Finite Horizon Worst Case Analysis of Linear Time-Varying Systems Applied to Launch Vehicle

Felix Biertümpfel, Nantiwat Pholdee, Samir Bennani, and Harald Pfifer

Abstract—This article presents an efficient approach to compute the worst case gain of the interconnection of a finite time horizon linear time-varying system and a perturbation. The input/output behavior of the uncertainty is described by integral quadratic constraints (IQCs). A condition for the worst case gain of such an interconnection can be formulated using dissipation theory as a parameterized Riccati differential equation, which depends on the chosen IQC multiplier. A nonlinear optimization problem is formulated to minimize the upper bound of the worst case gains over a set of admissible IQC multipliers. This problem can be efficiently solved with a custom-tailored logarithmically scaled, adaptive differential evolution algorithm. It provides a fast alternative to similar approaches based on solving semidefinite programs. The algorithm is applied to the worst case aerodynamic load analysis of an expendable launch vehicle (ELV). The worst case load of the uncertain ELV is calculated under wind turbulence during the atmospheric ascend and compared to results from nonlinear simulation.

Index Terms—Flight control, integral quadratic constraints (IQCs), metaheuristics, robust control, time-varying systems.

I. INTRODUCTION

Numerous systems follow a predefined trajectory with a given start and terminal point during their nominal operation. Examples are various and include robots [1], terminal guidance systems [2], aircraft on final approach, and space applications, such as launch vehicles [3] or vehicles for atmospheric reentry [4]. The dynamics of these examples can all be represented as time-varying systems. Most of these systems can be described by a linearization along a given operating trajectory. Consequently, they are linear time-varying (LTV) systems, i.e., linear systems whose state matrices depend solely on time. Finite horizon LTV systems are formally introduced in Section III-A. Closely related to these types of systems is another class of LTV systems, namely, periodic ones. Systems coverable by the periodic form are, for example, the flapping of a helicopter rotor blade in forward flight [5], wind turbines [6], and spinning satellites [7].

In the literature, numerous approaches to calculate the robustness of uncertain periodic LTV systems are given [8], [9], [10]. In contrast, the analysis of finite horizon LTV systems is significantly less exploited. The work in [11] proposes robustness measures for finite time trajectories using integral quadratic constraints (IQCs) to represent the uncertainties. However, [11] does not consider external disturbances. The analysis via gap metrics is covered in [12] or [13]. This article presents a worst case analysis for uncertain LTV systems. It is based on the extension of a finite horizon formulation of the bounded real lemma (BRL) given in [14] for nominal LTV systems under external disturbance. The system’s uncertainties are represented by IQCs. First, introduced by Megretski and Rantzer [15], IQCs present a general framework for robustness analyses. They are able to cover numerous types of perturbations, such as uncertainties, hard nonlinearities, or infinite-dimensional systems. In [15], a broad range of multipliers defined in the frequency domain is given. Due to the frequency domain interpretation in [15], only nominal systems that are linear time-invariant can be covered.

Frequency domain IQCs have been successfully applied for various aerospace applications [16], [17]. More recently, the IQC framework has been studied in the time domain, such as [18], [19], and [20]. This time-domain formulation of IQCs is given in Section III-B. The time-domain approach allowed the extension to cover linear parameter varying [21], [22] or nonlinear polynomial systems [23] under perturbations. They provide sufficient conditions on worst case input/output gains based on a BRL-like condition. These advances opened the IQC framework for discrete [24], [25] and continuous LTV systems [26]. The analysis conditions are based on the dissipation inequality conditions presented in [21] for uncertain LPV systems. This LTV robustness analysis framework is described in Section III-C. In [26], the analysis condition is only enforced on a finite-dimensional grid to overcome the problem’s infinite nature. In [27], the linear matrix inequality (LMI)-based approach in [26] is extended using an equivalent...
RDE formulation of the LMI conditions. The LMI and RDE are solved iteratively to mitigate the effect of the gridding and calculate a less conservative upper bound on the worst case gain. This approach was successfully demonstrated for academic examples over short time horizons. In [28], an alternative approach exclusively using the RDE formulation was proposed. This leads to a nonlinear optimization problem centered around solving a parameterized RDE and optimizing over the IQC parameterization. It was successfully applied to a simple launcher ascent problem in [29]. However, the approach was limited to a single uncertainty.

This article extends the work in [28] to complex industrial examples with highly time-varying dynamics over long time horizons (>10 s). Therefore, a tailored algorithm is developed using logarithmically scaled adaptive differential evolution with linear population size reduction (Log-L-SHADE). The algorithm’s search strategy is based on the original L-SHADE [30], but the logarithmic scale significantly reduces the search space. Furthermore, the algorithm is extensively updated to exploit the structure of the optimization problem and the IQC framework. A detailed description of the novel nonlinear program is given in Section IV.

The algorithm’s applicability to industry-relevant examples is demonstrated in a detailed analysis presented in Section V. There, the worst case aerodynamic load on an expendable launch vehicle (ELV) during its ascent under atmospheric turbulence is calculated. The analysis utilizes an advanced turbulence disturbance model adjusted to the finite horizon LTV framework’s requirements imposed by the BRL. Furthermore, an adequate uncertainty set in the IQC framework is introduced. To finish, a Monte Carlo simulation on the corresponding nonlinear simulation over the allowable uncertainty set and turbulence profiles is conducted. It is used to validate that the finite horizon LTV analysis using IQCs provides a valid and not too conservative upper bound for the nonlinear model in a fraction of time.

Thus, this article contributes an efficient algorithm specialized to calculate the upper bounds of the worst case gain of uncertain finite horizon LTV systems over large time horizons. It is based on the solvability of a parameterized RDE and nonlinear optimization avoiding the solution of SDPs and the inherent gridding problem. A specifically tailored metaheuristic exploits the optimization problem’s structure for computational efficiency. The algorithm’s feasibility is demonstrated on a high-fidelity example in the form of a worst case loads analysis of a space launcher during atmospheric ascent.

II. NOTATION

In the course of this article, \( \mathbb{R} \) and \( \mathbb{C} \) denote the set of real and complex numbers, respectively. The set of rational functions with real coefficients that are proper and have no poles on the imaginary axis is denoted by \( \mathbb{R}_{\mathbb{L}} \). Herein, \( \mathbb{R}_{\mathbb{L}} \) is the subset of functions in \( \mathbb{R}_{\mathbb{L}} \), which are analytical in the closed right half of the complex plane. The sets of \( m \times n \) matrices whose elements are in \( \mathbb{C} \), \( \mathbb{R}_{\mathbb{L}} \), and \( \mathbb{R}_{\mathbb{L}} \) are denoted \( \mathbb{C}_{m \times n} \), \( \mathbb{R}_{\mathbb{L}}_{m \times n} \), and \( \mathbb{R}_{\mathbb{L}}_{m \times n} \), respectively. Vectors are described by a single superscript, e.g., \( \mathbb{R}^n \) being the set of vectors whose elements are in \( \mathbb{R} \). A vertical concatenation of the vectors \( x \in \mathbb{R}^n \) and \( y \in \mathbb{R}^m \) is denoted by \( [x; y] \in \mathbb{R}^{n+m} \). The set of \( n \times n \) symmetrical matrices are denoted by \( \mathbb{S}^n \). \( \mathbb{R}_0^+ \) denotes the set of positive real numbers, including zero. The set of positive real numbers excluding zero is denoted by \( \mathbb{R}_+ \). For \( z \in \mathbb{C} \), \( \bar{z} \) denotes the complex conjugate of \( z \). The transpose of a matrix \( M \in \mathbb{C}_{m \times n} \) is denoted by \( M^T \).

Furthermore, structured matrices \( C \) are expressed concisely via the Kronecker product \( C = A \otimes B = (a_{ij} \cdot B) \). The size of signals \( v : \mathbb{R}_0^+ \to \mathbb{R}^n \) in this article is described by the Lebesgue 2-norm, which can be defined as [14]

\[
\|v\|_2^{[0,T]} = \left[ \int_0^T v(t)^T v(t) \, dt \right]^{\frac{1}{2}}.
\]

III. BACKGROUND ON THE ROBUSTNESS ANALYSIS OF FINITE HORIZON LTV SYSTEMS

A. Finite Horizon LTV Systems

An uncertain system is defined by the feedback interconnection of a nominal LTV system \( G \) and the perturbation \( \Delta \), as shown in Fig. 1. This interconnection is denoted by \( F_v(G, \Delta) \), with the LTV system \( G \) defined as

\[
\dot{x}_G(t) = A_G(t)x_G(t) + B_G(t)\begin{bmatrix} w(t) \\ d(t) \end{bmatrix} \\
\begin{bmatrix} v(t) \\ e(t) \end{bmatrix} = C_G(t)x_G(t) + D_G(t)\begin{bmatrix} w(t) \\ d(t) \end{bmatrix}
\]

where \( x_G(t) \in \mathbb{R}^{n_G} \), \( d(t) \in \mathbb{R}^{n_d} \), and \( e(t) \in \mathbb{R}^{n_e} \) denote the state, input, and output vectors, respectively. The matrices \( A_G \), \( B_G \), \( C_G \), and \( D_G \) are piecewise continuous locally bounded matrix-valued functions of time with the appropriate dimensions. To shorten the notation, the explicit time dependence is generally omitted in this article. The uncertainty \( \Delta : L_2^w[0, T] \to L_2^w[0, T] \) is a bounded and causal operator with input \( v \in \mathbb{R}^{n_v} \) and output \( w \in \mathbb{R}^{n_n} \). \( \Delta \) can describe hard nonlinearities, such as saturations, infinite-dimensional operators, such as time delays, and dynamic and real parametric uncertainties.

B. Integral Quadratic Constraints

The input/output behavior of the perturbations is bounded using IQCs. IQCs were first introduced in the context of robustness analysis in the frequency domain by [15]. More recently, the IQC framework has been investigated in the time domain [18], [20], [21]. Time-domain IQCs are generally distinguished into hard IQCs, which must hold over all finite time.
horizons in an infinite horizon analysis, and soft IQCs, which only hold over infinite horizons [15]. Hard factorizations are used in, e.g., [21], to assess the stability of LPV systems. The LTV analysis presented here only requires the IQC to hold over one specific finite analysis horizon [0, T].

A time-domain IQC is defined by the filter \( \Psi \in \mathbb{R}^{n_g \times (n + n_u)} \) and a symmetric matrix \( M \in \mathbb{S}^n \). The short notation \( \Delta \in \text{IQC}(\Psi, M) \) is used if the perturbation \( \Delta \) satisfies the IQC defined by \( \Psi \) and \( M \) over the interval \([0, T]\). This is the case if the output \( z \) of the filter \( \Psi \) with inputs \( v \) and \( w \) fulfills the quadratic time constraint

\[
\int_0^T z(t)^T M z(t) \, dt \geq 0 \tag{3}
\]

for all \( v \in L_2[0, T] \), \( w = \Delta(v) \), and zero initial conditions \( x_\Psi(0) = 0 \) given a finite interval \([0, T]\). The variable \( x_\Psi \in \mathbb{R}^{n_u} \) denotes the IQC filter’s state vector. The IQC framework allows to include \( k \) different perturbations \( \Delta_i \in \text{IQC}(\Psi_i, M_i) \) in a single IQC by diagonally combining them.

### C. LTV Robustness Analysis

The feedback interconnection of an LTV system \( G \) and uncertainty block \( \Delta \) in the IQC framework is pictured in Fig. 2. The input \( v \) and output \( w = \Delta(v) \) of \( \Delta \) are connected to the IQC filter \( \Psi \). Consequently, \( \Delta \) is excluded from the interconnection, as emphasized in Fig. 2. Accordingly, \( w \) is now treated as an external signal in the extended state space system \( H \)

\[
\dot{x}(t) = A(t) x(t) + [B_1(t) B_2(t)] [w(t) \, d(t)] \\
\begin{bmatrix} z(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} C_1(t) \\ C_2(t) \end{bmatrix} x(t) + [D_11(t) D_12(t) \, D_21(t) D_22(t)] [w(t) \, d(t)] \tag{4}
\]

where \( x(t) = [x_G(t)^T, x_\Psi(t)^T]^T \in \mathbb{R}^{n_G + n_u} \) represents the state vector containing the states of \( G \) and \( \Psi \), \( d(t) \in \mathbb{R}^n \) represents the external disturbance input vector, and \( e(t) \in \mathbb{R}^{n_u} \) represents the performance output vector. The internal signal \( v \) and the external signal \( w \) are subject to the time-domain constraint (3) enforced on the output of the IQC filter \( z \). Hence, the explicit formulation of \( w = \Delta(v) \) is replaced by the time-domain inequality (3). The interconnection \( F_u(G, \Delta) \) in Fig. 2 is said to be well-posed if, for all initial conditions, \( x_G \) and \( d \in L_2[0, T] \) unique solutions \( x_G \in L_2[0, T], v \in L_2[0, T], \) and \( w \in L_2[0, T] \) satisfying (4) and causally dependent on \( d \) exist.

The robust performance of an uncertain LTV system in the IQC framework can then be quantified by worst case finite horizon input/output gains. Specifically, two metrics are considered in this article. First, the finite horizon worst case \( L_2[0, T] \) to \( \|e(T)\|_2 \) gain defined as follows:

\[
\|F_u(G, \Delta)\|_2 := \sup_{\Delta \in \text{IQC}(\Psi, M)} \sup_{d \in L_2(0, T)} \sup_{d \neq 0, x_\Psi(0) = 0} \frac{\|e(T)\|_2}{\|d(t)\|_2} \tag{5}
\]

Geometrically interpreted, it describes the ball upper bounding the worst case output \( e(T) \) over all \( \Delta \in \text{IQC}(\Psi, M) \) for \( \|d(t)\|_{2[0,T]} = 1 \) and the considered finite time horizon \([0, T]\) with \( T \in [0, \infty) \). The second performance measure is the finite horizon worst case \( L_2[0, T] \)

\[
\|F_u(G, \Delta)\|_{2[0,T]} := \sup_{\Delta \in \text{IQC}(\Psi, M)} \sup_{d \in L_2(0, T)} \sup_{d \neq 0, x_\Psi(0) = 0} \frac{\|e(T)\|_{2[0,T]}}{\|d(t)\|_{2[0,T]}} \tag{6}
\]

It defines an upper bound on the worst case amplification of the system over all \( \Delta \in \text{IQC}(\Psi, M) \) for inputs \( d(t) \in L_2[0, T] \) and the respective finite time horizon \([0, T]\) with \( T \in [0, \infty) \).

### D. Bounded Real Lemma for LTV Systems Including IQCs

A dissipation inequality using the extended system \( H(4) \) and the finite time horizon IQC (3) is formulated to bound either the worst case gain in (5) or (6) of the interconnection \( F_u(G, \Delta) \) (see [27] and [28] for details). The respective dissipation inequality can be expressed as an equivalent RDE leading to the following two theorems.

**Theorem 1:** Let \( F_u(G, \Delta) \) be well-posed \( \forall \Delta \in \text{IQC}(\Psi, M) \); then, \( \|F_u(G, \Delta)\|_2 < \gamma \) if there exist a continuously differentiable \( P : \mathbb{R}_0^+ \rightarrow \mathbb{S}^n \) such that

\[
P(T) = \frac{1}{\gamma} C_2(T)^T C_2(T) \tag{7}
\]

\[
\dot{P} = Q + P \tilde{A} + \tilde{A}^T P - P SP \quad \forall t \in [0, T] \tag{8}
\]

and

\[
R = \begin{bmatrix} D_{11}(t) M D_{11}^T & D_{12}(t) M D_{12}^T \\ D_{11}^T M D_{11} & D_{12}^T M D_{12} - \gamma I_{n_2} \end{bmatrix} < 0 \tag{9}
\]

with

\[
\tilde{A} = [B_1 B_2] R^{-1} \begin{bmatrix} (C_1^T M D_{11})^T \\ (C_1^T M D_{12})^T \end{bmatrix} - A \tag{10}
\]

\[
S = -[B_1 B_2] R^{-1} \begin{bmatrix} B_1^T \\ B_2^T \end{bmatrix} \tag{11}
\]

\[
Q = -C_1^T M C_1 + \begin{bmatrix} (C_1^T M D_{11})^T \\ (C_1^T M D_{12})^T \end{bmatrix} R^{-1} \begin{bmatrix} (C_1^T M D_{11})^T \\ (C_1^T M D_{12})^T \end{bmatrix} \tag{12}
\]

**Proof:** The proof is only sketched, and a detailed version is given in [27]. It is based on the definition of a time-dependent quadratic storage function \( V : \mathbb{R}^{n_x} \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+ \). After perturbing (8), the resulting Riccati inequality can be rewritten as an LMI applying the Schur complement. The equivalence is guaranteed by condition (9), which also ensures the invertibility of \( R \). Multiplying \( [x^T, w^T, d^T]^T \) and \( [x^T, w^T, d^T]^T \) on the left- and right-hand sides, respectively, of the LMI results...
in a dissipation inequality. Integration from 0 to $T$ for zero initial conditions gives

$$ x(T)^T P(T) x(T) - \gamma \int_0^T d(t)^T d(t) dt + \int_0^T z(t)^T M z(t) dt \leq 0 $$

(13)

where the last term can be neglected according to (3). Equality (7) is perturbed and left and right multiplied with $x(T)^T$ and $x(T)$, respectively, in

$$ x(T)^T P(T) x(T) - \frac{1}{\gamma} x(T)^T C_2(T)^T C_2(T) x(T) $$

$$ = x(T)^T P(T) x(T) - \frac{1}{\gamma} e(T)^T e(T) \geq 0. $$

(14)

After applying $\Delta \in \text{IQC}(\Psi, M)$, (13) is substituted in (14). Subsequently, the vector 2-norm (Euclidean vector norm) $\|e(T)\|^2_2 = e(T)^T e(T)$ is used to conclude that the upper bound on (5) is given by $\gamma$.

Proof: Let $F_\nu(G, \Delta)$ be well-posed $\forall \Delta \in \text{IQC}(\Psi, M)$; then, $\|F_\nu(G, \Delta)\|_{20, T} < \gamma$ if there exist a continuously differentiable $P : \mathbb{R}^n_+ \to \mathbb{S}^n$ such that

$$ P(T) = 0 $$

$$ \dot{P} = \dot{Q} + P \hat{A} + \hat{A}^T P - P \hat{S} P $$

(15)

and

$$ \hat{R} = \begin{bmatrix} D_{11}^{T} M D_{11} + D_{11}^{T} D_{21} & D_{12}^{T} M D_{12} + D_{12}^{T} D_{22} \\ D_{12}^{T} M D_{12} + D_{12}^{T} D_{22} & D_{22}^{T} M D_{22} + D_{22}^{T} D_{22} - \gamma^2 I_{n_x} \end{bmatrix} < 0 $$

(17)

with

$$ \hat{A} = \begin{bmatrix} B_1 & B_2 \end{bmatrix} \hat{R}^{-1} \begin{bmatrix} (C_1^{T} M D_{11} + C_1^{T} D_{21})^T \\ (C_1^{T} M D_{12} + C_1^{T} D_{22})^T \end{bmatrix} - A $$

(18)

$$ \dot{S} = -\begin{bmatrix} B_1 & B_2 \end{bmatrix} \hat{R}^{-1} \begin{bmatrix} B_1^T \\ B_2^T \end{bmatrix} $$

(19)

$$ \dot{Q} = -C_1^{T} M C_1 - C_2^{T} C_2 $$

$$ + \begin{bmatrix} (C_1^{T} M D_{11} + C_1^{T} D_{21})^T \\ (C_1^{T} M D_{12} + C_1^{T} D_{22})^T \end{bmatrix} \hat{R}^{-1} \begin{bmatrix} (C_1^{T} M D_{11} + C_1^{T} D_{21})^T \\ (C_1^{T} M D_{12} + C_1^{T} D_{22})^T \end{bmatrix}. $$

(20)

In both examples, the matrix variables $X$ and $Y$ are free parameters, whereas $\psi_\nu$ is a fixed basis function with preselected $\nu$ and $\rho$.

Consequently, the upper bound $\gamma$ on the worst case gain in Theorems 1 and 2 depends on the choice of the IQC parameterization $M$. Hence, an optimization problem over the feasible set of IQC parameterizations $M \in \mathcal{M}$ minimizing $\gamma$ can be derived. This optimization problem is constrained by the integrability of the RDE, i.e., the RDE can be integrated fully backward in time from $T$ to $t = 0$. Hence, the RDE has a finite escape time larger than the analysis horizon. In the case of the finite horizon worst case $L_2[0, T]$ to $\|e(T)\|_2$ gain, the optimization problem is written as

$$ \min_{\gamma \in \mathcal{M}} \gamma $$

such that $\forall t \in [0, T]$

$$ P(T) = \frac{1}{\gamma} C_2(T)^T C_2(T) $$

$$ \dot{P} = Q + P \hat{A} + \hat{A}^T P - P \hat{S} P $$

$$ R < 0. $$

(23)

Note that $M$ enters (23) in a nonconvex way. The nonlinear optimization problem for the finite horizon worst case $L_2[0, T]$ gain can easily be derived from (23) by replacing $\hat{A}$, $S$, and $Q$.
A. Algorithm

A novel, custom-tailored optimization algorithm is proposed to deal with problem (23) efficiently. The optimization essentially consists of a simple bisection nested within a global optimization algorithm. The bisection is used to obtain a minimal γ for a given M, i.e., bisect (23) with a fixed M'. The optimization over M ∈ M is based on the L-SHADE algorithm proposed in [30], but it is significantly tailored toward the specific optimization problem (23).

The L-SHADE algorithm belongs to the class of so-called metaheuristics. Its key philosophy can be summarized in the following way. At the start of the optimization, a random set of possible tuning parameters (populations) is generated, i.e., specifically, for (23), an initial set \( \{M^{(1)}_j\}_{j=1}^{n_p} \), where \( n_p \) is the number of populations. Throughout this section, \( M^{(i)}_j \) is the elements of the IQC parameterization stacked as a vector, where \( j \) denotes the \( j \)th individual within the population and \( i \) denotes the population iteration. At each new iteration \( i \), the population is updated with a bias toward the best 10% solutions of the previous iteration \( i - 1 \) by

\[
M^{(i)}_j = M^{(i-1)}_j + F_j \left( M^{(i-1)}_{\text{pbest}_j} - M^{(i-1)}_j + M^{(i-1)}_{r_{1,j}} - M^{(i-1)}_{r_{2,j}} \right) \tag{24}
\]

where \( M_{\text{pbest}_j} \) is a randomly selected individual from the best 10% of the population, \( M_{r_1,j} \) and \( M_{r_2,j} \) are two randomly selected individuals from the whole population, and \( F_j \) is a scaling factor chosen as a Cauchy distributed random scalar with variance 0.1 and a randomly selected mean value out of the set \( S_F \) of previously well-performing scaling factors (see [30] for details).

After performing the mutation, each element in \( M^{(i)}_j \) is replaced by its parent \( M^{(i-1)}_j \) using the binomial crossover. A uniformly distributed random number in the interval (0,1) is assigned to each element in \( M^{(i)}_j \). If this number is larger than the element’s crossover rate (CR), the respective element is replaced by its parent. The CR is chosen as a normal distributed random number with variance 0.1 and a mean value randomly chosen from the set \( S_C \) of previously well-performing CRs. However, one random element in \( M^{(i)}_j \) always remains updated independently of its CR.

After finishing the crossover, the bound constraints are checked. In the case of a violation, the respective elements in \( M^{(i)}_j \) are set to the arithmetic mean value of the corresponding parent elements in \( M^{(i-1)}_j \) and the respective violated boundary.

The following adaptations have been made to the original algorithm proposed in [30]. The algorithm uses a logarithmic scaling of the decision variables instead of the linear scaling used in [30]. The search space for the IQC parameters usually covers several orders of magnitudes with no clear indication of good initial values. For instance, the diagonal entries of \( X \) in Example 2 are only restricted by positiveness. By searching over a logarithmic scale, the correlation between a change in \( \gamma \) and the variation of the elements in the IQC parameterization \( M \) is better represented. This is especially true considering very small magnitudes. Hence, the metaheuristic converges at the same speed independently of the optimal solution’s magnitudes. Using a logarithmic search space, the single elements in the decision vector \( M^{(i)}_j \) are now represented by two elements containing their sign and exponent to base 10. These two elements are stacked into single vectors \( M^{(i)}_{\text{Log},j} \) forming the actual set of tuning parameters in the logarithmic space \( \{M^{(i)}_{\text{Log},j}\}_{j=1}^{n_p} \). Note that this logarithmic scaling effectively increases the number of decision variables. It should be emphasized that, in general, it does not double the number of variables as many IQC parameters are sign defined, e.g., the diagonal entries of \( X \) in Example 2.

MHs do not require a valid initial population set, i.e., a finite \( \gamma \) value exists for a given \( M^{(i)}_j \) such that the RDE in (23) is fully integrable. In general, \( \gamma \) cannot be calculated for all \( M \in M \) due to the RDE’s finite escape time [31]. The algorithm is adjusted to require at least 20% valid members in its initial population before commencing the mutation to improve convergence. The chosen numerical value showed a good tradeoff between convergence and computational effort in multiple studies. In the case of multiple IQCs, an optional downscaling of the uncertainties’ norm bounds by a factor \( k_{\text{IQC}} \) facilitates identifying a valid initial population by increasing the feasible search space. Extensive studies showed that optimum locations are not noticeably affected by the value of the norm bounds. Thus, procedural rescaling them does not adversely affect the search performance. Furthermore, initial guesses \( M^{(i)}_{\text{Log,init}} \) can be added to the initial population.

Integrating the RDE is the main contributor to the algorithm’s computation time. Hence, several adjustments have been made to reduce the number of necessary integrations. The lower bound is provided either by the nominal \( \gamma \) value [14] or a theoretical lower bound imposed by \( R < 0 \) and Schur’s complement. For the initial population, an adaptive upper bound is implemented. Inside the iterative search procedure, the upper bound used for \( M^{(i)}_j \) is the minimal \( \gamma^{(i-1)} \) identified for the corresponding parent \( M^{(i-1)}_j \). If the RDE is not fully solvable for \( \gamma^{(i-1)} \), no bisection is executed, and \( M^{(i)}_j \) gets assigned \( \gamma^{(i)} = 10^{20} \). This adjustment is reasoned by the fact that, if the RDE is not solvable for the upper bound, it is also not solvable for all lower values of \( \gamma \), and thus, \( \gamma^{(i)} > \gamma^{(i-1)} \). As the mutation is solely based on the offspring with improved \( \gamma \) values, there is no adverse effect on the search performance.

Finally, the condition \( R < 0 \) in (23) can be further exploited to reduce the number of integrations in the optimization. If in a bisection step \( R \geq 0 \), it is immediately treated as a failed integration. The integration is also skipped if \( R \) is poorly conditioned, i.e., its condition number is larger than a user-defined threshold. This avoids numerical difficulties integrating the RDE. Extensive test scenarios showed no adverse effects on the search performance.

In general, RDEs are considered stiff [31]. The RDE is solved via MATLAB’s built-in solver ODE15s. It is chosen as it outperformed MATLAB’s remaining solvers for stiff
Algorithm 1 Log-L-SHADE

1: **Input**: \( n_{\text{Pmax}}, n_{\text{Pmin}}, i_{\text{max}}, k_{\text{F}}, k_{\text{CR}}, H, \mathcal{M}, s_{\text{min}}, s_{\text{max}}, \gamma_{\text{lim}}, M_{\log\text{init}}, k_{\text{QIC}}, \gamma_{\text{LB}}, \gamma_{\text{UB}}, \epsilon_{\text{BS}} \)
2: **Output**: \( \gamma_{\text{best}}, M_{\log\text{best}} \)
3: **Initialize**: \( S_{F}, S_{\text{CR}} \)
4: Scale original IQC norm bound by \( k_{\text{QIC}} \)
5: while Less than 20\% valid initial members do
6: Generate random initial population \( \{ M_{i, j}^{(t)} \}_{j=1}^{n_p} \)
7: Calculate \( \gamma(M_{i, j}^{(t)}) \) via bisection
8: end while
9: Find current best solution \( M_{\log\text{best}} \) and fitness \( \gamma_{\text{best}} \)
10: Set IQC norm bound upsampling threshold \( n_{p_{\text{QIC}}} = n_{p_{\text{max}}} \)
11: while \( (i \leq i_{\text{max}} \) OR \( \gamma_{\text{best}} > \gamma_{\text{lim}} \)) AND \( k_{\text{QIC}} < 1 \) do
12: \( i = i + 1 \)
13: if \( (\gamma_{\text{best}} \leq \gamma_{\text{lim}} \) OR \( n_{p} < 0.8n_{p_{\text{QIC}}} \)) AND \( k_{\text{QIC}} < 1 \) then
14: \( n_{p_{\text{QIC}}} = n_{P} \) and \( k_{\text{QIC}} = \min(3k_{QIC}, 1) \)
15: Upscale norm bound, recalculate \( \gamma(M_{i, j}^{(t-1)}) \) with user \( \gamma_{\text{UB}} \), and update \( M_{\log\text{best}} \) and \( \gamma_{\text{best}} \)
16: end if
17: for \( j = 1 \) to \( n_{p} \) do
18: Calculate \( M_{i, j}^{(t)} \) using (24) and crossover, enforce boundaries, and \( M_{i, j}^{(t)} \in \mathcal{M} \)
19: if RDE is solvable for \( M_{i, j}^{(t)} \) and \( \gamma(M_{i, j}^{(t-1)}) \) then
20: Execute bisection with \( \gamma_{\text{UB}} = \gamma(M_{i, j}^{(t-1)}) \)
21: else
22: Skip bisection and set \( \gamma(M_{i, j}^{(t)}) = 10^{20} \)
23: end if
24: if \( \gamma(M_{i, j}^{(t)}) > \gamma(M_{i, j}^{(t-1)}) \) then \( M_{i, j}^{(t)} = M_{i, j}^{(t-1)} \)
25: end if
26: end for
27: Update \( S_{F} \) and \( S_{\text{CR}} \) based on successful \( F \) and \( CR \)
28: Identify current best solution \( M_{\log\text{best}} \) and fitness \( \gamma_{\text{best}} \)
29: Update population \( n_{p} \) size via (25) and remove worst solutions from \( \{ M_{i, j}^{(t)} \}_{j=1}^{n_p} \)
30: end while

ODEs in multiple evaluated test scenarios and applications. Its integrated event function is used to detect blow-ups resulting from finite-escape times shorter than the analysis horizon and terminate the integration. The event function triggers if the largest absolute eigenvalue of \( \dot{P} \) is above a provided threshold. The computational effort is further reduced by exploiting the RDE’s symmetry. Thus, only \( 0.5n(n+1) \) instead of \( n^2 \) equations must be solved.

Algorithm 1 presents the pseudocode to illustrate the implementation of the optimization problem. The user must provide a total of sixteen inputs. The first two are the maximum and minimum population sizes \( n_{p_{\text{max}}} \) and \( n_{p_{\text{min}}} \), respectively. The population size refers to the number of guesses for the optimal solution evaluated at each iteration. These are followed by the maximum amount of population iterations \( i_{\text{max}} \) and the number of successful CRs and scaling factors \( k_{\text{CR}} \) and \( k_{\text{F}} \). \( H \) is the extended system (4), which includes the user-selected fixed IQC filter \( \Psi \). It is followed by \( \mathcal{M} \) describing the set of feasible IQC parameterizations. The inputs \( s_{\text{min}} \) and \( s_{\text{max}} \), with \( s_{\text{min}} \in \mathbb{N}^m \) and \( s_{\text{max}} \in \mathbb{N}^m \), define the logarithmic search space, i.e., minimal and maximal exponents, for each element in \( M_{i, j}^{(t)} \). The norm bound scaling \( k_{\text{QIC}} \) and initial guess \( M_{\log\text{init}} \) are optional. They can be provided to improve the search performance. The input \( \gamma_{\text{lim}} \) is used as scaling and a terminal condition. The remaining three inputs are required to run the bisection, its lower and upper bounds \( \gamma_{\text{LB}} \) and \( \gamma_{\text{UB}} \), and its relative tolerance \( \epsilon_{\text{BS}} \).

The algorithm is initialized with the vectors \( S_{F} \in \mathbb{R}^{k_{F}} \) and \( S_{\text{CR}} \in \mathbb{R}^{k_{CR}} \) containing \( k_{F} \) and \( k_{CR} \) elements, respectively, with a value of 0.5. Subsequently, an initial population is generated and evaluated via bisection using user-defined bounds. After the required amount of valid members is reached, the current best solution \( M_{\log\text{best}} \) and respective \( \gamma_{\text{best}} \) are identified.

Now, the tailored MH’s iterative search procedure starts. First, the uncertainty norm bounds are upscaled if either \( \gamma_{\text{best}} < \gamma_{\text{lim}} \) or \( n_{p} < 0.8n_{p_{\text{QIC}}} \), with \( n_{p_{\text{QIC}}} = n_{p_{\text{max}}} \) for the first iteration and \( k_{\text{QIC}} < 1 \). If the norm bound is upscaled, the population is reevaluated, and the new best solution is identified. The additional upsampling condition using 0.8\( n_{p_{\text{QIC}}} \) reduces the number of evaluations and showed no negative impact on the convergence. It proceeds with the population update using mutation (24) and crossover in the logarithmic domain. Afterward, the bisection with updated upper bounds is executed to calculate the minimal \( \gamma(M_{\log\text{best}}^{(t)}) \) to a relative accuracy of \( \epsilon_{\text{BS}} \). The computation is fully parallelizable, i.e., the number of accessible processor cores is the direct inverse of the computation time. If the child is an improvement over its parent, it replaces its parent in the next population iteration. Otherwise, the parent is used for the next iteration.

Before the next iteration starts, the population size \( n_{p} \) is updated using

\[
n_{p} = n_{\text{p, max}} - \min \left( \frac{(n_{\text{p, max}} - n_{\text{p, min}}) i}{i_{\text{max}} n_{\text{p, max}}} \right).
\]

If it decreases, then the worst excess solutions in \( \{ M_{i, j}^{(t)} \}_{j=1}^{n_p} \) are removed to match the new population size. The optimization concludes as soon as the maximum number of iterations is reached or the best \( \gamma \) value is lower than the threshold, given that the norm bounds are rescaled. The algorithm returns \( \gamma_{\text{best}} \) and the respective solution \( M_{\log\text{best}} \).

V. Example

The algorithm proposed in Section IV is applied to identify the worst case aerodynamic load acting on a space launcher during its atmospheric ascent. The analysis covers the flight segment from 25 to 95 s after liftoff, which includes the most critical regions of atmospheric disturbance and the dynamic pressure acting on the launcher. A realistic uncertainty set modeled via IQCs and a realistic wind disturbance model are introduced. The LTV worst case results are compared in a Monte Carlo type analysis conducted on the nonlinear model covering the same uncertainty and disturbance set.

A. Space Launcher Model

The investigated ELV is built of three solid rocket stages and an upper module using liquid propellant. It is designed to
launch small payloads into polar and low Earth orbits. During the ascent, the ELV is exposed to high dynamic pressures leading to substantial aerodynamic loads. This is accompanied by unsteady aerodynamics in the transonic region [32]. The launcher is also subject to various disturbances. The most influential of these is wind [33]. Nevertheless, the launcher has to stay inside a small design envelope to maintain its structural integrity and deliver the payload into the correct injection orbit.

1) Nonlinear Dynamics: The ELV is assumed symmetrical during the ascent as it is common practice for most space launchers [34]. Thus, the pitch and yaw motion can be described by the same dynamics with negligible cross-coupling [35]. As a result, it is sufficient to solely consider the space launcher’s pitch motion for the analysis. Furthermore, following the standard practice, the influences of a spherical and rotating Earth to analyze the atmospheric flight phase are neglected [35]. In addition, propellant sloshing and nozzle inertia are ignored. These mainly influence the flexible modes of an ELV. However, the considered analysis aims at finding the worst case static loads. In Fig. 3, a schematic of the launch vehicle is given. A launcher-fixed coordinate system with subscript \(b\) is used to formulate the nonlinear equations of motion. It is fixed to the center of mass of the launcher \(G\). Its \(x_b\)-axis is aligned with the launcher symmetry axis defined in the direction of forward travel. The \(z_b\)-axis forms a right-hand system with the \(y_b\)-axis pointing out of the page. Accordingly, the rigid body motion in the pitch plane formulated in body-fixed coordinates is described by

\[
\dot{\theta}_b(t) = \frac{\sum M_y(Ma, \alpha, h, t)}{J_y(t)} - \frac{T(t)l_{CG}(t)}{J_y(t)} \sin \delta_{TVC}(t)
\]

\[
\ddot{\theta}_b(t) = \frac{\sum F_y(Ma, \alpha, h, t)}{m(t)} - \dot{\theta}_b(t) \dot{z}_b(t)
\]

\[
\ddot{z}_b(t) = \frac{T(t) \cos \delta_{TVC}(t) - X(Ma, \alpha, h, t)}{m(t)} - g_0(h) \sin \theta_b(t) - \dot{\theta}_b(t) \dot{z}_b(t)
\]

where \(\sum M_y\) is the sum of the angular moments in the pitch plane with respect to the center of gravity \(G\). \(\sum F_y\) and \(\sum F_z\) describe the sum of forces in \(x_b\)- and \(z_b\)-directions, respectively. The angle \(\theta_b\) is the pitch angle of the launcher describing the angle between the body axis and the local horizon. The normal aerodynamic force is denoted \(N\). It is described by

\[
N(Ma, \alpha, h, t) = Q(h, t)S_{ref}C_{N_0}(Ma)\alpha(t)
\]

with

\[
Q(h, t) = 0.5 \rho(h, t)V(t)^2
\]

being the dynamic pressure. \(C_{N_0}\) is the normal lift force coefficient, which depends on the Mach number \(Ma\). \(V(t) = (\dot{x}_b(t)^2 + \dot{z}_b(t)^2)^{1/2}\) is the launcher’s velocity. The density of the air \(\rho\) is calculated according to the International Standard Atmosphere (ISA). \(N\) acts parallel to the \(z_b\)-axis. It is defined as positive in the negative \(z_b\)-direction. The axial aerodynamic force \(X\) is defined in the same way but with respect to the \(x_b\)-axis. \(X\) is described by the following equation:

\[
X(Ma, \alpha, h, t) = Q(h, t)S_{ref}(C_{x_0}(Ma) + C_{x_1}(Ma)\alpha)
\]

where \(C_{x_0}\) is the zero-lift and \(C_{x_1}\) is the lift-dependent axial force coefficient, respectively. Both coefficients are \(Ma\) dependent. The definition differs from the common aerospace convention formulating lift and drag parallel, respectively, orthogonal to the aerodynamic velocity \(V\). In (27) and (29), the angle of attack is approximated as

\[
\alpha(t) \approx \frac{\dot{z}_b(t) - v_w(t)}{\dot{x}_b(t)}
\]

where \(v_w\) is the wind speed. It is aligned with the \(z_b\)-axis and defined as positive in the \(z_b\)-direction. \(T\) denotes the time-dependent thrust of the engine acting at the nozzle pivot point \(C\). The geometric variables \(l_{GA}\) and \(l_{CG}\) denote the distance between the center of gravity \(G\) and the center of aerodynamic pressure \(A\) and \(C\), respectively. All aerodynamic forces act on \(A\). \(G\) moves forward during the flight due to the propellant burn, whereas \(A\)’s location depends on the \(Ma\) number. \(J_y\) and \(m\) denote the mass moment of inertia and the launcher’s mass, respectively, which vary with time due to the fuel burn. The gravitational acceleration \(g_0\) is calculated according to the world geodetic systems WGS-84 [36] as a function of altitude. As the only control variable, the deflection of the thrust vectoring control \(TVC\) \(\delta_{TVC}\) is available. The dynamics of the TVC actuator are given by the following second-order system:

\[
G_{TVC} = \frac{1}{0.000374 \text{ s}^2 + 0.0384 \text{ s} + 1}
\]

2) Trajectory and Control Design: The ascent trajectory is a standard gravity turn assuring \(\alpha \approx 0^\circ\) and \(\delta_{TVC} \approx 0^\circ\) during the nominal ascent. The purpose is to minimize the static aerodynamic load and maximize the longitudinal acceleration.
This is beneficial for the launcher design as it minimizes structural and fuel mass. The reference flight path parameters for the analyzed ELV are calculated by iteratively solving the following initial value problem:

\begin{align*}
\dot{h}_{\text{ref}}(t) &= \dot{x}_{b,\text{ref}}(t) \sin \theta_{b,\text{ref}}(t) \\
\dot{\theta}_{b,\text{ref}}(t) &= -\frac{g_0(h)}{x_{b,\text{ref}}(t)} \cos \theta_{b,\text{ref}}(t) \\
\ddot{x}_{b,\text{ref}}(t) &= \frac{T(t) - X(Ma, \alpha, h, t)}{m} - \frac{g_0(h)}{m} \sin \theta_{b,\text{ref}}(h)
\end{align*}

(32)
given in [37]. The initial values for \( \theta_b \), \( \dot{x}_b \), and the altitude \( h \) are iterated until the trajectory's desired terminal values are achieved. The calculated flight path represents a common mission scenario [38].

Due to the lack of aerodynamic surfaces, the analyzed ELV is aerodynamically unstable. Hence, feedback control is required to stabilize its dynamics along the trajectory. Furthermore, it is necessary to track the precalculated gravity turn trajectory. In this article, this is achieved by executing a pitch program following the reference pitch angle \( \theta_{b,\text{ref}} \) calculated via (32).

Many modern launch vehicles still rely on rather basic proportional, integral, and derivative (PID) control [34]. Therefore, a simple fixed-gain PID controller \( C \) with a derivative filter for \( \theta_b \) tracking was designed. As the design point, the point of the maximum dynamic pressure along the reference trajectory was chosen, which is at \( t = 54 \) s. The gains are calculated via loop shaping, providing a maximum tracking bandwidth of 6 rad/s without actuators. The controller satisfies phase and gain margins of 45° and 6 dB, respectively, along the trajectory as proposed in [39].

The controller was evaluated in the nominal nonlinear ELV simulation. The ELV follows the pitch program precisely under nominal conditions \((\Delta \theta(t) < 0.0001 \degree)\), i.e., without uncertainties or a wind disturbance. Therefore, \( \alpha \) remains approximately \( 0 \degree \) fulfilling the gravity turn. Under allowable wind disturbance, i.e., suitable launch conditions, \(|Q(t)\alpha(t)| < 22,000 \) Pa as it is required by the European Space Agency (ESA) guidelines. This assures that the ELVs structural limit loads are not exceeded [40].

3) Linear Dynamics: The nonlinear launcher dynamics in (26) are linearized along the ascent trajectory. This reduces the various parameter dependencies in (26) to a sole time dependence along the calculated trajectory. Thus, the linearization results in a (general) LTV system \( G_{\text{ELV}} \) as described by (2). In the case of the analyzed ELV, the input vector is \( d(t) = [\delta_{T\text{VC}}, v_w]^T \), the output vector is \( e(t) = [\theta_b, \dot{\theta}_b]^T \), and the states are \( \theta_b, \dot{\theta}_b, \) and \( \dot{x}_b \). \( Q \alpha \) is the product of \( Q(t) \) and the linearized \( \alpha \) output equation derived from the relation in (30). Note that \( \dot{x}_b \) does not need to be considered in the analysis, as it has no impact on the maximum aerodynamic load. The matrix functions \( A_G, B_G, C_G, \) and \( D_G \) are calculated via piecewise cubic Hermite interpolating polynomials based on grid points derived by numerical linearization with a grid density of \( \Delta t = 0.1 \) s. The chosen grid is dense enough to capture the fast-changing dynamics in the transonic region during the ascent.

4) Uncertainty Model: The uncertainty in the launcher’s aerodynamics arises mainly from the limited means of testing and the resulting reliance on simulation results. Furthermore, the launcher passes through the transonic \((0.8 \leq Ma \leq 2.0)\), for which the estimation of aerodynamic parameters is complicated. Especially difficult to estimate is the center of aerodynamic pressure, due to complicated airflow originating from the payload fairing. This is accounted for by uncertainty in \( l_{CG} \), which directly influences the aerodynamic instability of the launcher. In addition, the launcher’s center of gravity is subject to uncertainty due to variations in the fuel burn. Therefore, uncertainty in \( l_{CG} \) is introduced. It directly influences the controllability. Finally, the TVC’s dynamics are treated as uncertain, primarily to account for modeling errors. The introduced uncertainties have little influence on the ascent trajectory. Thus, the reference trajectory to obtain the nominal model maintains its validity.

The uncertainties in the launcher’s time-varying parameters for the ascent are all described by (repeated) LTI real parametric uncertainties, i.e., \( l_{CG}(t) = l_{CG,\text{nom}}(t)(1 + \delta_{lCG}) \). The uncertainty in the TVC dynamics is represented with a dynamic LTI uncertainty \( \Delta_{TVC} \) with \( \|\Delta_{TVC}\|_\infty < 1 \). It is implemented as

\[ G_{\text{TVC}} = G_{\text{TVC,\text{nom}}}(1 + W_{\text{TVC}}\Delta_{\text{TVC}}) \]

with a weighting filter \( W_{\text{TVC}}(s) \). \( W_{\text{TVC}} \) is calculated based on the approach in [41]. It covers a time delay of 10 ms and up to 10% uncertainty in the TVCs static gain, damping ratio, and eigenfrequency.

5) Wind Model: The wind disturbance is based on the Dryden turbulence filter for light lateral wind turbulence [42]. Dryden turbulence spectra are widely used in aerospace certification processes.

a) Wind filter nonlinear analysis: The Monte Carlo simulation of the nonlinear model applies the standard Dryden filter for lateral wind turbulence \( G_w \).

\[
\begin{align*}
\dot{x}_w(t) &= \begin{bmatrix} 0 \\ -\frac{1}{L(h)} \end{bmatrix} x_w(t) + \begin{bmatrix} 1 \\ \frac{V(t)}{L(h)} \end{bmatrix} w_w(t) \\
\dot{v}_w(t) &= \begin{bmatrix} \sigma(h) \sqrt{\frac{L(h)}{\pi V(t)}} \\ \sigma(h) \frac{\sqrt{L(h)}}{\sqrt{\pi V(t)}} \end{bmatrix} v_w(t)
\end{align*}
\]

(34)
as given in [42]. In (34), \( V(t) \) is the velocity of the ELV, \( \sigma \) is the turbulence intensity, and \( L_a \) is the turbulence scale length. The last two variables are altitude dependent, and their values are chosen for light turbulence according to [43]. This filter shapes a white noise input \( n_w(t) \) with a power spectral density (PSD) \( \Phi_{n_w} = 1 \) into a continuous turbulence signal \( v_w(t) \). Here, the Simulink internal band-limited white noise block is used to generate \( n_w(t) \). The lateral filter is chosen, as, for altitudes over 533 m, the wind turbulence is defined as being aligned with the body fixed coordinates. This holds for the analyzed trajectory segment. Hence, the calculated wind turbulence is consistent with the definition of \( v_w \) in (30).

b) Wind filter LTV analysis: The wind filter \( G_w \) cannot be applied directly in the LTV analysis due to the strict BRL. Recall that the Dryden filter (34), in contrast, assumes a white
noise input. Hence, to get meaningful analysis results, a wind filter for the LTV analysis must be designed to take any bounded L2 signal and generate realistic Dryden turbulence signals. Specifically, the wind filter design goal is to match the PSD of the Dryden turbulence.

The proposed design procedure is given as follows. In the first step, 10,000 random wind profiles are generated along the nominal trajectory over the analysis horizon using the Dryden filter (34) at a fixed sampling rate of 100 Hz. The second step is the calculation of the PSD. The wind signals are divided into 14 equidistant segments of 5 s. This separation of a wind signal \( v_{w,i}(t) \) into segments \( v_{w,i,n} \) accounts for the varying turbulence intensity along the trajectory. The PSD \( \Omega_{v_{w,i,n}} \) of a segment \( n \) of a time-domain wind signal \( v_{w,i}(t) \) is calculated using

\[
\Omega_{v_{w,i,n}}(\omega) = \lim_{T \to \infty} \frac{1}{T} \left| \int_{-T}^{T} v_{w,i,n}(t)e^{-j\omega t} dt \right|^2.
\] (35)

Therefore, the PSD of a time-domain signal is simply the average square of the signal’s Fourier transform. The Fourier transform of the wind signals can be calculated via a fast Fourier transform (FFT). Here, the internal MATLAB function \texttt{fft} was applied. This procedure is repeated for all segments \( n \) over all wind signals \( v_{w,i}(t) \). In the third step, a transfer function is calculated for each time segment, upper bounding the respective \( |\Omega_{v_{w,i,n}}(\omega)| \) over all wind signals. Therefore, the internal MATLAB function \texttt{fitmagfrd} is used, applying the lower bound \( [\Omega_{v_{w,i,n}}(\omega)] \). It calculates a minimum phase first-order transfer function based on the log-Chebychev magnitude design. The transfer functions are transformed into consistent state-space models. Afterward, the LTV representation \( G_{w,LTV} \) of the wind model is calculated by linear interpolating the system matrices’ coefficients over the analysis horizon. In the fourth step, the calculated wind filter is evaluated to check if it produces adequate wind signals. Therefore, the nominal launcher model is extended with \( G_{w,LTV} \), and the nominal LTV worst case disturbance signal \( d_{WC} \) for a given terminal time is calculated using the approach in [44]. Filtering \( d_{WC} \) through \( G_{w,LTV} \) provides the respective worst case wind signal. Its peak values are compared to the peaks of actual Dryden turbulence signals. In case the LTV wind signals underestimate the peak values of actual Dryden turbulence, steps 3 and 4 are repeated with an increased lower bound for \texttt{fitmagfrd} until the amplitudes show an adequate match. In combination, steps 3 and 4 assured matching PSDs of the LTV worst case wind signal and actual Dryden turbulence. In Fig. 4, the worst case LTV wind signal’s PSD for the time segment from 25 to 30 s is compared to the corresponding Dryden turbulence signal. The presented procedure to derive the LTV wind filter is not limited to Dryden turbulence but can be easily applied to any available wind data.

6) Analysis Interconnection: In Fig. 5, the interconnection used for the LTV analysis is shown. The ELV’s dynamics are described by the respective LTV system \( G_{ELV} \). All other blocks correspond to the systems discussed in Sections V-A1–V-A5. On the contrary, the Monte Carlo simulation directly utilizes the nonlinear launcher’s dynamics, as given in (26), and the wind model for the nonlinear analysis is applied.

Fig. 4. Comparison of power spectral densities for the analysis segment from 25 to 30 s: LTV worst case wind signal ( ) and Dryden turbulence ( ).

Fig. 5. \( Q\alpha \) analysis interconnection.

TABLE I

| Parameter | Notation | Value | Occurrences | Type |
|-----------|----------|-------|-------------|------|
| \( I_{CG} \) | \( \delta_{CG} \) | 10%   | 1           | real |
| \( I_{GA} \) | \( \delta_{GA} \) | 20%   | 2           | real |
| TVC | \( \Delta_{TVC} \) | 15.47 | 1           | dynamic |

Both analyses apply the same nominal TVC model \( G_{TVC} \) and PID controller \( C \).

B. Analysis Results

The LTV worst case aerodynamic load \( Q_{\alpha,WC} \) is calculated, solving the corresponding optimization problem (23) using Algorithm 1. Due to the definition of the \( L_2[0, T] \)-to-Euclidean gain, the value of \( Q\alpha \) is only upper bounded at the final time of the specific analysis horizon. Therefore, analysis over a set of final times \( T_i \) covering the trajectory is necessary to identify \( Q_{\alpha,WC} \). An evenly spaced set of final times \( T_i \) spanning from 30 s to 95 s with a step size of 5 s is chosen.

The interconnection in Fig. 5 must be transformed into the IQC framework, as specified in Section III-C. The individual uncertainties given in Table I are replaced by the IQC description given in Examples 1 and 2, respectively, and stacked diagonally in a single IQC. The respective values for the norm bound \( b \) (defining the uncertainty size), as well as the chosen parameter of the basis function \( \nu \) and \( \rho \), are given in Table II. Changing the value of \( \rho \) around the chosen values showed no noticeable influence on the calculated worst case gain. The worst case gain starts to increase for \( \rho \) approaching 100 and 0,
The turbulence is altitude-dependent, reaching its highest values between 30 and 45 s after liftoff, corresponding to altitudes of 4.66 and 9.95 km, respectively. Simultaneously, the dynamic pressure builds up from $4.52 \times 10^4$ to $5.47 \times 10^4$ Pa, resulting in the flight’s highest aerodynamic loads. However, when the dynamic pressure reached its maximum $Q_{\text{max}} = 5.603 \cdot 10^4$ Pa at 52 s after liftoff (12.9 km), the turbulence intensity is reduced to $\sigma_u = 0.05 \text{ m/s}$, and the wind turbulence is negligible.

Afterward, $Q_{\alpha_{\text{WC, max}}}$ is compared to the results of the corresponding nominal LTV worst-case analysis, i.e., with the nominal launcher and TVC dynamics. The nominal worst case load of $Q_{\alpha_{\text{nom, max}}} = 28.27\%$ occurring at 30 s after liftoff is less than $Q_{\alpha_{\text{WC, max}}}$. Thus, it can be concluded that the external disturbance rather than the perturbation in the launcher’s parameters is the key influence on the expected worst case loads in this study. This is an important outcome for control design approaches. The resulting maximum loads of the nominal and IQC (uncertain) worst case LTV analyses are summarized in the first row of Table III.

A Monte Carlo simulation is run on the nonlinear launcher model to validate the worst case analysis results. It directly applies the Dryden filter (34). Instead of a dynamic TVC uncertainty, the corresponding parametric uncertainties specified in Section V-A6 are directly implemented. One thousand white noise input disturbances are considered, generated by the Simulink internal band-limited white noise block with uniformly distributed random noise seeds. The parametric uncertainties are uniformly gridded over their definition space, i.e., five points to cover $\pm 20\%$ uncertainty in $l_{GA}$ and 5 points to cover $\pm 10\%$ uncertainty $l_{GG}$. Finally, the static gain, eigen-frequency, and damping ratio of the TVC are gridded over $\pm 10\%$ with three equidistant points. In addition, two different time delays are considered, namely, 0.005 and 0.01 s. Hence, the Monte Carlos analysis considers the original explicit uncertainty set of the TVC dynamics, which simplifies implementation. Subsequently, each of the resulting 1350 models is evaluated for every noise signal $n_{w}(t)$. A single execution of the nonlinear simulation takes an average of 1.5 s, resulting in an overall analysis time of approximately 15 and a half days for a relatively coarse analysis grid. It is reduced to an effective time of 3 d 22 h, distributing the analysis to four computers equipped with Intel Xeon E-5 1620 v4 processors and 32-GB memory.

| Uncertainty | $b$ | $\nu$ | $\rho$ |
|-------------|-----|------|------|
| $\Delta_{\text{TVC}}$ | 1   | 1    | -1   |
| $\delta_{IGA}$ | 0.2 | 1    | -1   |
| $\delta_{IGG}$ | 0.1 | 1    | -1   |

**TABLE II**
Parameters Used for the IQC

**TABLE III**
Comparison of Identified Maximal Values of $Q_{\alpha}$ as Percentage of Limit Load Along the Trajectory

| Analysis Type | Nominal Launcher | Launcher with Fixed Perturbations | Launcher with IQC Perturbations |
|---------------|-----------------|----------------------------------|---------------------------------|
| LTV Worst-Case| 28.27%           | 29.2%                            | 29.9%                           |
| Monte Carlo   | 19.6%            | 22.0%                            | -                               |

The whole interval can be solved using the proposed algorithm, on average, in 2 h 55 min over seven runs on a standard computer with an Intel Core i7 processor and 32-GB memory. For the analysis, the bisection step of the optimization was parallelized on eight physical cores. The maximum worst case load $Q_{\alpha_{\text{WC, max}}}$ calculated for the trajectory is 29.9% of the limit load $Q_{\alpha_{\text{lim}}} = 220 000 \text{ Pa}$ occurring at 30 s after liftoff. It correlates with the highest expected turbulence intensities.
The nonlinear simulation starts at $t_i = 25$ s and ends at $t_f = 95$ s after liftoff. A maximum aerodynamic load $Q_{\alpha_{\text{MC,max}}}$ of 22.0% of $Q_{\alpha_{\text{lim}}}$ is identified at 32.1 s, with the corresponding uncertainty description as follows: $\delta_{CG} = -0.1$, $\delta_{CA} = 0.2$, $\delta_{\zeta} = -0.1$, $\delta_{\omega} = -0.1$, $\delta_{\xi} = -0.1$, and $\tau = 0.01$ s. The respective $Q_{\alpha_{\text{MC,max}}}$ signal scaled by the limit load $Q_{\alpha_{\text{lim}}}$ is shown in black in Fig. 6, where it is compared to the LTV worst case aerodynamic load envelope. Note the latter’s values in-between the analysis grid points $T_i$ are linearly interpolated. Furthermore, an envelope covering the peaks of all simulated $Q_{\alpha_{\text{MC}}}$ signals and the corresponding signals are plotted in Fig. 6. The former begins at $T_1 = 30$ s, corresponding to the LTV analysis, by which it is upper bounded for the whole trajectory.

As for the LTV case, a nominal simulation is conducted to assess the influence of the uncertainties. Therefore, nominal nonlinear ELV is evaluated for all $n_{\alpha}(t)$. It provides a maximum nominal load $Q_{\alpha_{\text{MC,nom}}}$ of 19.6% of the limit load, showing little influence of the perturbations. The maximum $Q\alpha$ values of both Monte Carlo simulations are summarized in Table III. Note that Monte Carlo simulation under uncertainty corresponds to the “Launcher with fixed perturbations” column (as they have explicit values for each run).

For the gap between the LTV and the nonlinear analysis, the fundamental difference in both analyses’ nature is the leading cause. The worst case LTV analysis calculates a guaranteed upper bound for $Q\alpha$ of the interconnection in Fig. 5. On the contrary, the Monte Carlo analysis can only provide a lower bound. Furthermore, in the LTV analysis, the disturbance input is a worst case norm bounded signal, whereas, in the nonlinear simulation, it is an arbitrary band-limited white noise signal. The resulting wind disturbance signals have a comparable PSD due to the LTV wind filter design in Section V. However, it is unlikely that the exact LTV worst case wind signal will be under the evaluated signals in the Monte Carlo simulation. Another contributor is dynamic uncertainty $\Delta$, which likely introduces some conservatism in the LTV analysis compared to the nonlinear analysis’ parametric uncertainties.

In conclusion, the behaviors of LTV and the nonlinear launcher model are compared to further assess the suitability of the LTV approach. Therefore, the worst case disturbance signals of the LTV analysis interconnection in Fig. 5 are calculated for a fixed uncertainty combination. This means a fixed value is chosen for each uncertainty, which results in a new “nominal” launcher and actuator dynamic allowing a nominal LTV worst case analysis. This uncertainty combination corresponds to $Q_{\alpha_{\text{MC,max}}}$, namely, $\delta_{CG} = -0.1$, $\delta_{CA} = 0.2$, $\delta_{\zeta} = -0.1$, $\delta_{\omega} = -0.1$, $\delta_{\xi} = -0.1$, and $\tau = 0.01$ s. A second-order Padé approximation is used to approximate the behavior of the time delay. Afterward, the approach proposed in [44] is applied to calculate the worst case disturbance for each $T_i$. Note that the approach in [44] requires nominal LTV models and calculates the nominal worst case gain related to (5). The uncertainty combination must be identified before (here via the previous Monte Carlo simulation), and the approach, thus, can only provide a lower bound. Afterward, the perturbed LTV model is evaluated for the worst case disturbance signals $d_{\text{WC},i}(t)$ over the complete trajectory, i.e., from 25 to 95 s, with

$$
d_{\text{WC},i}(t) := \begin{cases} 
d_{\text{WC},i}(t), & \text{for } t \leq T_i \\
0, & \text{for } t > T_i 
\end{cases} \tag{36}
$$

To facilitate the comparison with the nonlinear simulation. The corresponding LTV worst case wind signals $w_{\text{WC},i}$ are then used as simulation input in the accordingly perturbed nonlinear model. Fig. 7 compares the resulting $Q\alpha$ signals of the LTV and nonlinear simulation representatively for final times $T = 35$ s and $T = 40$ s. A comparison to the LTV IQC worst case envelope shows no violation. Furthermore, the LTV and nonlinear simulation results match closely, validating the LTV modeling approach’s suitability for launch vehicles.

VI. Conclusion

The presented analysis framework offers an efficient approach to calculate the worst case gain of uncertain finite horizon LTV systems. A specifically developed algorithm efficiently exploits the structure of the optimization problem facilitating the analysis of industry-sized problems over large horizons. Its applicability is demonstrated using an LTV worst case loads analysis of a space launcher. It is validated against...
a Monte Carlo simulation conducted on the corresponding nonlinear model. The LTV analysis provides a valid, not overly conservative upper bound in a fraction of the time required for the Monte Carlo simulation. Hence, the presented analysis framework and novel algorithm provide a valuable supplemental tool for certification processes.

REFERENCES

[1] A. Hošovský, J. Pitel, K. Žídek, M. Tříthová, J. Sárosi, and L. Cvetinac, “Dynamic characterization and simulation of two-link soft robot arm with pneumatic muscles,” Mechanism Mach. Theory, vol. 103, pp. 98–116, Sep. 2016.

[2] F. He, L. Wang, Y. Yao, H. Qi, and Z. Jiang, “Analysis and design of linear time-varying systems based on a finite-time performance index,” in Proc. Amer. Control Conf. (ACC), Jul. 2015, pp. 5948–5953.

[3] W. Hauesermann, “Description and performance of the Saturn launch vehicle’s navigation, guidance, and control system,” IFAC Proc. Volumes, vol. 3, no. 1, pp. 275–312, Mar. 1970.

[4] S. Juliana, Q. P. Chu, J. A. Mulder, and T. J. van Baten, “Flight control of atmospheric re-entry vehicle with non-linear dynamic inversion,” in Proc. AIAA Guid., Navigat., Control Conf. Exhibit. Reston, VA, USA: American Institute of Aeronautics and Astronautics, Aug. 2004, p. 5330.

[5] N. M. Wereley, “Analysis and control of linear periodically time varying systems,” Ph.D. dissertation, Dept. Aeron. Astron., Massachusetts Inst. Technol., Boston, MA, USA, Feb. 1991.

[6] D. Ossmann, J. Theis, and P. Seiler, “Load reduction on a clipper liberty wind turbine with linear parameter-varying individual blade pitch control,” Wind Energy, vol. 20, no. 10, pp. 1771–1786, Jun. 2017.

[7] M. J. Balas and Y. J. Lee, “Controller design of linear periodic time-varying systems,” in Proc. Amer. Control Conf., vol. 5, Jun. 1997, pp. 2667–2671.

[8] G. E. Dullerud and K. Glover, “Robust performance of periodic systems,” IEEE Trans. Autom. Control, vol. 41, no. 8, pp. 1146–1159, Aug. 1996.

[9] J. Kim, D. G. Bates, and I. Postlethwaite, “Robustness analysis of linear periodic time-varying systems subject to structured uncertainty,” Syst. Control Lett., vol. 55, no. 9, pp. 719–725, Sep. 2006.

[10] D. Ossmann and H. Pfifer, “Robustness analysis of continuous periodic systems using integral quadratic constraints,” in Proc. IEEE 58th Conf. Decis. Control (CDC), Dec. 2019, pp. 5805–5810.

[11] U. T. Jönsson, “Robustness of trajectories with finite time extent,” IFAC Proc. Volumes, vol. 34, no. 6, pp. 1077–1082, Jul. 2001.

[12] M. Cantoni and G. Vinnicombe, “Linear feedback systems and the graph topology,” IEEE Trans. Autom. Control, vol. 47, no. 5, pp. 710–719, May 2002.

[13] M. Cantoni and H. Pfifer, “Gap metric computation for time-varying systems on finite horizons,” IFAC-PapersOnLine, vol. 50, no. 1, pp. 14513–14518, Jul. 2017.

[14] M. Green and D. J. N. Limebeer, Linear Robust Control. Upper Saddle River, NJ, USA: Prentice-Hall, 1995.

[15] A. Megretski and A. Rantzer, “System analysis via integral quadratic constraints,” IEEE Trans. Autom. Control, vol. 42, no. 6, pp. 819–830, Jun. 1997.

[16] J. Veenman, C. W. Scherer, and D. W. Bennett, “A comprehensive analytical tool for control validation of fixed-wing unmanned aircraft,” IEEE Trans. Control Syst. Technol., vol. 28, no. 5, pp. 1785–1801, Sep. 2020.

[17] P. Seiler, “Stability analysis with dissipation inequalities and integral quadratic constraints,” IEEE Trans. Autom. Control, vol. 60, no. 6, pp. 1704–1709, Jun. 2015.

[18] M. Cantoni, U. T. Jönsson, and S. Z. Khong, “Robust stability analysis for feedback interconnections of time-varying linear systems,” SIAM J. Control Optim., vol. 51, no. 1, pp. 353–379, Jan. 2013.

[19] J. Veenman, C. W. Scherer, and H. Köroğlu, “Robust stability and performance analysis based on integral quadratic constraints,” Eur. J. Control, vol. 31, pp. 1–32, Sep. 2016.

[20] H. Pfifer and P. Seiler, “Less conservative robustness analysis of linear parameter-varying systems using integral quadratic constraints,” Int. J. Robust Nonlinear Control, vol. 26, no. 16, pp. 3580–3594, Feb. 2016.