation of the Region of Attraction where other slow-switch techniques require the estimation of the dwell-time.

This work resulted in the publication of an article at Congresso Brasileiro de Automática — CBA in November 2020 (e Sousa et al., 2020). An extended version, including the experimental results obtained, is being prepared to be submitted to a journal.

5.1 Perspectives

This work only deals with LTI systems. Future research can extend the technique to models with uncertainties, varying the region of attraction or modifying the switching rule to accommodate them. In this case, the designer must pay attention to the possibility of the system becoming unstable if the switch happens when the system is not inside the region of attraction due to model mismatch, possibly leading to instability.

On those lines, switched systems composed of LPV subsystems also need special attention. In this case, the Region of Attraction can also become a function of the states, and special attention might be necessary to ensure there are intersections.

Last but not least, the extension of the proposed rule to non-command governor schemes also needs consideration. The Command Governor plays a role in not letting the system destabilize, so it becomes necessary in the proposed schema.

Bibliography

Ahuja, R. K., Mehlhorn, K., Orlin, J., & Tarjan, R. E. (1990). Faster algorithms for the shortest path problem. *Journal of the ACM (JACM)*, *37*(2), 213–223. https://doi.org/10.1145/77600.77615

Arabi, E., Yucelen, T., & Balakrishnan, S. (2020). A command governor approach to set-theoretic model reference adaptive control for enforcing partially adjustable performance guarantees. *International Journal of Dynamics and Control*, *8*(2), 675–689. https://doi.org/10.1007/s40435-019-00563-4

Avis, D., Bremner, D., & Seidel, R. (1997). How good are convex hull algorithms? [11th ACM Symposium on Computational Geometry]. *Computational Geometry*, *7*(5), 265–301. https://doi.org/https://doi.org/10.1016/S0925-7721(96)00023-5

Bemporad, A., Casavola, A., & Mosca, E. (1997). Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transactions on Automatic Control*, *42*(3), 340–349. https://doi.org/10.1109/9.557577

Bendixson, I. (1901). Sur les courbes définies par des équations différentielles. *Acta Math.*, *24*, 1–88. https://doi.org/10.1007/BF02403068

Blanchini, F., & Miani, S. (2008). *Set-Theoretic Methods in Control* [OCLC: 896260921]. Birkhäuser Boston.

Bochnak, J., Coste, M., & Roy, M.-F. (1998). *Real algebraic geometry*. Springer. https://doi.org/10.1007/978-3-662-03718-8

Boyd, S., Ghaoui, L. E., Feron, E., & Balakrishnan, V. (1994). *Linear matrix inequalities in systems and control theory*. SIAM Studies in Applied Mathematics.

Briat, C. (2014). *Linear parameter-varying and time-delay systems* (Vol. 3). Springer.

Casavola, A., Mosca, E., & Angeli, D. (2000). Robust command governors for constrained linear systems. *IEEE Transactions on Automatic Control*, *45*(11), 2071–2077. https://doi.org/10.1109/9.887628

Chen, C.-T. (2012, November). *Linear system theory and design (the oxford series in electrical and computer engineering)*. Oxford University Press.

Chesi, G., Colaneri, P., Geromel, J. C., Middleton, R., & Shorten, R. (2010). Computing upper-bounds of the minimum dwell time of linear switched systems via homogeneous polynomial lyapunov functions. *Proceedings of the 2010 American Control Conference, ACC 2010*. https://doi.org/10.1109/acc.2010.5530590

Colaneri, P. (2009). Dwell time analysis of deterministic and stochastic switched systems. *European Journal of Control*, *15*(3-4), 228–248. https://doi.org/10.3166/EJC.15.228-248

Deaecto, G. S., Geromel, J. C., & Daafouz, J. (2014). Robust h2 switched filter design for discrete-time polytopic linear parameter-varying systems. *Signal Processing*, *97*, 91–99. https://doi.org/10.1016/j.sigpro.2013.10.012

Dogan, K. M., Yucelen, T., Haddad, W. M., & Muse, J. A. (2020). Improving transient performance of discrete-time model reference adaptive control architectures. *International Journal of Adaptive Control and Signal Processing*, *34*(7), 901–918. https://doi.org/10.1002/acs.3114

Donchev, T., & Farkhi, E. (1998). Stability and euler approximation of one-sided lipschitz differential inclusions. *SIAM Journal on Control and Optimization*, *36*(2), 780–796. https://doi.org/10.1137/s0363012995293694

e Sousa, Á. C., Leite, V. J. S., & Rubio Scola, I. (2018). Affordable control platform with mpc application. *Studies in Informatics and Control*, *27*(3), 265–274. https://doi.org/10.24846/v27i3y201802

e Sousa, Á. C., Silva, L. F. P., Lucia, W., & Leite, V. J. S. (2020). Command governor strategy based on region of attraction control switching. *XXIII Congresso Brasileiro de Automática*.

Franco, A. E. O., Oliveira, L. S., & Leite, V. J. S. (2016). Síntese de ganhos para compensação robusta de sistemas linearizados por realimentação. *XXI Congresso Brasileiro de Automática*.

Franzè, G., Lucia, W., & Tedesco, F. (2017). Command governor for constrained switched systems with scheduled model transition dwell times. *International Journal of Robust and Nonlinear Control*, *27*(18), 4949–4967. https://doi.org/https://doi.org/10.1002/rnc.3841

Garone, E., Di Cairano, S., & Kolmanovsky, I. (2017). Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica*, *75*, 306–328. https://doi.org/10.1016/j.automatica.2016.08.013

Geromel, J. C., & Colaneri, P. (2005). Stabilization of continuous-time switched systems. *IFAC Proceedings Volumes*, *38*(1), 472–477. https://doi.org/10.3182/20050703-6-cz-1902.00649

Geromel, J. C., & Deaecto, G. S. (2014). Stability analysis of lur'e-type switched systems. *IEEE Transactions on Automatic Control*, *59*(11), 3046–3050.

Gilbert, E. G., & Kolmanovsky, I. (1999). Fast reference governors for systems with state and control constraints and disturbance inputs. *International Journal of Robust and Nonlinear Control*, *9*(15), 1117–1141. https://doi.org/https://doi.org/10.1002/(SICI)1099-1239(19991230)9:15<1117::AID-RNC447>3.0.CO;2-I

Graham, R. L., & Frances Yao, F. (1983). Finding the convex hull of a simple polygon. *Journal of Algorithms*, *4*(4), 324–331. https://doi.org/https://doi.org/10.1016/0196-6774(83)90013-5

Gruenwald, B. C., Yucelen, T., Dogan, K. M., & Muse, J. A. (2020). Expanded reference models for adaptive control of uncertain systems with actuator dynamics. *Journal of Guidance, Control, and Dynamics*, *43*(3), 475–489. https://doi.org/10.2514/1.G004326

Grünbaum, B. (1967). *Convex polytopes*. Interscience.

Hespanha, J. P. (2018, February). *Linear systems theory* [ISBN13: 9780691179575]. Princeton Press.

Hespanha, J. P., & Morse, A. S. (1999). Stability of switched systems with average dwell-time. *Proceedings of the IEEE Conference on Decision and Control*, *3*(December), 2655–2660. https://doi.org/10.1109/cdc.1999.831330

Kapasouris, P., Athans, M., & Stein, G. (1988). Design of feedback control systems for stable plants with saturating actuators. *Proceedings of the 27th IEEE Conference* *on Decision and Control*, 469–479 vol.1. https://doi.org/10.1109/CDC.1988.194356

Keel, L. H., & Bhattacharyya, S. P. (1997). Robust, fragile, or optimal? *IEEE Transactions on Automatic Control*, *42*(8), 1098–1105.

Klug, M., Castelan, E. B., Leite, V. J. S., & Silva, L. F. P. (2015). Fuzzy dynamic output feedback control through nonlinear takagi–sugeno models. *Fuzzy Sets and Systems*, *263*, 92–111. https://doi.org/10.1016/j.fss.2014.05.019

Lee, D. T. (1983). On finding the convex hull of a simple polygon. *International Journal of Computer & Information Sciences*, *12*(2), 87–98. https://doi.org/10.1007/BF00993195

Liberzon, D., & Morse, A. S. (1999). Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, *19*(5), 59–70.

Liberzon, D. (2003). *Switching in systems and control*. Birkhauser.

Lin, H., & Antsaklis, P. J. (2009). Stability and stabilizability of switched linear systems: A survey of recent results. *IEEE Transactions on Automatic Control*, *54*(2), 308–322. https://doi.org/10.1109/TAC.2008.2012009

Lopes, A. N. D., Leite, V. J. S., Silva, L. F. P., & Guelton, K. (2020). Anti-windup ts fuzzy pi-like control for discrete-time nonlinear systems with saturated actuators. *International Journal of Fuzzy Systems*, 46–61. https://doi.org/10.1007/s40815-019-00781-0

Lucia, W., & Franzè, G. (2017). Stabilization and reference tracking for constrained switching systems: A predictive control approach. *International Journal of Adaptive Control and Signal Processing*. https://doi.org/10.1002/acs.2804

Makavita, C. D., Jayasinghe, S. G., Nguyen, H. D., & Ranmuthugala, D. (2019). Experimental study of a command governor adaptive depth controller for an unmanned underwater vehicle. *Applied Ocean Research*, *86*(February), 61–72. https://doi.org/10.1016/j.apor.2019.02.016

McCallum, D., & Avis, D. (1979). A linear algorithm for finding the convex hull of a simple polygon. *Information Processing Letters*, *9*(5), 201–206. https://doi.org/https://doi.org/10.1016/0020-0190(79)90069-3

Milnor, J. (1985). On the concept of attractor. *The theory of chaotic attractors* (pp. 243–264). Springer New York. https://doi.org/10.1007/978-0-387-21830-4_15

Ong, C. J., Djamari, D. W., & Hou, B. (2020). A governor approach for consensus of heterogeneous systems with constraints under a switching network. *Automatica*, *122*, 109239. https://doi.org/10.1016/j.automatica.2020.109239

Peng, Z., Wang, J., & Wang, J. (2019). Constrained control of autonomous underwater vehicles based on command optimization and disturbance estimation. *IEEE Transactions on Industrial Electronics*, *66*(5), 3627–3635. https://doi.org/10.1109/TIE.2018.2856180

Pettie, S. (2004). A new approach to all-pairs shortest paths on real-weighted graphs. *Theoretical Computer Science*, *312*(1), 47–74. https://doi.org/10.1016/s0304-3975(03)00402-x

Ristevski, S., Dogan, K. M., Yucelen, T., & Muse, J. A. (2019). Transient performance improvement in reduced-order model reference adaptive control systems. *IFAC-PapersOnLine*, *52*(29), 49–54. https://doi.org/10.1016/j.ifacol.2019.12.620

Schwerdtner, P., Bortoff, S. A., Danielson, C., & DI Cairano, S. (2019). Projection-based anti-windup for multivariable control with heat pump application. *2019 18th European Control Conference, ECC 2019*, 1281–1287. https://doi.org/10.23919/ECC.2019.8795876

Seeber, R., Gölles, M., Dourdoumas, N., & Horn, M. (2019). Reference shaping for model-based control of biomass grate boilers. *Control Engineering Practice*, *82*(October 2018), 173–184. https://doi.org/10.1016/j.conengprac.2018.10.006

Shen, S., Song, A., Li, H., & Li, T. (2019). Constrained control for cloud robotic under time delay based on command governor with interval estimation. *IEEE Access*, *7*, 70999–71006. https://doi.org/10.1109/ACCESS.2019.2920017

Tarbouriech, S., Garcia, G., Gomes da Silva Jr., J. M., & Queinnec, I. (2011). *Stability and stabilization of linear systems with saturating actuators*. Springer Science.

Tedesco, F., & Casavola, A. (2020). Turn-based command governor strategies for interconnected dynamical systems with time-varying couplings. *Proceedings of the American Control Conference*, *2020-July*(1), 4564–4569. https://doi.org/10.23919/ACC45564.2020.9147325

Vahidi, A., Kolmanovsky, I., & Stefanopoulou, A. (2007). Constraint handling in a fuel cell system: A fast reference governor approach. *IEEE Transactions on Control Systems Technology*. https://doi.org/10.1109/TCST.2006.883242

Wang, L. (2009). *Model predictive control system design and implementation using matlab* (1st ed.). Springer-Verlag London. https://doi.org/10.1007/978-1-84882-331-0

Wilcher, K., Jaramillo, J., Yucelen, T., & Pakmehr, M. (2020). On performance improvement of gain-scheduled model reference adaptive control laws. *AIAA Scitech 2020 Forum*, *1 PartF*(January), 1–11. https://doi.org/10.2514/6.2020-1118

Yu, G., Hai-Hua; Duan. (2013). *Lmis in control systems: Analysis, design and applications*. CRC Press.

Zhang, X. (2016). *Fast mpc solvers for systems with hard real-time constraints* (Doctoral dissertation). TUDelft.