SMT-Based Stability Verification of an Industrial Switched PI Control Systems

Stylianos Basagiannis
Collins
stylianos.basagiannis@collins.com

Ludovico Battista
FBK
lbattista@fbk.eu

Anna Becchi
FBK
abecchi@fbk.eu

Alessandro Cimatti
FBK
cimatti@fbk.eu

Georgios Giantamidis
Collins
georgios.giantamidis@collins.com

Sergio Mover
Ecole Polytechnique
sergio.mover@lix.polytechnique.fr

Alberto Tacchella
FBK
atacchella@fbk.eu

Stefano Tonetta
FBK
tonettas@fbk.eu

Vassilios Tsachouridis
Collins
vassilios.tsachouridis@collins.com

***Abstract*—The control of complex systems is typically designed describing the physical system with differential equations. The standard approach to their verification employs numerical analysis, which is suitable to prove stability properties, but is susceptible to numerical errors. On the other side, symbolic techniques give precise analysis results but typically do not scale to industrial size problems.**

In this paper, we consider the control design of an aircraft engine. The engine model is represented by a linear state space model of 18 internal state variables, 4 outputs, and 3 inputs. The control switches between two PI controllers, one for thrust control and another for low-pressure compressor spool speed control, based on the engine state and pilot commands. After reformulating the PI controllers in terms of differential equations, we obtain a hybrid system with 21 state variables and two modes, for which we want to prove with symbolic techniques the robustness of the stable states to perturbation. We achieved the verification with standard methods to synthesize quadratic Lyapunov functions and SMT techniques to synthesize neighborhoods of the stable states for which we have symbolic proof of stability.

I. Introduction

The benefits of revolutionary new aircraft systems and architectures with high potential for energy efficiency gains are complex and difficult to assess with confidence, particularly in relation to their impact on large parallel systems and interconnected components. As air transport is one of the largest consumers of total primary energy, it is pivotal for safe and green-aware transportation initiatives to be transitioned to the industry as fast as possible. Thus, it is crucial for analysis and verification tools of advanced engine concepts to accurately and efficiently validate the associated control approaches.

One of the most studied properties of control system is stability, which requires that, after a variation of the input, the control brings the system to a steady state. This is typically a state where the output of the system reaches a reference value set in the input. Well-known control theories (e.g., Lyapunov theory) define various sufficient and necessary conditions for the stability of linear systems, where the dynamics is defined by linear differential equations. The standard approach to study stability is based on numerical analysis, which is very effective on linear systyems, but is susceptible of numerical errors. Moreover, one does not have guarantees on the output of the analysis. For example, on complex systems, it may happen that the function obtained with standard optimization algorithms to produce a Lyapunov function is indeed not positive or not decreasing on all points and this can be proved with symbolic techniques.

Related to stability, another relevant problem is the robustness of the control with respect to perturbation of the parameters. In particular, we are interested in finding guarantees that, for small variations of the states or of the references, the system does not switch control.

In this paper, we consider the control design of an aircraft engine. The engine model is represented by a linear state space model of 18 internal state variables, 4 outputs, and 3 inputs. The control switches between two PI controllers, one for thrust control and another for low-pressure compressor spool speed control, based on the engine state and pilot commands. After reformulating the PI controllers in terms of differential equations, we obtain a hybrid system with 21 state variables and two modes, for which we want to prove stability with symbolic techniques.

We first proved stability for specific reference values in each mode using Lyapunov methods and proving the properties of the obtained Lyapunov function with SMT techniques. This activity provided interesting insights on the reliability of numerical methods, as in some cases the synthesized functions were not satisfying the Lyapunov conditions. We also tried to synthesize piecewise Lyapunov for proving stability of the switched system, but all attempts failed due to numerical errors. Finally, we synthesized a region around the stable states that is proved to be robust to perturbation of the state. From this, we derive bounds on the variation of parameters for which we are guaranteed to stabilize without switching control.

Structure of the paper: The rest of the paper is organized as follows. In Section II, we clarify the connections with related

works. In Section III, we recall some basic notions of linear and affine dynamical systems, piecewise-affine switched systems and their stability properties. In Section IV, we explain how we reformulate a linear system whose inputs are coupled with a switched Proportional-Integral (PI) controller into an autonomous piecewise-affine switched system. In Section V, we detail the industrial case study. In Section VI we report on the verification results obtained on the case study. Finally, in Section VII, we draw some conclusions and directions for future works.

II. Related works

Some general approaches to the problem of proving the stability of a piecewise-linear hybrid system are surveyed in [12] and [11]; among them are common Lyapunov functions, piecewise quadratic Lyapunov functions, multiple Lyapunov functions, and higher order Lyapunov functions [9], [4], [15]. Some generalizations of the standard stability properties have also been considered, still with the aim of proving safety properties. In [16] the authors introduce *region stability*, which allows to formalize hybrid systems whose trajectories all eventually end up in a given region even though they may continue to oscillate within the allowance prescribed by the region. In [19] the authors discuss a similar stronger property that they call *persistence*. The idea is again to not require convergence to a single equilibrium point, but instead to ask that the system state eventually becomes always trapped within some set of states. More directly relevant to our problem is the `stabhyli` tool developed by a group at the University of Oldenburg [13]. This tool applies to linear and non-linear hybrid systems, and aims to provide a certificate of stability by means of Lyapunov function techniques combined with a decompositional proof scheme. To the best of our knowledge, this is the largest hybrid system (in terms of state variables) so far treated with symbolic techniques.

III. Background

We denote by $\mathbb{N}$ the set of natural numbers, by $\mathbb{R}$ the set of real numbers and by $\mathbb{R}_+$ the set of non-negative real numbers. We use boldface letters for vectors, e.g. $\boldsymbol{x} \in \mathbb{R}^n$; the transpose of a matrix A is denoted by A^{T}, the Hermitian transpose by $A^\dagger$, and the scalar product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ by $\boldsymbol{a}^{\mathrm{T}}\boldsymbol{b}$.

A. Linear dynamical systems

A *continuous-time linear dynamical system with state space* $\mathbb{R}^n$ is defined by a pair of matrix equations of the form

$$\begin{cases} \dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} \\ \boldsymbol{y} = C\boldsymbol{x} \end{cases} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the *state vector*, $\dot{\boldsymbol{x}}$ denotes the time derivative of $\boldsymbol{x}$, $\boldsymbol{u} \in \mathbb{R}^m$ (for some $m \geq 0$) is the *input vector*, $\boldsymbol{y} \in \mathbb{R}^p$ (for some $p \geq 0$) is the *output vector* and A, B and C are matrices of compatible sizes. The first equation in the system (1) specifies the temporal evolution of the state vector $\boldsymbol{x}$, whereas the second equation defines the outputs of the system as a function of the state. A linear system is called *autonomous* when $m = 0$ (i.e., the system has no inputs). An *equilibrium point* for an autonomous linear system $\dot{\boldsymbol{x}} = A\boldsymbol{x}$ is a state $\boldsymbol{x}_{\mathrm{eq}} \in \mathbb{R}^n$ such that $\dot{\boldsymbol{x}}_{\mathrm{eq}} = 0$.

B. PI controllers

Let $\mathcal{S} = (A, B, C)$ be a linear system, as defined in Section III-A. We assume the matrix A is not singular, so that the system has a single equilibrium point.

Given a vector of *reference values* $\boldsymbol{r} \in \mathbb{R}^p$ for the outputs of $\mathcal{S}$, the corresponding *error vector* is then defined by $\boldsymbol{e} \doteq \boldsymbol{r} - \boldsymbol{y}$. A *PI* (Proportional-Integral) *controller* for $\mathcal{S}$ is defined by imposing an input-output relation of the following form:

$$\boldsymbol{u} = K_P\boldsymbol{e} + K_I \int_0^t \boldsymbol{e}(\tau)\,\mathrm{d}\tau \tag{2}$$

where K_P and K_I are matrices realizing appropriate (linear) functions of the instantaneous error. We assume that every input of system $\mathcal{S}$ is controlled, so that both K_P and K_I are $r \times p$ matrices (not necessarily of full rank).

C. Switched systems

A *switched system* is a dynamical system in which the continuous evolution of the state is mixed with discrete "switching" events which may instantaneously alter the state and/or the evolution law of the system. Switched systems represent one of the simplest class of *hybrid systems* and have been extensively studied in the literature (see e.g. [10]).

In this paper we are only interested in a very specific class of such systems, in which the switching law is:

- *state-dependent*, that is, switching events are determined by the state evolution crossing certain *switching surfaces* defined in the state space;
- *autonomous*, which means that the switching events do not depend on external inputs;
- *continuous*, that is, the switching only involves a change in the evolution law and does not cause a discrete jump in the state.

Thus we assume that the state space $\mathbb{R}^n$ is partitioned into a finite number of *operating regions* or *modes* $(\mathcal{R}_i)_{i\in\mathcal{M}}$ by means of a family of switching surfaces (or guards). In each of these regions a differential equation specifies the evolution of the state variable:

$$\begin{cases} \dot{\boldsymbol{x}} = f_i(\boldsymbol{x}) \\ \boldsymbol{y} = g_i(\boldsymbol{x}) \end{cases} \qquad \text{if } \boldsymbol{x} \in \mathcal{R}_i. \tag{3}$$

Whenever the system trajectory hits a switching surface, the continuous state continues to evolve subject to a different evolution law.

We shall further specialize this setting by adopting the following two assumptions:

1) each (f_i, g_i) is a pair of *affine* functions of $\boldsymbol{x}$, in which case one speaks of a *piecewise-affine system*or PWA system for short. The system (3) then becomes

$$\begin{cases} \dot{\boldsymbol{x}} = A_i\boldsymbol{x} + b_i \\ \boldsymbol{y} = C_i\boldsymbol{x} + d_i \end{cases} \tag{4}$$

for some $n \times n$ matrices $(A_i)_{i\in\mathcal{M}}$, n-dimensional vectors $(b_i)_{i\in\mathcal{M}}$, $p \times n$ matrices $(C_i)_{i\in\mathcal{M}}$ and p-dimensional vectors $(d_i)_{i\in\mathcal{M}}$.

2) We shall assume that each region $\mathcal{R}_i$ is a *convex polytope*, possibly unbounded, in $\mathbb{R}^n$. Such polytopes can always be represented as the intersection of a finite set of half-spaces. A half-space in $\mathbb{R}^n$ can be represented by a vector $\boldsymbol{g} \in \mathbb{R}^n$ and a scalar $h \in \mathbb{R}$, as the locus of solutions of the affine inequality

$$\boldsymbol{g}^{\mathrm{T}}\boldsymbol{x} + h \rhd 0, \tag{5}$$

where $\rhd \in \{\geq, >\}$.

D. Stability

An equilibrium point $\boldsymbol{x}_{\mathrm{eq}}$ of an autonomous dynamical system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ is called:

- *stable* if $\forall \varepsilon > 0\ \exists \delta > 0$ such that $\|\boldsymbol{x}(0) - \boldsymbol{x}_{\mathrm{eq}}\| < \delta$ implies $\|\boldsymbol{x}(t) - \boldsymbol{x}_{\mathrm{eq}}\| < \varepsilon$ for every $t \geq 0$;
- *asymptotically stable* if it is stable and δ may be taken such that $\|\boldsymbol{x}(0) - \boldsymbol{x}_{\mathrm{eq}}\| < \delta$ implies that $\boldsymbol{x}(t)$ converges to $\boldsymbol{x}_{\mathrm{eq}}$ for $t \to \infty$;
- *exponentially stable* if there exist positive reals c, K and λ such that $\|\boldsymbol{x}(t) - \boldsymbol{x}_{\mathrm{eq}}\| \leq K \|\boldsymbol{x}(0) - \boldsymbol{x}_{\mathrm{eq}}\| e^{-\lambda t}$ whenever $\|\boldsymbol{x}(0) - \boldsymbol{x}_{\mathrm{eq}}\| < c$.

In general, exponential stability implies asymptotic stability; for a linear system the opposite implication also holds, so the two concepts are logically equivalent.

A *Lyapunov function* for the system $\dot{\boldsymbol{x}} = f(\boldsymbol{x})$ with equilibrium point $\boldsymbol{x}_{\mathrm{eq}}$ is a function $V\colon \mathbb{R}^n \to \mathbb{R}$ satisfying the following conditions:

1) $V(\boldsymbol{x}_{\mathrm{eq}}) = 0$ and $V(\boldsymbol{x}) > 0$ for every $\boldsymbol{x} \neq \boldsymbol{x}_{\mathrm{eq}}$;
2) $\dot{V}(\boldsymbol{x}) < 0$ for every $\boldsymbol{x} \neq \boldsymbol{x}_{\mathrm{eq}}$, where $\dot{V}$ denotes the Lie derivative of V along the vector field f (i.e., $\dot{V} = \nabla V \cdot f$).

According to *Lyapunov's theorem*, the existence of a Lyapunov function guarantees that the equilibrium point $\boldsymbol{x}_{\mathrm{eq}}$ is asymptotically stable. If in addition V satisfies the stronger property

$$\dot{V}(\boldsymbol{x}) \leq -\alpha V(\boldsymbol{x}) \tag{6}$$

for some $\alpha \in \mathbb{R}_+$ then the equilibrium point is also exponentially stable.

E. Methods for Lyapunov function synthesis

We briefly summarize three different strategies that can be used to synthesize a Lyapunov function for an autonomous linear system $\dot{\boldsymbol{x}} = A\boldsymbol{x}$.

a) Lyapunov equation: The classic method is to look for a *quadratic* Lyapunov function $V(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}P\boldsymbol{x}$ by solving the (continuous-time) *Lyapunov equation*:

$$A^{\mathrm{T}}P + PA + Q = 0 \tag{7}$$

where Q is any symmetric positive definite matrix. This amounts to solving a linear system in $n(n+1)/2$ unknowns (the entries of the symmetric matrix P), which is usually done using numerical algorithms tailored to the specific form of the problem. In the absence of other clues, one often takes Q to be just the identity matrix.

b) Lyapunov function derived from a modal matrix: As a particular case of the previous construction, let us suppose that the matrix A is diagonalizable and let M be any modal matrix for A, i.e. an invertible matrix such that $M^{-1}AM = D$ with D diagonal. The entries of D are exactly the eigenvalues of A; since A is a real matrix, these eigenvalues are either real or pair of complex conjugate numbers. Then the matrix product

$$P = {M^{-1}}^{\dagger}M^{-1} \tag{8}$$

defines a Lyapunov function for the linear system $\dot{\boldsymbol{x}} = A\boldsymbol{x}$. Indeed it is not hard to verify that P solves the Lyapunov equation (7) for $Q = -{M^{-1}}^{\dagger}(D+\overline{D})M^{-1}$. This matrix is real and symmetric by construction and positive definite when the matrix A is stable, since $D + \overline{D} = 2\mathrm{Re}\, D$ and the eigenvalues of a stable matrix have strictly negative real part.

Compared to the previous method, here we trade the task of solving the linear system (7) with the task of diagonalizing (and finding a modal matrix for) the matrix A.

c) Lyapunov function synthesis via LMIs: Another way to synthesize the matrix P is by solving a system of *linear matrix inequalities* (LMI) [3]. In its simplest version, the LMI problem reads as follows: find $P = P^{\mathrm{T}}$ such that

$$\begin{cases} P > 0 \\ A^{\mathrm{T}}P + PA < 0 \end{cases} \tag{9}$$

Here a notation like $X > 0$ (resp. $X < 0$) means that the matrix X is required to be positive definite (resp. negative definite). The advantage of this setting is that one can modify slightly the LMI problem (9) in order to obtain Lyapunov functions with stroger properties. For instance, a solution for the LMI problem

$$\begin{cases} P > 0 \\ A^{\mathrm{T}}P + PA + \alpha P < 0 \end{cases} \tag{10}$$

where $\alpha \in \mathbb{R}_+$ is a fixed parameter, yields a Lyapunov function satisfying property (6); the best possible value of the parameter α gives a quantitative measure of the speed of convergence to the equilibrium point which can be used to estimate the settling time of the system.

F. Lyapunov functions for hybrid systems

The theory of Lyapunov stability also applies to hybrid systems; we refer the reader to [14] for a review of this topic. Assuming that the PWA system (4) has a single (globally stable) equilibrium point, we can try to synthesize a global Lyapunov function which directly witnesses this fact. The simplest approach is to consider a *piecewise-quadratic Lyapunov function*, namely a function defined by a quadratic form $\boldsymbol{x}^{\mathrm{T}}P_i\boldsymbol{x}$ for each region $\mathcal{R}_i$, where the matrices $(P_i)_{i\in\mathcal{M}}$ are parameterized in such a way to ensure the continuity of $V(\boldsymbol{x})$ on each boundary between regions. Such a Lyapunov function can be synthesized using an appropriate generalization of the LMI problem (9), in which the global Lyapunov constraints are applied locally in each region using the so-called *S-procedure* [3]. The details of the LMI problem used can be found in

[14, Theorem 3.10]. A drawback of this approach is that the class of Lyapunov functions considered in this way can be too rigid, precluding the successful solution of the LMI problem. The S-procedure is also known to be conservative in general, providing only a sufficient (but not necessary) condition for the local validity of the Lyapunov conditions.

IV. Switched PI Controller and Reformulation of the Closed-Loop Dynamics

A. Switched PI Controller

In this paper we are interested in *switching* controllers, that is controllers in which the linear functions appearing in the feedback law (2) may change according to some *switching conditions* formulated as linear inequalities on the outputs of the system. Accordingly, the matrices K_P and K_I appearing in equation (2) are replaced by a pair of finite set of matrices,

$$(K_{I,i})_{i\in\mathcal{M}} \qquad (K_{P,i})_{i\in\mathcal{M}} \tag{11}$$

where $\mathcal{M}$ is the set of operating modes of the switching controller. For each mode $i \in \mathcal{M}$ the input-output relation becomes

$$\boldsymbol{u} = K_{P,i}\boldsymbol{e} + K_{I,i}\int_0^t \boldsymbol{e}(\tau)\,\mathrm{d}\tau. \tag{12}$$

By the assumptions made in III-C, for each $i \in \mathcal{M}$ we can write the activating conditions of mode i as a finite system of affine inequalities of the form

$$\begin{cases} (\boldsymbol{g}_1^{(i)})^{\mathrm{T}}\boldsymbol{y} + h_1^{(i)} \rhd_1^{(i)} 0 \\ \vdots \qquad\qquad \text{for some } \ell_i > 0, \\ (\boldsymbol{g}_{\ell_i}^{(i)})^{\mathrm{T}}\boldsymbol{y} + h_{\ell_i}^{(i)} \rhd_{\ell_i}^{(i)} 0 \end{cases} \tag{13}$$

where $\rhd_1^{(i)}, \ldots, \rhd_{\ell_i}^{(i)} \in \{\geq, >\}$, $\boldsymbol{g}_1^{(i)}, \ldots, \boldsymbol{g}_{\ell_i}^{(i)}$ are vectors in $\mathbb{R}^p$ and $h_1^{(i)}, \ldots, h_{\ell_i}^{(i)}$ are scalars in $\mathbb{R}$.

Substituting $\boldsymbol{y} = C\boldsymbol{x}$ we can rewrite the k-th inequality in the system (13) as

$$(\boldsymbol{g}_k^{(i)})^{\mathrm{T}} C\boldsymbol{x} + h_k^{(i)} \geq 0 \tag{14}$$

B. Reformulation into a Switched System

Let $\mathcal{S} = (A, B, C)$ be the open-loop linear system and $\pi = (K_{P,i}, K_{I,i})_{i\in\mathcal{M}}$ the associated switching PI controller.

To model the closed-loop system obtained by the feedback connection between S and π we build a PWA switched system as follows.

- The *state space* is $\mathbb{R}^{n+r}$, coordinatized by the vector obtained by concatenating the state vector $\boldsymbol{x} \in \mathbb{R}^n$ and the input vector $\boldsymbol{u} \in \mathbb{R}^r$:

$$\boldsymbol{w} \doteq \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix} \tag{15}$$

- The *set of modes* is the same set $\mathcal{M}$ specified by the switching controller π; the corresponding partition of the state space is defined by reinterpreting the conditions (13) as inequalities on $\mathbb{R}^{n+r}$ which do not involve the coordinates $w_{n+1} = u_1, \ldots, w_{n+r} = u_r$.
- The *flow* in region $\mathcal{R}_i$ ($i \in \mathcal{M}$) is given by the differential equations of the original linear system (with $\boldsymbol{u}$ reinterpreted as a state variable),

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} \tag{16}$$

supplemented by the differential equations obtained by taking the time derivative of both sides of the PI control relation (12) (assuming constant reference values):

$$\dot{\boldsymbol{u}} = -K_{P,i}\dot{\boldsymbol{y}} + K_{I,i}(\boldsymbol{r} - \boldsymbol{y}) \tag{17}$$

Substituting $\boldsymbol{y} = C\boldsymbol{x}$ and rearranging we get

$$\dot{\boldsymbol{u}} = -K_{P,i}C\dot{\boldsymbol{x}} - K_{I,i}C\boldsymbol{x} + K_{I,i}\boldsymbol{r} \tag{18}$$

and using equation (16) we obtain finally

$$\dot{\boldsymbol{u}} = (-K_{P,i}CA - K_{I,i}C)\boldsymbol{x} - K_{P,i}CB\boldsymbol{u} + K_{I,i}\boldsymbol{r} \tag{19}$$

In terms of the vector $\boldsymbol{w}$, the system of ODEs consisting of equations (16) and (19) can be written more compactly as

$$\dot{\boldsymbol{w}} = \begin{pmatrix} A & B \\ N_i & M_i \end{pmatrix}\boldsymbol{w} + \begin{pmatrix} 0 \\ K_{I,i} \end{pmatrix}\boldsymbol{r} \tag{20}$$

where we have defined $N_i \doteq -K_{P,i}CA - K_{I,i}C$ and $M_i \doteq -K_{P,i}CB$.

- The *outputs* of the reformulated systems are simply the outputs of $\mathcal{S}$, extended to the new state vector in the trivial way (no dependence on $\boldsymbol{u}$):

$$\boldsymbol{y} = \begin{pmatrix} C & 0 \end{pmatrix}\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix}. \tag{21}$$

We shall denote by $\mathcal{S}_\pi$ the reformulated system obtained in this way. Note that $\mathcal{S}_\pi$ is an *autonomous* switched PWA system, so its stability properties can be analyzed using the standard tools recalled in Section III.

V. Case Study

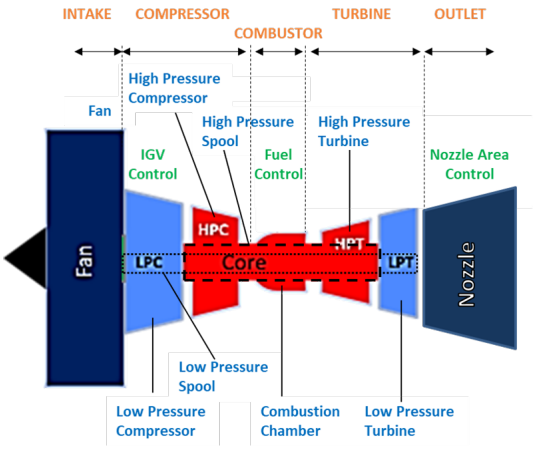

Fig. 1. An example of a dual spool engine

A. Basic Engine Operation

A basic outline of an aircraft turbofan engine is depicted in Figure 1; the main components, in order from left to right, are: the inlet (engine front), the compressor (low and high pressure stages), the combustion chamber, the gas turbine (high and low stages) and the exhaust nozzle (back of the engine). An aircraft engine provides a constant supply of air for the pressure vessel, passing through the two-stage compression operation (low and high pressure) and mixed with fuel in the combustion chamber to generate thrust. Several control sections must be governed by the engine control system to perform basic steps of intake, compression, combustion, and exhaust. During this operation, and apart from performance requirements, several critical safety parameters need to be respected concerning safety. These parameters relate to engine surge and stall avoidance, combustion chamber temperature limits and so on. It is, therefore, of paramount importance that any control approach designed on the engine's embedded controllers be verified and certified against certain safety requirements.

B. Use Case Description

The framework presented in the paper is demonstrated for a jet engine control system application. More specifically, the control design problem of a turbofan engine in [18], [17] is addressed using single input single output PI (proportional plus integral) controllers [5] in contrast to the multivariable controllers in [18], [17]. The control architecture choice above is made because of the simplicity of the PI controller as opposed to the multivariable designs in [18], [17]. This enables a lean control implementation with minimal complexity and creates a suitable use case that is comprehensible by non-experts in control. This by no means limits the scalability of the analysis and method of this paper, which is equally applicable to more complex control system designs as in [18], [17]. Such demonstrations are outside the illustration purposes of the present paper and would be the subject of future research.

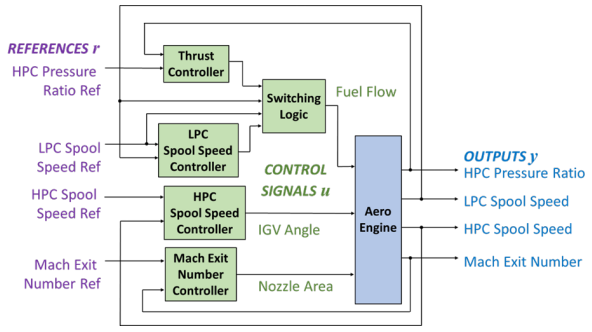

Fig. 2. UC5 engine control model

A block diagram of the control system under study is shown in Figure 2. There are four PI controllers, controlling the Low Pressure Compressor (LPC) Spool Speed, the High Pressure Compressor (HPC) Pressure Ratio, the Mach Exit Number and the HPC Spool Speed. The above quantities are denoted as y_0, y_1, y_2 and y_3, respectively, and constitute the engine's four measured output signals which are further grouped as the system (engine) output column vector $\boldsymbol{y} = \begin{pmatrix} y_0 & y_1 & y_2 & y_3 \end{pmatrix}^{\mathrm{T}}$. The respective desired reference values for the engine outputs, denoted by r_0, r_1, r_2, and r_3 are the command signals coming from a supervisory engine management system and grouped as the system reference (command) vector $\boldsymbol{r} = \begin{pmatrix} r_0 & r_1 & r_2 & r_3 \end{pmatrix}^{\mathrm{T}}$. As Figure 2 shows, each reference signal and the corresponding engine output are inputs to a single PI controller. The signal outputs of the PI controllers are used to determine the actuation signals for the Fuel flow, the Nozzle Area at the engine exhaust, and the Inlet Gain Vane (IGV) Angle at the HPC. The above three control signals are denoted as u_0, u_1, u_2 and they form the control signal vector $\boldsymbol{u} = \begin{pmatrix} u_0 & u_1 & u_2 \end{pmatrix}^{\mathrm{T}}$. It is worth noting that the Fuel Flow actuation signal u_0 is determined by switching operational modes between the Thrust and LPC Spool Speed controllers. More specifically, if y_0 is greater than r_0, then the Flow Rate is determined by the signal output of the LPC Spool Speed controller; otherwise, it is determined by the minimum of HPC Pressure Ratio and LPC Spool Speed control outputs. The above switching operation ensures safety protection from engine compressor surge instabilities by limiting the LPC Spool Speed (r_0 command). On the other hand, when no LPC Spool Speed constraints are enforced (i.e. $y_0 \leq r_0$) then the minimum signal selection above will always result in fuel savings [17]. Whenever the switching between the Thrust and the LPC Spool Speed controller occurs, the integrator of the activated controller is initialised (reset) to the value of the current state of the deactivated integrator. The Nozzle Area actuation signal u_1 is determined solely by the Mach Exit Number controller, while the IGV Angle actuation signal u_2 is determined solely by the HPC Spool Speed controller.

The mathematical model of the engine system in [18], [17] is a linear system of eighteen continuous time ordinary differential equations that can be written in matrix form as in Equation (1), with $\boldsymbol{x} \in \mathbb{R}^{18}$ the vector of internal variables and $\boldsymbol{y}$, $\boldsymbol{u}$ the engine output and control signal vectors as described previously. The engine model parameters A, B and C, are constant real matrices of appropriate dimensions, the numerical values of which can be found in [18].

The PI controllers designs in continuous time [5] are given by the equations (12), with $\mathcal{M} = \{0, 1\}$ as the set of modes. The matrices (11) expressing the integral and proportional controller gains are given by

$$K_{I,0} = \begin{pmatrix} 10 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix} \quad K_{I,1} = \begin{pmatrix} 0 & 20 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

$$K_{P,0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0.5 \end{pmatrix} \quad K_{P,1} = \begin{pmatrix} 0 & 0.1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0.5 \end{pmatrix}$$

Based on the previously mentioned functional operation of the control system and the safety switching, the switching law is determined as follows:

$$i = \begin{cases} 0 & \text{if } r_0 - y_0 < \Theta \\ 1 & \text{otherwise} \end{cases}$$

where Θ is a safety margin to switch the control. In our case, we fix $\Theta = 1$. The two operating regions $\mathcal{R}_0$ and $\mathcal{R}_1$ are then defined (in the notations of Section IV) by taking $\ell_0 = \ell_1 = 1$ and

$$\begin{array}{lll} \boldsymbol{g}^{(0)} = (1,0,0,0)^{\mathrm{T}} & h^{(0)} = \Theta - r_0 & \rhd^{(0)} = > \\ \boldsymbol{g}^{(1)} = (-1,0,0,0)^{\mathrm{T}} & h^{(1)} = r_0 - \Theta & \rhd^{(1)} = \geq \end{array}$$

with $\mathcal{R}_0$ corresponding to the nominal operation region.

Applying the reformulation described in Section IV-B, we obtain a hybrid system of the form

$$\dot{\boldsymbol{w}} = \begin{cases} A_0\boldsymbol{w} + B_0\boldsymbol{r} \text{ if } \boldsymbol{g}^T\boldsymbol{w} < h \\ A_1\boldsymbol{w} + B_1\boldsymbol{r} \text{ otherwise} \end{cases} .$$

VI. Verification

A. Reduced models

The entire system consisting of 18 internal variables is difficult to handle. To assess the scalability of the methods presented, we consider several reduced models that preserve some properties of the full system while being more approachable from the computational point of view. By using *Balanced Truncation Model Reduction* on the full system, we obtained reduced models with $3, 5, 10, 15$ state variables respectively. For sizes $3, 5, 10$, we also consider truncated version of the system matrices, obtained by rounding entries to the closest integer values.

B. Synthesis and validation of Lyapunov

1) Single modes: For the synthesis of Lyapunov functions for a single operating mode, we consider 6 different methods derived from the strategies described in Section III-E.

- `eq-smt`: solves symbolically Equation 7 with SymPy;
- `eq-num`: solves numerically Equation 7 with module Control;
- `modal`: numerically derives a Lyapunov function from a modal matrix, as in Equation 8;
- `LMI`: finds a Lyapunov function solving the LMI problem of Equation 9 using Picos module, which gives access to different backend solver for the semidefinite programming: CVXOPT, Mosek, SMCP;
- `LMI`α: modifies the problem of method `LMI` with additional parameters for α, as in Equation 10;
- `LMI`α+: modifies the problem of method `LMI`α by using constraint $P - \nu I > 0$, where $\nu \in \mathbb{R}_+$ is a fixed parameter. In this way, we force the solution to have greater eigenvalues.

The function obtained applying these methods is of the form

$$V_i^{\boldsymbol{r}}(\boldsymbol{w}) = (\boldsymbol{w} - \boldsymbol{w}_{\mathrm{eq}_i})^T P_i (\boldsymbol{w} - \boldsymbol{w}_{\mathrm{eq}_i})$$

where $\boldsymbol{w}_{\mathrm{eq}_i}$ is the equilibrium point of each linear dynamical system. With the only exception of `eq-smt`, the aforementioned methods are numerical and can only synthesize a *candidate* Lyapunov function, whose correctness with respect to requirements 1 and 2 (described in Section III-D) is still to be validated. We do so symbolically by employing SMT solvers and check the validity of conditions 1 and 2, expressed as quantifier-free non-linear real arithmetic formulae.

Experimental setup: We ran out experiments on 2.40GHz CPUs with 30 GB memory limit and 2 hours time limit.

In Table I we report the results of the synthesis and validation of Lyapunov functions from every operating mode of the considered models. For the different benchmarks, clusterized based on their size (columns "size 3",... "size 18"), and for all synthesis method used, we report the average time for obtaining a candidate Lyapunov function (column "synth.time"), and the ratio of successfully validated ones (column "valid"). Between the synthesis methods, solver SMCP is the least efficient one. Nonetheless, the 192 cases, time outs (i.e., cases where none of the tried validation methods succeeded) interest only the `eq-smt` approach, for sizes 15 and 18. In only two instances the numerical methods returned an invalid result (`LMI`α+ with Mosek for both operating modes of size 18). The candidate Lyapunov functions were validated rounding the entries of the numerically synthesized matrices at the 10th significative figure. Rounding at the 6th and 4th significative figure led to find 4 and 25 invalid entries respectively. The only method that could synthesize valid Lyapunov functions even when rounding at the 4th significant figure is `LMI`α.

In Figure 3 we evaluate the times spent in validating the candidate Lyapunov functions on different symbolic solvers: SymPy's `is_positive_definite` procedure, an ad-hoc implementation of the Sylvester's method [7, Theorem 7.2.5], Mathematica [8], z3 [6] and cvc5 [2] as SMT solvers. For the latter cases, we consider a "+ det" option, where the check $\forall \boldsymbol{w} \neq 0 : V_i(\boldsymbol{w}) > 0$ (resp. $\forall \boldsymbol{w} \neq 0 : \dot{V}_i(\boldsymbol{w}) < 0$) is encoded with $\forall \boldsymbol{w} : V_i(\boldsymbol{w}) \geq 0 \wedge \det(P_i) \neq 0$ (resp. $\forall \boldsymbol{w} : \dot{V}_i(\boldsymbol{w}) \leq 0 \wedge \det(A_i^T P_i + P_i A_i) \neq 0$). From the plot, we observe that Mathematica highly benefits from this version of the check, but, due to the initialization time required by such SMT solver, Sylvester's method is the validation method performing best.

2) Switched System: We applied the approach described in Section III-F to synthesize a piecewise quadratic Lyapunov function [14, Theorem 3.10]. We applied two different LMI formulations to take care of the non-increasing conditions necessary for the Lyapunov functions on the switching surface (i.e., the value of the switched Lyapunov function should not increase when switching mode). The first LMI formulation requires the value of the Lyapunov functions for each mode to be equal on the switching surface. The second LMI encoding provides a more relaxed problem formulation, requiring the Lyapunov function for each mode to not increase close to the surface (while it can decrease exactly on the switch). We experimented with both LMI encoding on the set of reduced models. While the LMI solver always finds a candidate Lyapunov function, the subsequent validation using an SMT solver

		size 3		size 5		size 10		size 15		size 18	
method	solver	synth.time	valid	synth.time	valid	synth.time	valid	synth.time	valid	synth.time	valid
`eq-smt`		1.1	4 / 4	2.25	4 / 4	314.26	4 / 4	TO	0 / 2	TO	0 / 2
`eq-num`		0.02	4 / 4	0.03	4 / 4	0.1	4 / 4	0.2	2 / 2	0.26	2 / 2
`modal`		0.02	4 / 4	0.03	4 / 4	0.09	4 / 4	0.2	2 / 2	0.25	2 / 2
`LMI`	cvxopt	0.05	4 / 4	0.06	4 / 4	0.16	4 / 4	0.34	2 / 2	0.6	2 / 2
`LMI`	mosek	0.06	4 / 4	0.08	4 / 4	0.25	4 / 4	0.35	2 / 2	0.5	2 / 2
`LMI`	smcp	0.52	4 / 4	3.53	4 / 4	12.68	4 / 4	28.03	2 / 2	67.14	2 / 2
`LMI`α	cvxopt	0.33	4 / 4	0.32	4 / 4	0.83	4 / 4	1.76	2 / 2	3.93	2 / 2
`LMI`α	mosek	0.19	4 / 4	0.35	4 / 4	0.7	4 / 4	1.56	2 / 2	2.46	2 / 2
`LMI`α	smcp	4.24	4 / 4	38.88	4 / 4	152.3	4 / 4	519.98	2 / 2	901.57	2 / 2
`LMI`α+	cvxopt	0.36	4 / 4	0.53	4 / 4	1.36	4 / 4	3.38	2 / 2	5.71	2 / 2
`LMI`α+	mosek	0.33	4 / 4	0.37	4 / 4	0.6	4 / 4	1.15	2 / 2	1.38	0 / 2
`LMI`α+	smcp	5.0	4 / 4	78.69	4 / 4	289.28	4 / 4	1099.74	2 / 2	2340.19	2 / 2

TABLE I
SYNTHESIS AND VALIDATION OF LYAPUNOV FUNCTIONS.

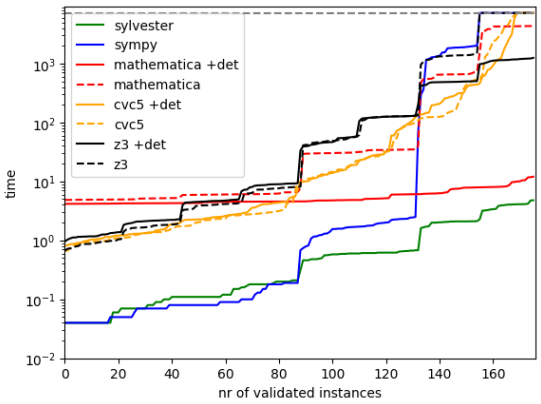

Fig. 3. Validation time with different solvers.

always fails. In particular, in our experiments the validation of the constraint requiring the Lyapunov function to not increase on the switching surface always failed.

C. Robustness to Perturbation

1) Synthesis of Robust Regions: Leveraging the computed Lyapunov functions for the single operating modalities of the system, we now try to synthesize robust regions around the stable states. Given an assignment to the reference values $\boldsymbol{r}$, let $\boldsymbol{w}_{\text{eq}_i}$ and $V_i(\boldsymbol{w})$ [1] be the equilibrium point and a Lyapunov function of modality $i \in \mathcal{M}$. For the purposes of this analysis, we consider reference assignments where, for all $i \in \mathcal{M}$, $\boldsymbol{w}_{\text{eq}_i} \in \mathcal{R}_i$. For all mode $i \in \mathcal{M}$, we want to find a region around $\boldsymbol{w}_{\text{eq}_i}$ whose points are guaranteed to converge to $\boldsymbol{w}_{\text{eq}_i}$ without switching operating mode. Namely, we look for k_i such that

$$\forall \boldsymbol{w} : (V_i(\boldsymbol{w}) \leq k_i \wedge sw = 0 \implies \beta_i) \tag{22}$$

where $sw = 0$ is the switching surface between $\mathcal{R}_0$ and $\mathcal{R}_1$, i.e., $\boldsymbol{g}^T\boldsymbol{w} - h = 0$, and β_i is its subset where $\dot{sw}(\boldsymbol{w})$ points towards $\mathcal{R}_i$, according to the dynamics of operating mode i. We do so by computing candidate k_i values with numerical methods, and by using the SMT solver Mathematica to validate requirement 22 (and to prove that it is optimal up to a 10^{-3} factor). The robust region will then be given by the truncated ellipsoid $W_i = \{V_i(\boldsymbol{w}) \leq k_i\} \wedge \mathcal{R}_i$.

2) Robustness to reference value changes: We want to address the problem of robustness with respect to changes of reference values $\boldsymbol{r}$. To stress the dependecy on $\boldsymbol{r}$ of the objects that we use, we write $W_i^{\boldsymbol{r}}$ instead of W_i and $\boldsymbol{w}_{\text{eq}_i}^{\boldsymbol{r}}$ instead of $\boldsymbol{w}_{\text{eq}_i}$. We find an $\epsilon_i > 0$ (that depends on $\boldsymbol{r}$) such that if $\boldsymbol{r'}$ is in the ball $\mathcal{B}(\boldsymbol{r}, \epsilon_i)$ centered in $\boldsymbol{r}$ and of radius ϵ_i, then $\boldsymbol{w}_{\text{eq}_i}^{\boldsymbol{r}} \in W_i^{\boldsymbol{r'}}$. In order to obtain such ϵ_i, we need to estimate how much the robust region $W_i^{\boldsymbol{r}}$ changes when we change the parameters. We denote the spectral norm of a matrix by $\|\cdot\|_2$.

If the vector field $\dot{\boldsymbol{w}} = A_i\boldsymbol{w} + B_i\boldsymbol{r}$ is constant on the switching surface $sw = 0$, the stable region $W_i^{\boldsymbol{r}}$ is the whole $\mathcal{R}_i$. This happens for every reference value $\boldsymbol{r'}$ for which the stable point $\boldsymbol{w}_{\text{eq}_i}^{\boldsymbol{r'}}$ is in $\mathcal{R}_i$, hence we just have to check this condition. We can make sure that $\boldsymbol{w}_{\text{eq}_i}^{\boldsymbol{r'}} \in \mathcal{R}_i$ by taking

$$\epsilon_i = \frac{\text{dist}(\boldsymbol{w}_{\text{eq}_i}^{\boldsymbol{r}}, sw = 0)}{\left\|A_i^{-1}B_i\right\|_2}.$$

If the vector field $\dot{\boldsymbol{w}} = A_i\boldsymbol{w} + B_i\boldsymbol{r}$ is not constant on the switching surface $sw = 0$, we call $\boldsymbol{p}$ the orthogonal projection of $A_i^T\boldsymbol{g}$ on $\boldsymbol{g}^\perp$. Notice that $\boldsymbol{p} \neq 0$. We also call $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n > 0$ the eigenvalues of P_i. We fix the following quantities:

- $\alpha > 0$ such that $\mathcal{B}(\boldsymbol{w}_{\text{eq}_i}^{\boldsymbol{r}}, \alpha) \subseteq W_i^{\boldsymbol{r}}$;
- $\beta = \left\|A_i^{-1}B_i\right\|_2 > 0$;
- $\gamma = \frac{\|\boldsymbol{g}B_i\|}{\|\boldsymbol{p}\|} > 0$;
- $\delta = \text{dist}(\boldsymbol{w}_{\text{eq}_i}^{\boldsymbol{r}}, sw = 0) > 0$;
- $\mu = \sqrt{\frac{\mu_n}{\mu_1}} > 0$.

We can make sure that $\boldsymbol{w}_{\text{eq}_i}^{\boldsymbol{r}} \in W_i^{\boldsymbol{r'}}$ by taking

$$\epsilon_i = \min\left\{\frac{\alpha\mu}{\mu(\beta+\gamma)+\beta}, \frac{\delta}{\beta}\right\} > 0.$$

3) Results: In Table II we show the results of the synthesis of robust stable regions and of the computation of the radius of the ball in the space of parameters centered in $\boldsymbol{r}$. Due to space constraints, we report only the results regarding the two largest systems (size 15 and size 18). Column "time" reports the time

[1] It can be computed by translating the one found for the homogeneous system (i.e., with $\boldsymbol{r} = 0$) by $\boldsymbol{w}_{\text{eq}_i}$.

		size 15						size 18					
		mode 0			mode 1			mode 0			mode 1		
method	solver	time	vol	ϵ	time	vol	ϵ	time	vol	ϵ	time	vol	ϵ
eq-num		286	7e-10	7e-7	235	1e-4	2e-6	808	**5e+38**	4e-9	916	**9e+44**	1e-8
modal		161	7e-18	3e-6	148	7e-10	1e-5	680	2e+31	2e-10	679	3e+37	5e-10
LMI	cvxopt	302	8e+0	3e-5	307	1e+6	7e-5	642	2e+26	**3e-8**	569	3e+32	**8e-8**
LMI	mosek	321	9e+0	3e-5	324	1e+6	6e-5	707	3e+26	3e-8	713	7e+32	8e-8
LMI	smcp	295	9e+0	3e-5	310	1e+6	7e-5	558	9e+25	2e-8	547	2e+32	7e-8
LMIα	cvxopt	309	**1e+1**	2e-5	199	**1e+6**	5e-5	769	1e+25	2e-8	594	4e+32	6e-8
LMIα	mosek	189	8e+0	2e-5	167	1e+6	5e-5	692	1e+25	2e-8	692	3e+32	6e-8
LMIα	smcp	226	1e+1	2e-5	198	1e+6	5e-5	747	7e+24	1e-8	799	2e+32	6e-8
LMIα+	cvxopt	276	1e+0	2e-5	281	1e+5	5e-5	803	2e+25	2e-8	731	5e+32	7e-8
LMIα+	mosek	255	6e+0	**3e-5**	280	8e+5	**7e-5**	-	-	-	-	-	-
LMIα+	smcp	257	5e+0	2e-5	198	7e+5	6e-5	555	1e+25	2e-8	760	3e+32	7e-8

TABLE II
SYNTHESIS OF ROBUST REGION

in seconds needed to compute k_i; column "vol" reports the volume of the truncated ellipsoid W_i; column "ϵ" reports the radius of the ball $\mathcal{B}(\boldsymbol{r}, \epsilon)$ such that if $\boldsymbol{r'} \in \mathcal{B}(\boldsymbol{r}, \epsilon)$, then $\boldsymbol{w}_{\mathrm{eq}_i^{\boldsymbol{r}}} \in W_i^{\boldsymbol{r'}}$. For each problem, we highlight the maximum value for the volume of the robust region and the robustness ϵ.

VII. Conclusions and Future Works

In this paper, we considered the problem of proving with symbolic SMT-based techniques the stability of a switched control system, and its robustness to perturbation. We targeted an industrial control system for an aircraft engine represented by a linear state space model with 18 state variables, 4 outputs and 3 inputs, while the control switches between two PI controllers. We successfully proved stability in each mode and provided formal guarantees on the robustness to small changes to the state or to the references.

The directions of potential future works are manifold. In fact, this case study provides an industrial-size benchmark of general interest for formal verification. We will therefore archive it for the Competition on Applied Verification for Continuous and Hybrid Systems (see, e.g., [1]). We will investigate the problem of the stability and robustness across the switching modes. We will consider generalizing the approach to a wider set of temporal properties. We will find connections among the method used to synthesize the Lyapunov function and the quantities shown in Table II.

Acknowledgments: This work has been developed in the VALU3S project, which has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876852. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Czech Republic, Germany, Ireland, Italy, Portugal, Spain, Sweden, and Turkey. It has been partly supported by project "AI@TN" funded by the Autonomous Province of Trento and produced with the support of the Autonomous Province of Trento.

References

[1] Matthias Althoff, Erika Ábrahám, Marcelo Forets, Goran Frehse, Daniel Freire, Christian Schilling, Stefan Schupp, and Mark Wetzlinger. ARCH-COMP21 category report: Continuous and hybrid systems with linear continuous dynamics. In *ARCH@ADHS*, volume 80 of *EPiC Series in Computing*, pages 1–31. EasyChair, 2021.

[2] Haniel Barbosa, Clark W. Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, Abdalrhman Mohamed, Mudathir Mohamed, Aina Niemetz, Andres Nötzli, Alex Ozdemir, Mathias Preiner, Andrew Reynolds, Ying Sheng, Cesare Tinelli, and Yoni Zohar. cvc5: A versatile and industrial-strength SMT solver. In *TACAS (1)*, volume 13243 of *Lecture Notes in Computer Science*, pages 415–442. Springer, 2022.

[3] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[4] Michael S Branicky. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on automatic control*, 43(4):475–482, 1998.

[5] John Joachim D'Azzo and Constantine Dino Houpis. *Linear Control System Analysis and Design: Conventional and Modern*. McGraw-Hill Higher Education, 4th edition, 1995.

[6] Leonardo Mendonça de Moura and Nikolaj S. Bjørner. Z3: an efficient SMT solver. In *TACAS*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008.

[7] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[8] Wolfram Research, Inc. Wolfram engine. https://www.wolfram.com/engine/. Champaign, IL, 2022.

[9] Mikael Johansson and Anders Rantzer. Computation of piecewise quadratic lyapunov functions for hybrid systems. In *1997 European Control Conference (ECC)*, pages 2005–2010. IEEE, 1997.

[10] Daniel Liberzon. *Switching in systems and control*, volume 190. Springer, 2003.

[11] Hai Lin and Panos J Antsaklis. Stability and stabilizability of switched linear systems: a survey of recent results. *IEEE Transactions on Automatic control*, 54(2):308–322, 2009.

[12] Jan Lunze and Françoise Lamnabhi-Lagarrigue. *Handbook of hybrid systems control: theory, tools, applications*. Cambridge University Press, 2009.

[13] Eike Möhlmann and Oliver E. Theel. Stabhyli: a tool for automatic stability verification of non-linear hybrid systems. In *HSCC*, pages 107–112. ACM, 2013.

[14] Jens Oehlerking. *Decomposition of stability proofs for hybrid systems*. PhD thesis, Universität Oldenburg, 2011.

[15] Antonis Papachristodoulou and Stephen Prajna. On the construction of lyapunov functions using the sum of squares decomposition. In *CDC*, pages 3482–3487. IEEE, 2002.

[16] Andreas Podelski and Silke Wagner. Region stability proofs for hybrid systems. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 320–335. Springer, 2007.

[17] Raza Samar and Ian Postlethwaite. Design and Implementation of a Digital Multimode H_∞ Controller for the Spey Turbofan Engine. *Journal of Dynamic Systems, Measurement, and Control*, 132(1), 12 2009. 011010.

[18] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2nd edition, 2005.

[19] Andrew Sogokon, Paul B Jackson, and Taylor T Johnson. Verifying safety and persistence in hybrid systems using flowpipes and continuous invariants. *Journal of Automated Reasoning*, 63(4):1005–1029, 2019.