Alternative controller synthesize methods with $\mathcal{H}_\infty$ performance for an uncertain system based on metaheuristic optimization

Roberto M. Fuentes[1], Yamisleydi Salgueiro[2], Márcio J. Lacerda[3], and Jonathan M. Palma[4]

[1] Engineering Systems Doctoral Program, Faculty of Engineering, Universidad de Talca, Curicó, 334000, Chile (e-mail: roberto.fuentes@utalca.cl)
[2] Department of Industrial Engineering, Faculty of Engineering, Universidad de Talca, Campus Curicó, Chile (ysalgueiro@utalca.cl)
[3] Department of Electrical Engineering, Federal University of São João del-Rei, São João del-Rei, Minas Gerais, Brazil (lacerda@ufsj.edu.br)
[4] Department of Electrical Engineering, Faculty of Engineering, Universidad de Talca, Curicó, 334000, Chile (e-mail:jonathan.palma@utalca.cl)

Abstract. This work focuses on the challenging task of designing robust controllers for linear systems with time-varying parameters. It explores metaheuristic methods, specifically Particle Swarm Optimization and Genetic Algorithm, to search for controllers that optimize temporal performance and consider bounded inputs and outputs. The study demonstrates the feasibility of the approach by conducting tests on an Autonomous Vehicle Platoon system, showcasing the achieved control gains and temporal behavior, and comparing it with a technique based on LMIs.

Keywords: Autonomous Vehicle Platoon; Controller Design; Genetic Algorithm; Metaheuristic Method; Particle Swarm Optimization.

1 Introduction

Designing robust controllers for Linear Parameter-Varying (LPV) systems is a challenging task. The state-of-the-art control theory design methods for LPV systems are based on the use of the Lyapunov theory, which allows the conditions to be formulated in terms of Linear Matrix Inequalities (LMIs). The existing conditions for the synthesis of filters and controllers for LPV systems are only sufficient. There are methods employing Polya's relaxation and variable substitution to formulate positive definite optimization problems reducing the conservatism of the solutions [1–3].

However, LMI-based controller synthesis is non-trivial, both in terms of formulation and solution methods, which is why the solution is obtained offline using conventional or high-performance desktop computers, depending on the complexity and order of the system [4, 5]. Robust synthesis methods for robust controllers in LPV systems using Lyapunov theory generate guaranteed stable controllers [6]. However, there is a wide range of dynamic LPV systems based on real engineering systems where the variation and dynamics of the system are slow and predictable [7]. In systems with low uncertainty and no stochastic component, where the open-loop behavior is already stable, temporal performance becomes our primary criterion, as stability is easier to achieve. Another point of interest is the ability to manipulate and recalculate controllers locally, which is why these algorithms should be capable of implementation on Digital Signal Processors (DSP) or industrial computers with reduced computing power compared to modern desktop computers, but suitable for deployment in hostile environments.

Concerning the applications, autonomous vehicles have become very relevant in the last decades. There are even commercial services, such as Waymo by Google or RoboRide by Hyundai, both with an autonomy level of 4. In addition, some lower-level automation vehicles have assisted driving systems, such as Adaptive Cruise Control (ACC) [8–10] and Lane Keeping Assistance (LKA) [11,12], for instance. Once their use becomes widespread, it will be time to start the next step in self-driving, which is synchronizing several vehicles to operate as a whole system. In the literature, this is a concept that has been studied under different names, such as the Autonomous Vehicle Convoy (AVC) system [13,14], o Vehicle Leader-Follower (VLF) [15–17], Unmanned Surface Vehicle (USV) [18], among others. For this work, the Autonomous Vehicle Platoon (AVP) approach is considered.

One of the requirements of this kind of system is the tracking of the trajectory between vehicles. One may cite [19] where a Model Predictive Control (MPC) is implemented, mainly designed to reject external disturbance, [20] where a consensus-based control is used, focused on avoiding communication delays, and also [21] that design an adaptive strategy based on Distributed Integrated Sliding Mode (DISM) to optimize vehicle spacing. These studies work with Linear Time-Invariant (LTI) systems, except the work of [21], which offers an adaptive control to operate in an LPV environment. However, the method does not consider the propagation of errors that could be generated when operating with long rows of vehicles. The string stability in systems composed of several agents is important because if a disturbance affects the tracking, this error can be incorporated for the rest of the platoon, even reaching instability. A minimized $\mathcal{H}_\infty$ performance can avoid this phenomenon if the upper bound cost is less than 1 [22, 23]. In [24] the authors propose a synthesized method based on $\mathcal{H}_\infty$ performance. Still, this work only considers the longitudinal dynamics of the platoon under the assumption that all vehicles had the information from the leader. This aspect can generate communication problems if the lateral dynamics and numerous vehicles in the platoon are considered.

This work is interested in the synthesis of controllers for LPV systems using metaheuristics. The main goals are: i) achieving the best temporal performance for a simple stabilizing system, and ii) considering a bounded set of inputs and outputs, as it corresponds to a real-world plant process. To achieve these goals, we utilize the Particle Swarm Optimization (PSO) [25, 26] and the Genetic Algorithm (GA) [27,28] methods for designing robust controllers for a discretized version of continuous-time LPV systems. For comparison, the $\mathcal{H}_\infty$ discrete-time LMI-based synthesize method proposed in [29] is tested.

The Global Optimization Toolbox [30] in Matlab is used to solve the optimization problem, and constraints are imposed based quadratic stability of the discrete closed-loop system. We assume the system to be a simple stabilizing one, focusing solely on stability concepts for LTI systems, to present the feasibility of the proposed approach. To demonstrate the viability of our proposal, tests are conducted on an Autonomous Vehicle Platoon system, showcasing the temporal behavior and the achieved control gains.

2 Mathematical framework

Consider a discrete-time parameter-dependent system in state-space, such as

$$\mathcal{G} = \begin{cases} x(k+1) = A(\theta(k))x(k) + J(\theta(k))w(k) + B(\theta(k))u(k), \\ z(k) \quad\;\;\; = C(\theta(k))x(k) + E(\theta(k))w(k). \end{cases} \tag{1}$$

where $x(k) \in \mathbb{R}^n$ is the state vector, $u(k) \in \mathbb{R}^m$ is the input vector, and $w(k) \in \mathbb{R}^q$ is the external disturbance vector. The matrices $A(\theta(k)) \in \mathbb{R}^{n\times n}$, $B(\theta(k)) \in \mathbb{R}^{m\times n}$, $J(\theta(k)) \in \mathbb{R}^{n\times q}$, $C(\theta(k)) \in \mathbb{R}^{p\times n}$, and $E(\theta(k)) \in \mathbb{R}^{p\times q}$ are structured in the form

$$X(\theta(k)) = X_0 + \sum_{i=1}^{M} X_i \Big(\prod_{j=1}^{V} \theta_j(k)^{\lambda_j^i}\Big), \tag{2}$$

where X_j, $j = 0, \ldots, V$, are known matrices, M is the number of combinations of the time-varying parameters in the system, V is the number of time-varying parameters in the system, λ is the parameter degree in the i-th combination, and $\theta(k)$ is a vector of bounded time-varying parameters belonging to the set

$$\big\{(\theta_1, \ldots, \theta_V)(k) \in \mathbb{R}^V : \underline{\theta_j} \le \theta_j(k) \le \overline{\theta_j},\ j = 1, \ldots, V\big\}, \tag{3}$$

for all $k \ge 0$. Furthermore, the bounds $\underline{\theta_j}$ and $\overline{\theta_j}$ are known values.

2.1 Closed-loop system

If one applies a state-feedback control law $u(k) = \mathbb{K}x(k)$ to (1), the closed-loop system obtained is

$$\mathcal{G}_{\mathbb{K}} = \begin{cases} x(k+1) = A_{cl}(\theta(k))x(k) + J(\theta(k))w(k), \\ z(k) \quad\;\;\; = C(\theta(k))x(k) + E(\theta(k))w(k), \end{cases} \tag{4}$$

where, $A_{cl}(\theta(k)) = A(\theta(k)) + B(\theta(k))\mathbb{K}$ and $\mathbb{K} \in \mathbb{R}^{m\times n} = [K_1, K_2, \cdots, K_n]$ is the gains matrix.

2.2 Stability in LPV systems based on Lyapunov theory

For discrete-time LPV systems, a sufficient condition to certify stability is expressed as follows:

$$A_{cl}(\theta(k))' P A_{cl}(\theta(k)) - P \preceq 0, \tag{5}$$

where $P \in \mathbb{R}^{n\times n}$ is a parameter independent symmetric definite positive matrix. The constraint (5) is known as quadratic stability for discrete systems with time-varying parameters. These constraints into an infinite-dimensional problem that can be reduced to a finite set of tests involving parameter-dependent LMIs.

3 Performance based on guaranteed cost $\mathcal{H}_\infty$

A performance index widely employed in the control literature is the $\mathcal{H}_\infty$ guaranteed cost. This cost is, in short, the upper bound cost guaranteed by the relationship between the output $z(k)$ concerning the exogenous input $w(k)$, such that

$$\frac{||z(k)||_2^2}{||w(k)||_2^2} < \rho, \tag{6}$$

where ρ is the upper bound. If one minimizes this bound, at the same time, one minimizes the effect of the exogenous input $w(k)$ on the output $z(k)$. This parameter can also be computed by

$$\rho = \max\left(C(\theta(k)) \left(e^{j\omega T_s} I - A_{cl}(\theta(k))\right)^{-1} J(\theta(k)) + E(\theta(k)) \right) \tag{7}$$

where T_s is the sampling time of digital implementation, and the frequency must be evaluated in $\omega = [0, {}^{2\pi}/_{T_s}]$.

It is important to indicate that a lower $\mathcal{H}_\infty$ guaranteed cost, in general, implies better performance. In other words, two controllers can be compared for the guaranteed cost offered by each, where the lowest cost indicates the best controller.

4 Formulation of the Optimization Problem

A useful way to obtain a controlled system that offers a lower $\mathcal{H}_\infty$ guaranteed cost is to use an optimization method to synthesize the controller. For this, one must define the parameters, decision variables, cost function, and constraints of the problem. As for the objective function, it must be taken into account that the objective of the synthesis method is to obtain the lowest possible cost, therefore, the objective is to minimize ρ in (7).

Concerning the decision variables, from (4) it is clear that the only variables are the matrix of control gains, as a result, the optimization problem has a matrix of decision variables with elements $n \times m$, each belonging to $\mathbb{R}$.

In terms of constraints, it is important that the closed-loop system can always be stable. This is reached only if (5) is satisfied, therefore, (5) is the most important constraint. Moreover, all possible values of the elements from the decision variables vector must be restricted to bound the candidate region solution, therefore, we propose a magnitude constraint, such that

$$k_{\min} \leq k_{i,j} \leq k_{\max}, \quad \forall k_{i,j} \in \mathbb{K}. \tag{8}$$

In this way, now the space of solutions is bounded and can be denoted by $\mathcal{K} \in \mathbb{R}^{n\times m}$, being a convex space.

About the parameters, the matrices of the system (1) are fixed and known; for this reason, these matrices represent the parameters of the optimization problem.

In summary, let $A(\theta(k))$, $B(\theta(k))$, $J(\theta(k))$, $C(\theta(k))$ and $E(\theta(k))$ the problem can be expressed by[5]

$$\begin{aligned} \min_{\mathbb{K}\in\mathcal{K}} \; & \arg\max\Big(C(\theta)\left(e^{j\omega T_s}I - A_{cl}(\theta,\mathbb{K})\right)^{-1}J(\theta) + E(\theta)\Big) \\ \text{s.t. } & A_{cl}(\theta,\mathbb{K}) \text{ satisfying (5)}, \\ & k_{\min} \leq k_{i,j} \leq k_{\max}. \end{aligned} \tag{9}$$

In summary, the solver fits the decision variables until finds a minimum in the objective function that satisfies all constraints. These decision variables vector is returned by the solver as the control gains vector.

Note that (9) presents one non-linear objective function, one non-linear inequality constraint, and two bound constraints. Some optimization solvers are not compatible with inequality/equality constraints; therefore, it is necessary to create an alternative formulation.

4.1 Alternative formulation for non-constrained solvers

To give an alternative formulation of an inequality/equality constraint problem, a penalty term is considered in the objective functions. Since the inequality constraint is given to guarantee the stability of the system. When the closed-loop is not able to satisfy (5) the cost function must be increased. The new formulation proposed is given by

$$\begin{aligned} \min_{\mathbb{K}\in\mathcal{K}} \; & \arg\max\Big(C(\theta)\left(e^{j\omega T_s}I - A_{cl}(\theta,\mathbb{K})\right)^{-1}J(\theta) + E(\theta)\Big) \\ & \qquad\qquad\qquad\qquad\qquad + \sigma\Big(A_{cl}(\theta,\mathbb{K})\Big) \\ \text{s.t. } & k_{\min} \leq k_{i,j} \leq k_{\max}, \end{aligned} \tag{10}$$

with the $\sigma(\cdot)$ function given by

$$\sigma\left(A_{cl}(\theta,\mathbb{K})\right) = \begin{cases} \text{if (5) is satistied} & \text{then is } 0, \\ \text{if (5) is not satistied} & \text{then is } 100. \end{cases}$$

Thus, when the candidate controller leads to an unstable system, the candidate is severely penalized.

4.2 Optimization methods

In this work, we compare the performance of two metaheuristics: PSO and GA. Both solvers were obtained from the Global Optimization Toolbox in Matlab. It is important to note that the GA method allows handling with equality/inequality constraints, while PSO does not. These solvers were selected because the problem's objective function is non-linear and non-differentiable, similar to the constraints. In addition, when the decision variables increase, there is a combinatorial explosion of solution candidates. From this point of view, some solvers such as those based on gradient or linear methods are not recommended. The comparison criterion between both algorithms is the maximum number of Evaluations of the Objective Function (EOF) during the optimization process.

[5] In (9)-(10) the time-dependency was omitted to simplify the notation.

Parameters for solvers: To maintain the same number of EOF, the multiplication of the number of elements with the number of iterations must be constant between both solvers, which is definite as $N = 2000$. Furthermore, the maximum execution time was 20 s, the tolerance of 1×10^{-6}, the maximum stall iteration of 500, and all data are double type. The parameters used for each solver are described below: For GA, the population size and the maximum generation is proved by a hyperparameter tuning, the genetic variation is generated by `@mutationuniform` function with a ratio of 0.2; the crossover is generated by the `@crossover intermediate` function with ratio 0.8; and the elite count is 5% of population size, when the population size is greater than 20, otherwise, the elite count is 1. For PSO, inertia belongs to the interval from 0.1 to 1.1; the cognitive adjustment weight is 1, the social adjustment weight is 2; and the neighborhood for social adjustment is 0.25. In terms of the setup to solve the comparison method [29], the parsers ROLMIP [31] and YALMIP [4]; and the solver MOSEK [5] was used.

Note that Eqs. (9) and (10) refers to the general case given by (1). It is applied to an AVP cyber-physic system to validate these methods. This system is described in the next section.

5 Autonomous Vehicle Platoon Modeling

An AVP system is composed of a platoon of terrestrial vehicles that form a line, one behind the other. The first car on the line is called the leader, and the rest are followers, as shown in Fig. 1.

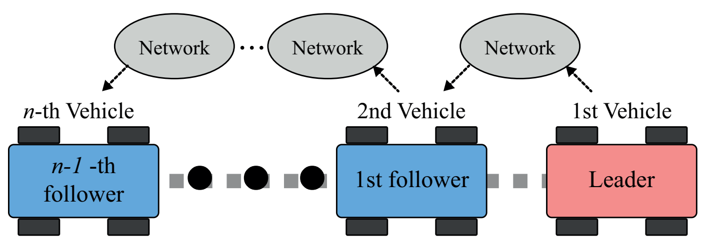

Fig. 1: Diagram of vehicle platoon with communication leader-follower.

The goal behavior is that all the elements on the line have the same trajectories along the time, for which each vehicle is proved with the states of the immediately preceding vehicle and its own states to generate a control signal, as in Fig. 2. A simplification of this case is considering each pair $(n-1)$-th and n-th as a leader-follower subsystem, thus forming an Autonomous Car-Following (ACF) problem. However, note that this behavior causes a phenomenon of error propagation, which occurs because each ACF subsystem contains the accumulated errors of previous subsystems. This phenomenon is not a problem if each subsystem has a cost $\mathcal{H}_\infty$ lower than 1.

5.1 Modelling

This system was modeled in [32], considering an LPV approach and assuming homogeneous vehicles as both the leader and the follower. The authors propose a state-spaces model from a force and

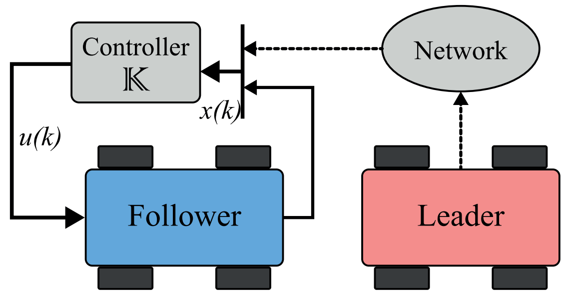

Fig. 2: Look-ahead communication topology is used for the control system of each ACF subsystem.

momentum balance applied to a single-track model given by

$$\begin{bmatrix} \dot{v}_y(t) \\ \dot{r}(t) \end{bmatrix} = \begin{bmatrix} \frac{-\left(c_r+c_f\right)}{mv_x} & \frac{\left(c_r l_r - c_f l_f\right)}{mv_x} - v_x \\ \frac{\left(l_r c_r - l_f c_f\right)}{Jv_x} & \frac{\left(l_r^2 c_r + l_f^2 c_f\right)}{Jv_x} \end{bmatrix} \begin{bmatrix} v_y(t) \\ r(t) \end{bmatrix} + \begin{bmatrix} \frac{c_f}{m} \\ \frac{l_f c_f}{J} \end{bmatrix} \left[\delta(t)\right], \tag{11}$$

where m is the mass of the vehicle, J is the yaw axis inertia, l_r and l_f are the distance between the center-of-gravity and the extremes of the axles, C_r and C_f are the cornering stiffness of the axles coefficients, F_y and F_x are the forces in the y-axis and x-axis, v_y and v_x are the velocities in their respective axes. The steering angle of the front axle is $\delta(t)$, $r(t)$ is a variation of the yaw angle, and α is the slip angle on both axles. Note that this model considers the lateral dynamic, addressing the problem with a Lane Keeping (LK) ACF system. From Fig. 1, this LK-ACF system with two homogeneous automated vehicles is proposed by

$$\begin{aligned} x(k+1) &= \begin{bmatrix} A_l(\theta(k)) & 0 \\ 0 & A_l(\theta(k)) \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ B_l(\theta(k)) \end{bmatrix} u(k) \\ &\qquad + \begin{bmatrix} B_l(\theta(k)) & 0 \\ 0 & e_d B_l(\theta(k)) \end{bmatrix} w(k), \\ z(k) &= \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ e_u \end{bmatrix} u(k). \end{aligned} \tag{12}$$

where, $A_l(\theta(k))$ and $B_l(\theta(k))$ are the polytopic matrices obtained from discretize the system in (11), considering that are parameters time-dependent in $v_x(k)$, such that $\theta_1(k) = {}^1/_{v_x(k)}$ and $\theta_2(k) = v_x(k)$. In this sense, $x(k)$ corresponds to the augmented state vector such that $x(k) = [v_{y1}(k)\ r_1(k)\ v_{y2}(k)\ r_2(k)]$, $u(k) = \delta_2(k)$ represents the control input, and $w(k) = [\delta_1(k)\ p(k)]$ is an exogenous input, where $\delta_1(k)$ is the arbitrary input of the first vehicle provided by a real driver, and $p(k)$ is a disturbance in the input angle of the second vehicle. The output $z(k)$ is the variation between both state vector cars (leader and follower), the scalar e_d is an influencing factor of disturbance on the second input, and the scalar e_u is an error adjustment constant; more information in [33]. Taking the parameter values for (11) from [34], the numeric matrices $A_l(\theta(k))$ and $B_l(\theta(k))$ result in

$$\begin{aligned} A_l(\theta(k)) = \quad & \underline{\theta_1\theta_2}\begin{bmatrix}0.9818 & -0.035\\ 0 & 0.9804\end{bmatrix} + \overline{\theta_1}\underline{\theta_2}\begin{bmatrix}0.9818 & -0.015\\ 0 & 0.9804\end{bmatrix} \\ & + \underline{\theta_1}\overline{\theta_2}\begin{bmatrix}0.9922 & -0.035\\ 0 & 0.9916\end{bmatrix} + \overline{\theta_1\theta_2}\begin{bmatrix}0.9922 & -0.015\\ 0 & 0.9916\end{bmatrix}, \\ B_l(\theta(k)) = \quad & \underline{\theta_1\theta_2}\begin{bmatrix}0.1364\\ 0.1261\end{bmatrix} + \overline{\theta_1}\underline{\theta_2}\begin{bmatrix}0.1364\\ 0.1261\end{bmatrix} \\ & + \underline{\theta_1}\overline{\theta_2}\begin{bmatrix}0.1364\\ 0.1261\end{bmatrix} + \overline{\theta_1\theta_2}\begin{bmatrix}0.1364\\ 0.1261\end{bmatrix}. \end{aligned}$$

6 Results and comparison

The synthesizing process is based on hyperparameters tuning, which consists in executing seven times both solver, where the values of elements and maximum number of iterations per solver vary, for instance, in the first execution a population/swarm size of 1000 and a maximum number of iterations of 10 was set, and for the last execution an element's size of 10 and maximum iteration of 1000 was used. Note that the maximum number of EOF corresponds to the multiplication of the number of elements and the maximum iteration admissible, which implies that these parameters are linearly dependent on each other by the expression $EOF = ES \times MNI$ where EOF is the evaluations of the objective function, ES is the element size (population or swarm) and MNI is the maximum number of iterations. The result of this hyperparameter tuning is reported in Fig. 3, where this figure shows the optimized value (from both solvers) with respect to the population size and the maximum amount of iteration. Recall that the multiplication of these two parameters can be equal to 10000, therefore, while one value increases, the other decreases to hold the EOF constants.

Since Fig. 3, the minimum optimized value was chosen from all the simulations, and the solution vectors associated with these values are used to generate the control gain vectors. The controllers obtained based on these solutions are summarized in Table 1, where the optimized vector values of the solvers are the respective vectors of control gains. Moreover, this table shows the execution time and the best function value obtained by each solver.

Table 1: Obtained Controllers

Solver	**Obtained controller** $\mathbb{K}'$	**Best Function Value**	**Time**
GA	$[-4.42\ 12.74\ 4.42\ -12.74]$	**0.1563**	34.12 s
PSO	$[-35.17\ 46.65\ 35.09\ -46.73]$	**0.1651**	16.79 s
LMI[3]	$[-2.27\ 10.34\ 2.27\ -10.34]$	**1.4510**	0.44 s

[3]**Synthesize by the method in [29]**

To compare the solutions obtained by the solvers, the fitness value column of Table 1 directly indicates the minimized value, where a minimum value is better. In this sense, the GA obtains

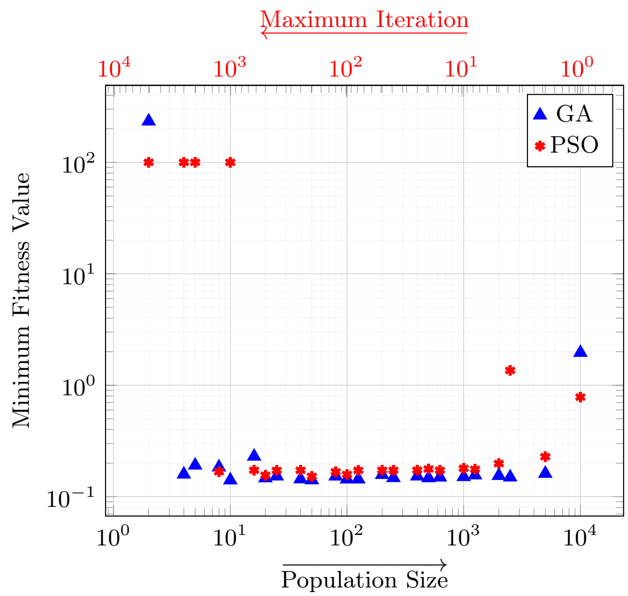

Fig. 3: Results for tuning of population size versus allowed maximum iterations. Note that each simulation holds constant the number of objective function evaluations.

the minimum $\mathcal{H}_\infty$ cost; however, it requires 34.12 s running. On the other hand, the solution from the LMI-based solver was obtained in 0.44 s, being the fastest solver, although its guaranteed cost $\mathcal{H}_\infty$ value is higher than the fitness value obtained by the GA solver. But the proposal of this work is to prove an off-line controller synthesizes method, therefore, besides the $\mathcal{H}_\infty$ index, the temporal performance of the controller is important and must be tested. For this, the usefulness of the synthesized controllers is identified based on the performance index $\mathcal{J}$ given by (13), which corresponds to the integrated square output.

$$\mathcal{J} = \frac{1}{T}\int_0^T z'(\tau)z(\tau)\mathrm{d}\tau \tag{13}$$

This index penalizes the most deviating values from the origin. For testing, the disturbance signal (14) is used. This vector is composed of two oscillatory decreasing signals 90 degrees out of phase with each other, both of which have the same frequency and decay rates.

$$w(t) = \begin{bmatrix}\delta_1(t)\\ p(t)\end{bmatrix} = \begin{bmatrix}0.1e^{-0.02t}cos(0.5t)\\ 0.1e^{-0.02t}sin(0.5t)\end{bmatrix}. \tag{14}$$

This disturbance generates some behavior in the $v_{y1}(k)$ and $r_1(k)$ states, which must follow the $v_{y2}(k)$ and $r_2(k)$ states. Note that the $z(k)$ output corresponds to $v_{y1}(k) - v_{y2}(k)$ and $r_1(k) - r_2(k)$, that is, hold $z(k) \to 0$ is the performance goal. This behavior is shown in the Fig. 4, which is obtained based on a 200 s simulation. As validation, the $\rho_{\mathcal{K}}$ is measured by the cost

$$\rho_{\mathcal{K}} = \max {}^{z(k)'z(k)}/_{w(k)'w(k)}, \tag{15}$$

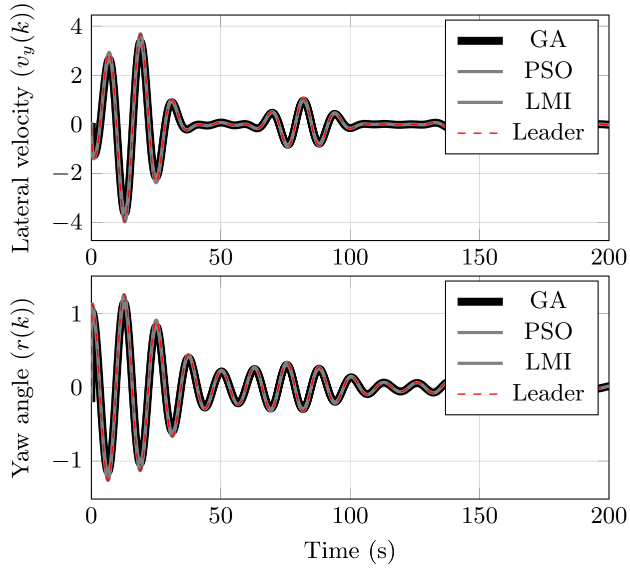

Fig. 4: Temporal response of the controlled system. The upper figure shows the $v_{y1}(k)$ signal and the $v_{y2}(k)$ signals obtained by each controller of each solver. The lower figure shows the $r_1(k)$ signal and the $r_2(k)$ signals.

where for GA solver, a $\rho_{GA} = 0.1386$ was obtained; for PSO solver, a $\rho_{PSO} = 0.0590$ was obtained; and for LMI-based method, a $\rho_{LMI} = 0.2736$ was obtained. This shows that this particular system formulation is correct, due to the OF values from the solvers are major than the temporal value for this case. By visual inspection, all controlled systems are stable and $z(k)$ converge to zero. However, the cost index $\mathcal{J}$ allows a numerically precise comparison to be made. The cost obtained by each controller is 0.0020 for the GA solver, 0.0038 for the PSO solver, and 0.0031 for the LMI-based method, where a lower cost index $\mathcal{J}$ implies better performance, therefore, GA solver obtained the best cost index $\mathcal{J}$ even over the LMI-based method.

7 Conclusion

An alternative synthesizing process based on minimizing cost $\mathcal{H}_\infty$ using metaheuristic methods was tested in an AVP system, obtaining $\mathcal{H}_\infty$ indices of 0.156-0.165 for the GA and PSO optimization methods, respectively. For this, two optimization problem statements were proposed, one with a non-linear inequality/equality constraint, without penalty term, and the other without constraint, but with penalty term instead. Regarding the results, the GA method obtains a lower optimized value of the objective function (an 5.33% lower with respect to the PSO method) and better temporal performance (47% improvement), but the time required to synthesize was almost twice as long as for the PSO method. When these solvers are compared with the LMI-based method, the time required to find the solution is almost two magnitude orders higher, but as the aim is goal an offline controller, the temporal performance is the best criterion.

Note that the assumption of stability open-loop and the validation only in the vertices in the studied LPV system limits the proposed synthesizing method, therefore, future works may include a method of theoretical validation, the use of LMI as a constraint of stability, and the cost $\mathcal{H}_\infty$ at the same time. Applying this validation allows us to guarantee theoretical performance, even if metaheuristic methods alone cannot ensure it.

Acknowledgment

The founded by the Chilean National Agency for Research and Development (ANID) scholarship ANID-Doctorado finance code N° 21232329 and scholarship *Guillermo Blanco*, University of Talca.The project ANID-Fondecyt Iniciación finance code 11201049 and ANID-Regular 1220903. Project Stic-Amsud AMSUD220013, Stic-Amsud/CAPES NetConHybSDP, code 22-STIC-09. Thematic Network $723RT0150$ "*Red para la integración a gran escala de energías renovables en sistemas eléctricos (RIBIERSE-CYTED)*" financed by the call for Thematic Networks of the CYTED (Ibero-American Program of Science and Technology for Development).

References

1. R. Kamyar and M. M. Peet, "Decentralized computation for robust stability analysis of large state-space systems using polya's theorem," in *2012 American Control Conference (ACC)*, 2012, pp. 5948–5954.
2. A. Sala and C. Ariño, "Asymptotically necessary and sufficient conditions for stability and performance in fuzzy control: Applications of polya's theorem," *Fuzzy Sets and Systems*, vol. 158, 2007.
3. T. Zou and H. Yu, "Asymptotically necessary and sufficient stability conditions for discrete-time takagi-sugeno model: Extended applications of polya's theorem and homogeneous polynomials," *Journal of the Franklin Institute*, vol. 351, 2014.
4. J. Lofberg, "Yalmip : a toolbox for modeling and optimization in matlab," in *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, 2004, pp. 284–289.
5. M. Aps, "Mosek optimization toolbox for matlab," *User's Guide and Reference Manual*, p. 327, 2021.
6. A. Gupta, M. Mejari, P. Falcone, and D. Piga, "Computation of parameter dependent robust invariant sets for lpv models with guaranteed performance," *Automatica*, vol. 151, p. 110920, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109823000705
7. P. M. Oliveira, J. M. Palma, E. G. Nepomuceno, and M. J. Lacerda, "Reinforcement learning for control design of uncertain polytopic systems," *Information Sciences*, vol. 625, pp. 417–429, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025523000439
8. M. Althoff, S. Maierhofer, and C. Pek, "Provably-correct and comfortable adaptive cruise control," *IEEE TRANSACTIONS ON INTELLIGENT VEHICLES*, vol. 6, no. 1, pp. 159–174, MAR 2021.
9. C. Bian, G. Yin, L. Xu, and N. Zhang, "Bidirectional adaptive cruise control for intelligent vehicles," *INTERNATIONAL JOURNAL OF HEAVY VEHICLE SYSTEMS*, vol. 28, no. 4, pp. 467–486, 2021.
10. I. Karafyllis, D. Theodosis, and M. Papageorgiou, "Nonlinear adaptive cruise control of vehicular platoons," *INTERNATIONAL JOURNAL OF CONTROL*, vol. 96, no. 1, pp. 147–169, JAN 2 2023.
11. M. Liu, X. Deng, Z. Lei, C. Jiang, and C. Piao, "Autonomous lane keeping system: Lane detection, tracking and control on embedded system," *JOURNAL OF ELECTRICAL ENGINEERING & TECHNOLOGY*, vol. 16, no. 1, pp. 569–578, JAN 2021.
12. Y. Bian, J. Ding, M. Hu, Q. Xu, J. Wang, and K. Li, "An advanced lane-keeping assistance system with switchable assistance modes," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, vol. 21, no. 1, pp. 385–396, JAN 2020.

13. C. Cheung, S. Rawashdeh, and A. Mohammadi, "Jam mitigation for autonomous convoys via behavior-based robotics," *Applied Sciences*, vol. 12, no. 19, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/19/9863
14. H. B. Jond and J. Platoš, "Differential game-based optimal control of autonomous vehicle convoy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 2903–2919, 2023.
15. N. Wang and H. Li, "Leader–follower formation control of surface vehicles: A fixed-time control approach," *ISA Transactions*, vol. 124, pp. 356–364, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019057820302305
16. J. Schulte, M. Kocherovsky, N. Paul, M. Pleune, and C.-J. Chung, "Autonomous human-vehicle leader-follower control using deep-learning-driven gesture recognition," *Vehicles*, vol. 4, 2022.
17. C. L. Chen, D. Yu, and L. Liu, "Automatic leader-follower persistent formation control for autonomous surface vehicles," *IEEE Access*, vol. 7, 2019.
18. X. Sun, G. Wang, Y. Fan, D. Mu, and B. Qiu, "A formation collision avoidance system for unmanned surface vehicles with leader-follower structure," *IEEE Access*, vol. 7, 2019.
19. L. Zhou, G. Wang, K. Sun, and X. Li, "Trajectory tracking study of track vehicles based on model predictive control," *Strojniski Vestnik/Journal of Mechanical Engineering*, vol. 65, 2019.
20. J. Chen, H. Liang, J. Li, and Z. Lv, "Connected automated vehicle platoon control with input saturation and variable time headway strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4929–4940, 2021.
21. L. Zuo, D. Meng, and J. Zhang, "Optimal spacing policy for vehicle platoon control with road-friction coefficient," *Journal of Advanced Transportation*, vol. 2021, 2021.
22. C.-Y. LIANG and H. PENG, "String stability analysis of adaptive cruise controlled vehicles," *JSME International Journal Series C*, vol. 43, no. 3, pp. 671–677, 2000.
23. N. Bekiaris-Liberis, "Robust string stability and safety of cth predictor-feedback cacc," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8209–8221, 2023.
24. K. Halder, U. Montanaro, S. Dixit, M. Dianati, A. Mouzakitis, and S. Fallah, "Distributed h∞ controller design and robustness analysis for vehicle platooning under random packet drop," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4373–4386, 2022.
25. R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.
26. A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," *Archives of Computational Methods in Engineering*, vol. 29, 2022.
27. M. O. Okwu and L. K. Tartibu, "Genetic algorithm toolbox for use with matlab," *Studies in Computational Intelligence*, vol. 927, 2021.
28. M. Kumar, M. Husain, N. Upreti, and D. Gupta, "Genetic algorithm: Review and application," *SSRN Electronic Journal*, 2020.
29. J. De Caigny, J. F. Camino, R. C. Oliveira, P. L. D. Peres, and J. Swevers, "Gain-scheduled $\mathcal{H}2$ and $\mathcal{H}\infty$ control of discrete-time polytopic time-varying systems," *IET control theory & applications*, vol. 4, no. 3, pp. 362–380, 2010.
30. "Matlab global optimization toolbox," 2023, the MathWorks, Natick, MA, USA.
31. C. M. Agulhari, A. Felipe, R. C. L. F. Oliveira, and P. L. D. Peres, "Algorithm 998: The Robust LMI Parser — A toolbox to construct LMI conditions for uncertain systems," *ACM Transactions on Mathematical Software*, vol. 45, no. 3, p. 36:1–36:25, August 2019.
32. R. Fuentes, P. M. Oliveira, L. P. Carvalho, M. J. Lacerda, and J. M. Palma, "Autonomous vehicle platoon packet based control problem under denial-of-service attacks," in *CONTROLO 2022*, L. Brito Palma, R. Neves-Silva, and L. Gomes, Eds. Cham: Springer International Publishing, 2022, pp. 474–486.
33. L. P. Carvalho, J. M. P. Olate, A. P. de Castro Goncalves, and C. Duran-Faundez, "Vehicle following problem: A control approach for uncertain systems with lossy networks," *IEEE Latin America Transactions*, vol. 16, no. 9, pp. 2392–2399, 2018.

34. R. M. Fuentes, J. M. Palma, H. G. Júnior, M. J. Lacerda, L. d. P. Carvalho, A. J. Rojas, and R. C. L. F. Oliveira, "Gain-scheduled control design applied to classical dc–dc converters in photovoltaic systems and constant power loads," *Mathematics*, vol. 10, no. 19, 2022. [Online]. Available: https://www.mdpi.com/2227-7390/10/19/3467