Optimal Segmentation of LPV Systems for Control Applications via Genetic Algorithms

Zoltán Téczely[a] and Bálint Kiss[b]
Budapest University of Technology and Economics, Department of Control Engineering and Information Technology, Műegyetem rkp. 3, H-1111 Budapest, Hungary

Keywords: LPV, Polytopic, Conservatism, LMI, Genetic Algorithms, Global Optimization.

Abstract: The paper presents an automatic method for subdividing parameter regions in a Linear Parameter-Varying (LPV) controlled system based on global optimization. A known limitation of the LPV framework is the conservatism originating from excessive parameter regions. This conservatism can be relaxed if the controller design is performed in a collection of subregions of the parameter bounding box wherein local controllers are synthesized yielding an increased performance level. The choice of subregion boundaries, however, is usually based on heuristics. This, combined with the recurring issue of scheduling variable selection motivates an automated LPV parameter space description. The paper suggests genetic algorithms to automate parameter space subdivision where the problem is posed in terms of global optimization, considering closed-loop performance, computational complexity and parameter-dependent performance constraints. The benefits of the proposed approach are demonstrated on a pitch-axis missile autopilot, which is formulated as a quasi-LPV model but generally does not admit the polytopic framework. Hence, the necessary simplifications and selection criteria are introduced to effectively employ polytopic LPV methods in the vertical acceleration control for such a missile.

1 INTRODUCTION

Nonlinear mechatronic systems are often modelled and controlled based on the polytopic Linear Parameter-Varying framework. Numerically efficient controller synthesis is usually formulated as a set of linear matrix inequality (LMI) problems largely built around the works of (Gahinet and Apkarian, 1994) and (Apkarian and Gahinet, 1995). Arguably the most important limitation characterizing LPV control is the conservatism introduced by the definition of the parameter region which might be a result of inadequate knowledge of the system operating range or geometric and computational complexity. Traditionally, in the overwhelming majority of research efforts, a large emphasis has been put on parameter dimension reduction, e.g. (Kwiatkowski and Werner, 2008) or the notion of parameter-dependent Lyapunov functions (PDLFs) (Gahinet et al., 1996). The celebrated paper (Scherer, 2001) introduced the idea of subdivision of the parameter space stating that, performance can be increased arbitrarily, while (Kruszewski et al., 2009) uses a fuzzy approach. Regarding the transition between two sets of polytope regions, several famous articles have been presented including (Lu and Wu, 2004) and (Yan and Özbay, 2007) stating the theoretical problem whilst more recent advancements tend to be more application-oriented e.g. (Jiang et al., 2015) and (Huang et al., 2021) and (Robert et al., 2007). Most of these articles are concerned with multiple Lyapunov functions assigned to neighbouring systems resulting in the problem of switching LPV systems. Although these methods form the basis of current developments in LPV control, a more sophisticated mathematical effort prevents them from being used day-to-day in engineering practices. Although polytope segmentation is nothing new, an automated framework for scheduling variable selection and dealing with more complicated nonlinear matters is still lacking. On the other hand, computational complexity issues and along with them, associated solutions typically originate from either high-complexity structures or inherently numerically demanding models like flows (Das and Heiland, 2023).

The primary contribution of this paper, therefore, is a systematic method for designing segmented poly-

[a] https://orcid.org/0009-0005-7592-9038
[b] https://orcid.org/0000-0002-5532-3435

Téczely, Z. and Kiss, B.
Optimal Segmentation of LPV Systems for Control Applications via Genetic Algorithms.
DOI: 10.5220/0013067200003822
In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024) - Volume 1*, pages 503-510
ISBN: 978-989-758-717-7; ISSN: 2184-2809

topic LPV controllers. The proposed method uses a common Lyapunov function as the basis of the solution and partitions the parameter space into multiple smaller sub-polytopes according to a cost function involving a joint metric describing the performance gain and additional complexity. A secondary contribution is a case study on a missile, wherein the key hardships and limitations are also highlighted along with open questions and improvement possibilities. The general missile model is a well-known example of quasi-LPV modelling and subsequent control but the polytopic approach is not particularly suited for it unless the necessary design consideration steps are taken.

The remainder of the paper is structured as follows: the following section introduces the necessary background for LPV controller synthesis and proposes the segmented polytopic description. Section 3 presents a genetic algorithm-based optimization for such a system and Section 4 introduces the missile model in a polytopic LPV formulation. Finally, Sections 5 and 6 include validation results performed in a simulation environment and conclusive remarks respectively.

2 BACKGROUND

2.1 LPV Models and Robust Controller Synthesis

A polytopic LPV model can be generated from general LPV models with the sector nonlinearity approach (Ohtake et al., 2001), even if the parameters enter the matrices nonlinearly. Consider a polytopic LPV model in the form

$$\begin{aligned} \dot{x} &= \sum_{i=1}^{N} \mu_i(\theta)(A_i x + B_{2i} u + B_{1i} d) \\ y &= \sum_{i=1}^{N} \mu_i(\theta)(C_{2i} x) \end{aligned} \tag{1}$$

where μ_i are weighting functions belonging to the simplex

$$\Xi_\theta = \left\{ \mu(\theta) \in \mathbb{R}^n : \sum_{i=1}^{N} \mu_i(\theta) = 1 : \ \mu_i(\theta) \geq 0 \right\} \tag{2}$$

and the scheduling parameter vector θ and its derivative ρ are available in real-time and lie inside the hypercube defined by

$$\begin{aligned} \mathcal{S}_\theta &= \left\{ (\theta_1 \ldots \theta_s)^T : \theta_i \in (\theta_{min}\ \theta_{max}) \right\} \\ \mathcal{S}_\rho &= \left\{ (\rho_1 \ldots \rho_s)^T : \rho_i \in (\rho_{min}\ \rho_{max}) \right\} \end{aligned} \tag{3}$$

with $N = 2^s$.

Based on these parameter boxes, vectors can be formed by taking all possible permutations of the bounds of the parameters. These vectors will, in turn, accommodate the polytopic LPV form. If the dependence of the state space matrices in (1) on the scheduling variables θ is affine, all the possible models will be located inside the polytope of models whose vertices are the images of the vertices, ω_i, that is, the parameter vector belongs to the convex hull formed by the vertices

$$\begin{aligned} &P(\theta) \in Co \begin{pmatrix} A_i & B_i \\ C_i & D_i \end{pmatrix} := \\ &\left\{ \sum_{i=1}^{N} \mu_i(\theta) M_i : \ \mu_i(\theta) \geq 0, \ \sum_{i=1}^{N} \mu_i(\theta) = 1 \right\} \end{aligned} \tag{4}$$

Controller synthesis for an LPV system with H_∞ objective can be built on the well-known Bounded-Real Lemma (BRL) (Apkarian et al., 1995). For the closed-loop, it states that the LPV system of the form (1) has quadratic performance γ if and only if there exists a single matrix $X \succ 0$ such that

$$\begin{pmatrix} A_{cl}^T(\theta)X + XA_{cl}(\theta) & * & * \\ B_{cl}^T(\theta)X & -\gamma I & * \\ C_{cl}(\theta) & D_{cl}(\theta) & -\gamma I \end{pmatrix} \prec 0 \tag{5}$$

is feasible for any values of the parameter vector. Based on convex geometric considerations, the above theorem holds if and only if it holds at the vertices. This result holds for the situation, where the matrices are affine functions of the scheduling parameters (Apkarian et al., 1995). For the static state feedback

$$u(t) = K(\theta(t))x(t) \tag{6}$$

we need the controller, that is parameterized by the scheduling parameters

$$K(\theta(t)) = \sum_{i=1}^{N} \mu_i(\theta) K_i, \ \mu_i(t) \geq 0, \ \sum_{i=1}^{N} \mu_i(t) = 1 \tag{7}$$

where the controller corresponding to the vertex is designed with respect to a common Lyapunov function. The resulting controller can be scheduled in real-time based on (7) by the weighting functions μ.

2.2 Segmentation Strategy

Now let us consider the following scenario: the operating region is assumed to fully cover the parameter space $\mathcal{S}_\theta$ and an H_∞ LPV controller is designed for $\mathcal{P}$ with γ performance. Let us now define $\mathcal{P}_1 \subset \mathcal{P}$ and $\mathcal{P}_2 \subset \mathcal{P}$ such that $\mathcal{P}_1 \cap \mathcal{P}_2 = \emptyset$ and corresponding H_∞

controllers are designed with better ($\gamma_1 < \gamma$ and $\gamma_2 < \gamma$) performances such that $(\mathcal{P}_1 \cup \mathcal{P}_2) = \mathcal{P}$. In general, it is always assured, that whichever subregion the system resides in, corresponds to a less conservative performance level, that is, $\gamma_1 < \gamma$ and $\gamma_2 < \gamma$. For now, we do not care about the relation between γ_1 and γ_2.

Starting from (1) with a slight reformulation, let us consider i number of convex hulls and $\mathcal{P}_i$. These sets of systems lie within these polytopes

$$P_i(\theta_i) = \left(\begin{array}{c|cc} A(\theta) & B_1(\theta) & B_2(\theta) \\ \hline C_1(\theta) & D_{11}(\theta) & D_{12}(\theta) \\ C_2(\theta) & D_{21}(\theta) & D_{22}(\theta) \end{array}\right) \in \mathcal{P}_i \quad (8)$$

that is, the model sets are parameterised by i sets of scheduling variables that describe the same quantities but have different bounds.

For each of these sets of models, we require the closed-loop BRL (5) to hold, moreover, the semidefinite programming task should be performed simultaneously corresponding to a common Lyapunov matrix X. The resulting optimization problem is then formulated as

$$\min_X \gamma_i \quad (9)$$

subject to

$$X \succ 0$$

$$\begin{pmatrix} A^T_{cl,i}(\theta)X + XA_{cl,i}(\theta) & * & * \\ B^T_{cl,i}(\theta)X & -\gamma_i I & * \\ C_{cl,i}(\theta) & D_{cl,i}(\theta) & -\gamma_i I \end{pmatrix} \prec 0 \quad (10)$$

where i denotes the number of subsegments. Based on these minimization problems, dynamic or static feedback controllers can be designed and the resulting sub-controllers can be easily switched in real time according to the instantaneous state of the system.

Remark 1. *The closed-loop is quadratically stable and has H_∞ performance with γ_i performance limit for the separate subsections as long as all regions correspond to a common Lyapunov function.*

3 AUTOMATIC SEGMENTATION

Polytope segmentation and, in fact, the selection of the scheduling variables especially if there is nonlinear coupling in the system is usually a heuristic process, in many cases, leading to suboptimal choices. Therefore, to depart from this nature, a systematic procedure is proposed to refine the number of subpolytopes and their respective boundaries. This leads to an optimization task, which prescribes a maximization of overall closed-loop performance over the entire parameter range with additional penalty terms for computational complexity and operating point-dependent weighting in the loss function. Since these constraints often counteract each other, a multiobjective optimization framework is adopted in the form of genetic algorithms. The next two sections present an overview of genetic algorithms and multiobjective global optimization. The described methods are directly applicable to the internal LMI optimization loop with a user-defined loss function involving a number of configurable performance and robustness metrics, out of which a key element is the closed-loop H_∞ norm γ.

3.1 Genetic Algorithms

Genetic algorithms are based on the natural selection scheme initializing with a random population evolving through the genetic operations *selection*, *crossover* and *mutation* into a subsequent population. First, the survival of the fittest individuals is ensured by the selection process, then the crossover and mutation operations will generate the following population. The optimization is continued until the optimization objectives or any exit criteria (e.g. number of generations) are reached. For the interested reader, an overview of genetic algorithms can be found in (Kramer, 2017).

3.2 Multiobjective Optimization

Genetic algorithms are suitable options for multiobjective optimization problems described by

$$\min_{x \in C} F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix} \quad (11)$$

where n is the number of optimization objectives. Of course, especially if these objectives are conflicting, the existence of a x^* solution is not guaranteed. The usual solution to this is through the notion of *Pareto-sets* (Marler and Arora, 2004). In this paper, the weighted sum method is applied, which converts multiple objectives to a single objective via the convex combinations of the individual objectives (Do et al., 2011) resulting in

$$\min J = \sum_{i=1}^{n} \alpha_i f_i(x), \quad x \in C \quad (12)$$

where $\sum_{i=1}^{n} \alpha_i = 1$.

3.3 Optimization of LPV Segment Boundaries for H_∞ Control

Let us denote the vector of the optimization variable with η. Then, the general augmented LPV system is

$$\begin{pmatrix} \dot{x} \\ z \\ y \end{pmatrix} = \left(\begin{array}{c|cc} A(\theta,\eta) & B_1(\theta,\eta) & B_2(\theta,\eta) \\ \hline C_1(\theta,\eta) & D_{11}(\theta,\eta) & D_{12}(\theta,\eta) \\ C_2(\theta,\eta) & D_{21}(\theta,\eta) & D_{22}(\theta,\eta) \end{array}\right) \begin{pmatrix} x \\ w \\ u \end{pmatrix} \tag{13}$$

where $x \in \mathbb{R}^n$ is the state vector, $w \in \mathbb{R}^m$ is the input vector of disturbances, $u \in \mathbb{R}^{n_u}$ is the vector of control inputs, $z \in \mathbb{R}^p$ are the performance outputs and $y \in \mathbb{R}^{n_y}$ are the measured outputs.

For any vertex model, all parameter boundaries - or the extreme values of scheduling variables - are part of the optimization vector, that is, $\theta \subset \eta$. Hence, the dependence (η,θ) will be shortened to (η) Moreover, η may contain terms corresponding to the controller (tuning parameters) and meta-information regarding the number of sub-polytopes formulated by the 'closeness' of the parameter boundaries.

The dynamic LPV controller is defined as

$$\begin{pmatrix} \dot{x}_c \\ u \end{pmatrix} = \begin{pmatrix} A_c(\theta) & B_c(\theta) \\ C_c(\theta) & D_c(\theta) \end{pmatrix} \begin{pmatrix} x_c \\ y \end{pmatrix} \tag{14}$$

with $A_c \in \mathbb{R}^{n\times n}$, $B_c \in \mathbb{R}^{n\times n_y}$, $C_c \in \mathbb{R}^{n_u\times n}$ and $D_c \in \mathbb{R}^{n_u\times n_y}$.

The standard way of closed-loop controller synthesis is to start from (10) and develop controllers with a single Lyapunov matrix for all vertices. The number of vertices for a single polytope is exact but for a segmented polytope, the number of subsegments (i) needs to be defined in advance. Therefore our strategy is to declare a maximum number of segments on each dimension and set the boundaries according to the optimization outcome, which in the extremes may mean that either no segmentation takes place (all boundaries are optimized to be the same) or the maximum number of subsegments are generated.

Let us now introduce the shorthand notations with separated performance outputs $z = [z_1 \quad z_2]^T$ and disturbance inputs for reference and actual disturbance parts $w = [w_1 \quad w_2]^T$

$$\begin{aligned} \hat{A}(\eta) &:= A(\eta) - B_2(\eta)C_{12}(\eta) \\ \tilde{A}(\eta) &:= A(\eta) - B_{12}(\eta)C_2(\eta) \\ B_1(\eta) &:= [B_{11}(\eta) \quad B_{12}(\eta)] \\ C_1^T(\eta) &:= [C_{11}^T(\eta) \quad C_{12}^T(\eta)] \end{aligned}$$

where the dependence on (η) suggests that each matrix corresponds to an individual in the genetic algorithm population.

Based on (Wu et al., 1995) the general formulation of (10) is augmented with the optimization variables leading to the set of synthesis LMIs

$$\begin{gathered} \begin{pmatrix} Y\hat{A}^T + \hat{A}Y - B_2B_2^T & * & * \\ C_{11}Y & I_{ne1} & * \\ \gamma^{-1}B_1^T & 0 & -I_{nd} \end{pmatrix} \prec 0 \\ \begin{pmatrix} \tilde{A}^TX + X\tilde{A} - C_2^TC_2 & * & * \\ B_{11}^TX & I_{nd1} & * \\ \gamma^{-1}C_1 & 0 & -I_{ne} \end{pmatrix} \prec 0 \\ \begin{pmatrix} X & \gamma^{-1}I_n \\ \gamma^{-1}I_n & Y \end{pmatrix} \succeq 0 \end{gathered} \tag{15}$$

with $\hat{A} = \hat{A}(\eta)$, $\tilde{A} = \tilde{A}(\eta)$, $B_\star = B_\star(\eta)$, $C_\star = C_\star(\eta)$ and $\gamma = \gamma(i)$ dependence omitted for conciseness. This semidefinite programming task will have to be solved for all possible edge models for all individuals in the population.

Once the optimization is performed, the controller can be recovered based on (Wu et al., 1995).

3.4 Recombination and Mutation Definition

Mutation in the genetic algorithm for the nested LMI optimization task can be described by

$$\eta^{child} = \eta^{parent}\, rand[0, \frac{\eta^{parent}}{\theta_{max}}] \tag{16}$$

Recombination in the genetic algorithm for the nested LMI optimization task is performed as described in (Deep et al., 2009) by the Laplace crossover logic as

$$\begin{aligned} \eta^{child,1} &= \eta^{parent,1} + \lambda\left|(\eta^{parent,1} - \eta^{parent,2})\right| \\ \eta^{child,2} &= \eta^{parent,2} + \lambda\left|(\eta^{parent,1} - \eta^{parent,2})\right| \end{aligned} \tag{17}$$

where λ is an integer randomly chosen from a Laplace distribution.

3.5 Objective Function

The decision vector will include the optimizing variables η, the closed-loop performance γ_i for each subsegment and an added computational complexity metric (Cc). With each term, a weighting α will be associated. Moreover, a kernel function is introduced with the objective of weighting the relevant parameter subspace.

Since the scheduling parameter space might cover combinations that are physically not reachable (this can happen e.g. when there is a coupling between multiple scheduling variables) an extra weighting function is introduced to remove the importance of unreachable subregions. This is formulated via a kernel function denoted with K_g and the corresponding weights are given as the convolution of K_g and the closed-loop H_∞ norm γ by the equation

$$w(1,\dots,i) = K_g * \|\mathcal{F}_l(P(\eta),\tilde{K}(\theta))\|_{H\infty} \tag{18}$$

with the controller

$$\tilde{K}(\theta) = \sum_{i=1}^{N} \alpha_i(\theta) K_i \tag{19}$$

where $\theta = [\theta_1 \dots \theta_r]$ is the scheduling parameter vector, r is the number of scheduling parameters, $w(\eta_1,\dots,\eta_r)$ is the weight of the actual subsegment and $\mathcal{F}_l$ represents the lower linear fractional transformation operation. The optimization term corresponding to subsegment weighting then reads

$$\int_{\underline{\theta}}^{\bar{\theta}} \frac{w(i)\|\mathcal{F}_l(P(\eta),\tilde{K}(\theta))\|_{H\infty}}{r} d^r\eta \tag{20}$$

Note, that in (20) the integral is a multiple integral depending on the dimension of θ. An example result of the subsection weighting is presented in Figure 1.

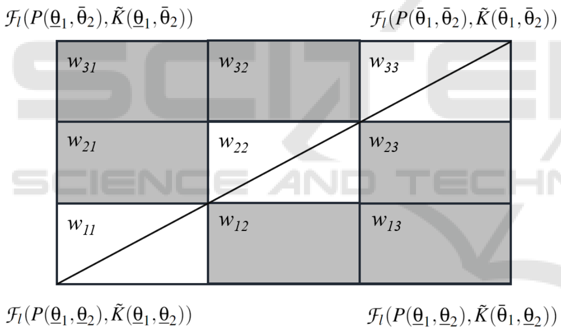

Figure 1: Edge models encompassing parameter subsections with corresponding weights represented by colours.

Moreover, computational complexity is formulated as a polynomial function of the number of subpolytopes to reflect the number of LMIs to be solved, that is,

$$Cc = r^3 \tag{21}$$

The combination of the subsegment performance terms will be predefined according to the system constraints and performance criteria. With that, the objective function reads

$$\begin{aligned} J(\eta,\gamma,Cc,w) &= \alpha_1 r^3 \\ &+ \alpha_2 \int_{\underline{\theta}}^{\bar{\theta}} \frac{w(i)\|\mathcal{F}_l(P(\eta),\tilde{K}(\theta))\|_{H\infty}}{r} d^r\eta \end{aligned} \tag{22}$$

3.6 Genetic Algorithm Program for Optimised LMI Synthesis

3.6.1 Algorithm

The program for the nested genetic algorithm - LMI optimization task is presented in the following algorithm.

Data: η,γ,Cc
Result: Optimal boundaries of polytopes
Initialize with $\eta = \eta_0$ and $\gamma = \gamma_0$ and $Cc = Cc_0$;
Set frozen w weighting to each subsegment;
while *exit criteria* **do**
    Solve synthesis LMIs (15) to acquire γ;
    Calculate computational complexity value (Cc);
    Compute the loss function $J(\eta,\gamma,Cc,w)$ for the current generation;
    Select the individuals;
    Apply crossover and mutation to generate η_{child};
end

Algorithm 1: Polytopic segmentation based on closed-loop gain optimization.

3.6.2 Computational Complexity

As per the synthesis process, for r number of scheduling variables the number of LMIs to be solved is $(3r+2)$ to make sure X and Y in (15) are positive definite. That is, for a single polytopic LPV region. Assuming equal segmentation in all dimensions, the growth can be described by $(3r+2)r^2$. This results in $O(n^3)$ which means the problem is tractable and can be solved in polynomial time.

4 PITCH-AXIS CONTROL OF A MISSILE

For the validation of the automatic segmentation method, the well-known example of a generic missile is presented based on (Shamma and Cloutiert, 1993) and (Tan et al., 2000) with the single purpose of vertical acceleration control. The main design steps and modelling ideas are taken over from (Wu et al., 1995), while the fundamental difference of using a polytopic formulation is also introduced. The notations are collected in table 1. The plant variables are tail deflection ($\delta[deg]$), angle of attack ($\alpha[deg]$), pitch rate ($q[\frac{deg}{s}]$), requested and measured normal acceleration ($A_{z,req}[g]$ and $A_z[g]$) respectively and the Mach number (M).

Table 1: Parameters of the missile model.

Notation	Value	Description
P_0	4748 $\frac{kg}{m^2}$	Static pressure at 6000 m
S	0.04 m^2	Surface area
m	204 kg	Mass
v_s	315 $\frac{m}{s}$	Speed of sound
d	0.2286 m	Diameter
I_y	0.7	Pitch-axis inertia
K_α	7	scale(P_0S/mv_s)
K_q	7	scale(P_0Sd/I_y)
K_z	3.5	scale(P_0S/m)
C_a	-0.3	Drag coefficient
a_n	0.000103f deg^{-3}	
b_n	-0.00945f deg^{-2}	
c_n	-0.1696f deg^{-1}	
d_n	-0.034f deg^{-1}	
a_m	0.000215f deg^{-3}	
b_m	-0.0195f deg^{-2}	
c_m	0.051f deg^{-1}	
d_m	-0.206f deg^{-1}	

where f is the power-dependent deg to rad conversion operator.

4.1 Quasi-LPV Model of a Missile

The nonlinear model of the missile reads

$$\begin{aligned}\dot{\alpha} &= K_\alpha M C_n(\alpha,\delta,M)cos(\alpha)+q \\ \dot{q} &= K_q M^2 C_m(\alpha,\delta,M)\end{aligned} \tag{23}$$

with the aerodynamic coefficients

$$\begin{aligned}C_n(\alpha,\delta,M) &= \alpha(a_n|\alpha|^2+b_n|\alpha|+c_n(2-\frac{M}{3})) \\ &\quad +d_n\delta \\ C_m(\alpha,\delta,M) &= \alpha(a_m|\alpha|^2+b_m|\alpha|+c_m(-7+\frac{8M}{3})) \\ &\quad +d_m\delta\end{aligned} \tag{24}$$

and the measurement equation is

$$A_z = K_z M^2 C_n(\alpha,\delta,M) \tag{25}$$

One common way to recover an LPV model from the missile is to select the scheduling variables $\theta = (\alpha, M)$. However, the scheduling variables do not enter the state space matrices linearly preventing the adoption of polytopic LPV techniques. The usual way therefore is to create a grid in the parameter space and design controllers at those specific operating points. On the other hand, polytopic LPV allows for desirable qualities, like asymptotic stability over the entire polytope region as long as the edge closed-loop models are stable (Apkarian and Gahinet, 1995). We may therefore choose the new scheduling variables as $\theta = (\alpha^2, |\alpha|)$ and use the approximation $cos(\alpha) \approx 1$ for small angles. Moreover, the dependence on M may remain frozen at a single value and the same design procedure can be performed over a grid over the feasible parameter range. In real-time, traditional gain scheduling can be adopted, leading to the quasi-LPV model

$$\begin{aligned}\begin{pmatrix}\dot{\alpha}\\ \dot{q}\end{pmatrix} &= \begin{pmatrix}K_\alpha M p_n & 1\\ K_q M^2 p_m & 0\end{pmatrix}\begin{pmatrix}\alpha\\ q\end{pmatrix} \\ &\quad + \begin{pmatrix}K_\alpha M d_n\\ K_q M^2 d_m\end{pmatrix}\delta \\ \begin{pmatrix}A_z\\ q\end{pmatrix} &= \begin{pmatrix}K_z M^2 p_n & 0\\ 0 & 1\end{pmatrix}\begin{pmatrix}\alpha\\ q\end{pmatrix} + \begin{pmatrix}K_z M^2 d_n\\ 0\end{pmatrix}\delta\end{aligned} \tag{26}$$

with the shorthands

$$\begin{aligned}p_n &:= a_n\theta_1+b_n\theta_2+c_n(2-\frac{M}{3}) \\ p_m &:= a_m\theta_1+b_m\theta_2+c_m(-7+\frac{8M}{3})\end{aligned} \tag{27}$$

Note, that for the current paper, actuator dynamics character is not considered, but a general saturation for the fin deflection is set at ± 40 deg.

Remark 2. *The new scheduling variables bear the burden of conservative parameter layout design due to the coupling between them. In other words, to hide the nonlinear character of the missile model, we have to add an extra dimension to the variable vector (Figure 1). To offset this, the proposed method helps to reduce the weight of the infeasible parameter region.*

Remark 3. *Since the state α is not available as a measured variable, we have to estimate it. For the sake of this paper, estimation was modelled with a second-order lowpass filter.*

Figure 2 presents the extended model structure with weights.

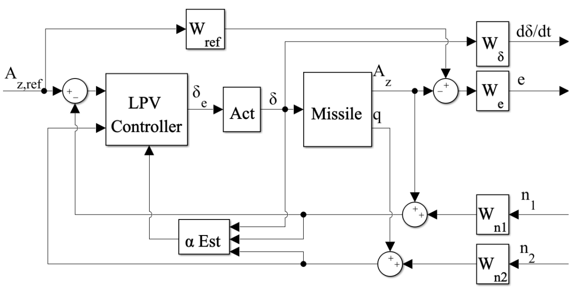

Figure 2: Augmented closed-loop control structure with synthesis weights.

There is a good case to be made for the inclusion of these weights in the optimization vector η but for demonstrative purposes, they were kept frozen at

$$\begin{aligned} W_{ref} &= \frac{200(-0.05s+1)}{0.7s^2+19.2s+200}, \quad Act = 1, \\ W_\delta &= \frac{s}{0.01s+10}, \\ W_e &= \frac{10(s+1)}{8s+0.1}, \quad W_{n1} = W_{n2} = 0.001. \end{aligned} \tag{28}$$

4.2 Optimizing Missile Polytope Parameters

Parameter boundaries were preset as $\theta_1 = [0 \quad 0.09]$ rad^2 and $\theta_2 = [0 \quad 0.3]$ rad. For ease, the maximum number of subsegments was set at 9. The optimization process was terminated after 15 generations and the optimized scheduling variable boundaries were found to be

$$\begin{aligned} \theta_1 &= [0 \quad 0.016 \quad 0.063 \quad 0.09] \\ \theta_2 &= [0 \quad 0.11 \quad 0.21 \quad 0.3] \end{aligned}$$

The objective function (22) was slightly modified for better tunability by the addition of a constant in the optimization vector η on the individual edge controllers. These can then be set optimally so that the tuning weights can be kept frozen throughout the entire process. It is to be noted though, that a more comprehensive solution would be to include actuator input and error weight transfer function terms in the vector η. Controller design based on (15) was nested in the genetic algorithm framework. The resulting LPV controllers are then selected and subsequently gain-scheduled based on the scheduling variable vector in real time.

5 SIMULATIONS

Every step of the calculations was implemented in Matlab/Simulink with a sampling time of 1 ms. The actuator effort was set to an equal average power in all cases. Vertical acceleration control is demonstrated on a step manoeuvre. For reference, a standard PID controller and a regular polytopic LPV controller are given.

Figure 3 gives a visual representation of the optimized parameter boundaries for 9 regions and the corresponding parameter trajectories.

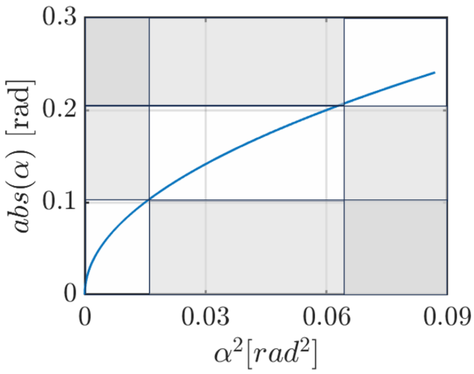

Figure 3: Optimized polytopic subsegments with actual scheduling variable trajectory. The colours denote the weighting of each subsegment (darker - less weight)).

It is easy to see that the darker areas are outside of interest with respect to the parameter trajectory. A standard LPV controller weights the overall edge controllers without consideration of the actual values of the parameter vector while the proposed method can efficiently exclude those controllers and automatically deploy the controllers more closely related to the actual operating point.

Figure 4 presents a comparison between standard LPV and the proposed approach for a typical vertical acceleration reference series.

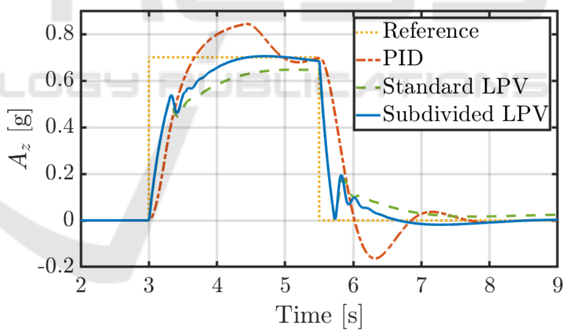

Figure 4: Comparison of controllers in a series of step input reference tracking task.

Since the PID is a linear controller and the missile is nonlinear, it is no wonder, that the PID has considerably lesser performance. The standard LPV on the other hand shows much better behaviour but the proposed partitioned controller can surpass its performance due to the reduced conservatism.

In Figure 5 the vertical acceleration in relation to the angle of attack is presented for a series of step inputs in different directions.

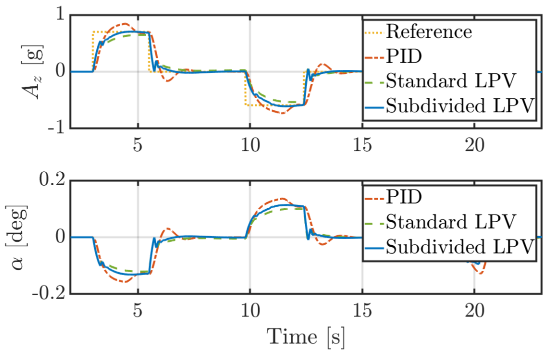

Figure 5: Vertical acceleration and corresponding angle of attack.

6 CONCLUSIONS

The proposed method demonstrates an automatic decision method for polytopic LPV parameter range subdivision. The results of the paper showed that controller performance can be significantly improved by reducing the allowable operating region. Moreover, an example was given for a missile control problem otherwise not well-suited for the polytopic control framework. As a future field of study, further investigation into the optimization process might be worthwhile and also, a deeper inspection of several nonlinear models is needed to map the possible modification requisites and potential improvement options related to individual classes of systems.

REFERENCES

Apkarian, P. and Gahinet, P. (1995). A convex characterization of gain-scheduled h∞ controllers. *Automatic Control, IEEE Transactions on Automatic Control*, 40:853 – 864.

Apkarian, P., Gahinet, P., and Becker, G. (1995). Self-scheduled h∞ control of linear parameter-varying systems: a design example. *Automatica*, 31(9):1251–1261.

Das, A. and Heiland, J. (2023). Low-order linear parameter varying approximations for nonlinear controller design for flows.

Deep, K., Singh, K. P., Kansal, M., and Mohan, C. (2009). A real coded genetic algorithm for solving integer and mixed integer optimization problems. *Applied Mathematics and Computation*, 212(2):505–518.

Do, A., Sename, O., Dugard, L., and Soualmi, B. (2011). Multi-objective optimization by genetic algorithms in h_∞/lpv control of semi-active suspension. *IFAC Proceedings Volumes*, 44(1):7162–7167. 18th IFAC World Congress.

Gahinet, P. and Apkarian, P. (1994). A linear matrix inequality approach to h∞ control. *International Journal of Robust and Nonlinear Control*, 4(4):421–448.

Gahinet, P., Apkarian, P., and Chilali, M. (1996). Affine parameter-dependent lyapunov functions and real parametric uncertainty. *IEEE Transactions on Automatic Control*, 41:436 – 442.

Huang, B., Lu, B., Li, Q., and Tong, Y. (2021). Average dwell time based smooth switching linear parameter-varying proportional-integral-derivative control for an f-16 aircraft. *IEEE Access*, 9:30979–30992.

Jiang, W., Dong, C., and Wang, Q. (2015). A systematic method of smooth switching lpv controllers design for a morphing aircraft. *Chinese Journal of Aeronautics*, 28(6):1640–1649.

Kramer, O. (2017). *Genetic Algorithm Essentials*.

Kruszewski, A., Sala, A., Guerra, T. M., and AriÑo, C. (2009). A triangulation approach to asymptotically exact conditions for fuzzy summations. *IEEE Transactions on Fuzzy Systems*, 17(5):985–994.

Kwiatkowski, A. and Werner, H. (2008). Pca-based parameter set mappings for lpv models with fewer parameters and less overbounding. *Control Systems Technology, IEEE Transactions on*, 16:781 – 788.

Lu, B. and Wu, F. (2004). Control design of switched lpv systems using multiple parameter-dependent lyapunov functions. In *Proceedings of the 2004 American Control Conference*, volume 4, pages 3875–3880 vol.4.

Marler, R. and Arora, J. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26:369–395.

Ohtake, H., Tanaka, K., and Wang, H. (2001). Fuzzy modeling via sector nonlinearity concept. In *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, volume 1, pages 127–132.

Robert, D., Sename, O., and Simon, D. (2007). A reduced polytopic lpv synthesis for a sampling varying controller: Experimentation with a t inverted pendulum. In *2007 European Control Conference (ECC)*, pages 4316–4323.

Scherer, C. (2001). Lpv control and full block multipliers. *Automatica*, 37(3):361–375.

Shamma, J. and Cloutiert, J. (1993). Gain-scheduled missile autopilot design using linear parameter varying transformation. *Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM*, 16:256–263.

Tan, W., Packard, A., and Balas, G. (2000). Quasi-lpv modeling and lpv control of a generic missile. In *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, volume 5, pages 3692–3696 vol.5.

Wu, F., Packard, A., and Balas, G. (1995). Lpv control design for pitch-axis missile autopilots. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 1, pages 188–193 vol.1.

Yan, P. and Özbay, H. (2007). On switching h_∞ controllers for a class of linear parameter varying systems. *Systems & Control Letters*, 56(7):504–511.