

Learning Quasi-LPV Models and Robust Control Invariant Sets with Reduced Conservativeness

Sampath Kumar Mulagaleti and Alberto Bemporad, *Fellow, IEEE*

Abstract—We present an approach to identify a quasi-Linear Parameter Varying (qLPV) model of a plant, with the qLPV model guaranteed to admit a robust control invariant (RCI) set. It builds upon the concurrent synthesis framework presented in [1], in which the requirement of existence of an RCI set is modeled as a control-oriented regularization. Here, we reduce the conservativeness of the approach by bounding the qLPV system with an uncertain LTI system, which we derive using bound propagation approaches. The resulting regularization function is the optimal value of a nonlinear robust optimization problem that we solve via a differentiable algorithm. We numerically demonstrate the benefits of the proposed approach over two benchmark approaches.

Index Terms—Uncertain systems, Identification for control, Linear parameter-varying systems

I. INTRODUCTION

FOR synthesizing control schemes for nonlinear systems, the first step involves identifying a dynamic model via system identification techniques [2]. Typically, controllers use the predictive capabilities of these models to optimize performance. This approach, however, might prove inadequate in scenarios when the underlying system is subject to constraints, since the predicted model behavior might deviate from the system’s response. Robust identification methods address this by capturing both a nominal behavior and prediction errors [3]–[5]. Yet, decoupling identification from controller synthesis can lead to feasibility issues. This has motivated the development of concurrent synthesis approaches that co-identify an uncertain model and a robust controller [6], [7]. Some reinforcement learning approaches, e.g., [8], can also be interpreted in such a framework.

For control-oriented identification, an effective model class is quasi-Linear Parameter Varying (qLPV) systems [9]. In qLPV systems, a.k.a. self-scheduled LPV systems, the dynamics are described by linear models that change over time as a function of a scheduling vector, whose values are generated by a nonlinear function of the model state. This has led to extensive research on identifying such models [10], [11]. Recently, in [1] we introduced a concurrent synthesis framework that guarantees the existence of a robust control invariant (RCI) set [12] for the identified qLPV model based on a softmax scheduling function. This in turn ensures that a feedback controller exists for the plant generating the data.

The authors are with the IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100, Lucca, Italy (s.mulagaleti, alberto.bemporad@imtlucca.it). This work was supported by the European Research Council (ERC), Advanced Research Grant COMPACT (Grant Agreement No. 101141351).

The RCI set was parameterized as a configuration-constrained polytope [13], which provides a convenient representation for developing such frameworks. At the core of the concurrent synthesis framework introduced in [1] is a control-oriented regularization function. This function is defined as the value function of a convex quadratic program (QP) and quantifies the size of the largest robust control invariant (RCI) set obtainable for a given parameterization. In fact, any model for which this regularization function admits a finite value becomes a candidate for controller synthesis. By exploiting the softmax parameterization, invariance is achieved by encapsulating the qLPV model within a linear time-invariant (LTI) model with multiplicative uncertainty.

Contribution: This paper improves the approach of [1], by identifying models that admit RCI sets with reduced conservativeness. We develop a new control-oriented regularization based on an uncertain linear system that encapsulates the qLPV system, and is tighter than one used in [1]. The resulting regularization is optimal value of a nonlinear robust optimization problem, which is solved using a differentiable algorithm. The algorithm is then embedded into the concurrent synthesis framework. Through a numerical example, we demonstrate the benefits of the proposed approach. The paper is structured as follows: Section II formalizes the concurrent synthesis problem, and recalls the approach of [1] to formulate the regularization function; Section III presents the novel regularization function formulation, along with the differentiable algorithm to evaluate it; Section IV presents an algorithm to solve the concurrent synthesis problem; Section V presents a numerical example, and validates it against benchmark approaches; Section VI summarizes the contribution and discusses future research directions.

Notation: The set $\mathbb{I}_a^b := \{a, \dots, b\}$ indicates indices between a and b . Given sets $\mathcal{A}, \mathcal{B} \subseteq \mathbb{R}^n$, $\mathcal{A} \oplus \mathcal{B}$ and $\mathcal{A} \ominus \mathcal{B}$ denote their Minkowski sum and difference respectively, with the sum denoted as $a \oplus \mathcal{B}$ if $\mathcal{A} = \{a\}$ is a singleton. Given $a \in \mathbb{R}^n$, $|a| \in \mathbb{R}^n$ denotes the element-wise absolute value vector, and the function $\text{softmax} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is defined with components $e^{a_i} / \sum_{j=1}^n e^{a_j}$ for $i \in \mathbb{I}_1^n$. The set $\text{CH}\{x_i, i \in \mathbb{I}_1^m\}$ is the convex hull of vectors x_1, \dots, x_m . The symbol \otimes denotes the Kronecker product.

II. PROBLEM SETUP

We have dataset $\mathcal{D} := \{(u_t, y_t), t \in \mathbb{I}_0^{N-1}\}$ of input-output measurements from the nonlinear plant

$$\mathbf{z}^+ = \mathbf{f}(\mathbf{z}, u), \quad y = \mathbf{g}(\mathbf{z}), \quad (1)$$

where $u \in \mathbb{R}^{n_u}, y \in \mathbb{R}^{n_y}, z \in \mathbb{R}^{n_z}$ are the input, output, and state vectors. Assuming the functions \mathbf{f} and \mathbf{g} , and state dimension n_z , are unknown, we tackle the following problem:

Problem 1: Identify a model to predict the behaviour of (1) using \mathcal{D} , while ensuring that the model can be used to synthesize a feedback controller to regulate the plant output inside a set Y , i.e., $y \in \mathbb{Y}$, using control inputs $u \in \mathbb{U}$.

While Problem 1 can be tackled stage-wise, i.e., system identification with uncertainty characterization followed by robust controller synthesis, it is possible that the latter may be suboptimal/infeasible. We avoid this using a concurrent synthesis framework which integrates both phases.

A. Concurrent synthesis framework

Consider the qLPV model

$$x^+ = A(p(x))x + B(p(x))u, \quad \hat{y} = Cx \quad (2)$$

of (1), where $x \in \mathbb{R}^{n_x}$ is the model state, and the matrix-valued functions $A(p)$ and $B(p)$ are parameterized as $A(p) := \sum_{i=1}^{n_p} p_i A_i$ and $B(p) := \sum_{i=1}^{n_p} p_i B_i$ respectively. The scheduling function $p: \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_p}$ is defined as

$$p(x; \theta) := \text{softmax}([\mathcal{N}(x; \theta_1), \dots, \mathcal{N}(x; \theta_{n_p})]^T), \quad (3)$$

where each $\mathcal{N}(x; \theta_i)$ is a feedforward neural network (FNN) whose weights and biases are collected in the vector θ_i . The parameterization in (3) enforces p to belong to the simplex

$$\mathcal{P} := \left\{ p \mid \sum_{i=1}^{n_p} p_i = 1, 0 \leq p \leq 1 \right\}.$$

Remark 1: The results in the sequel can be extended to parameterizations $p(x, u)$, i.e., with dependency also on u .

The system identification problem computes model parameters $\mathbf{A} := (A_1, \dots, A_{n_p})$, $\mathbf{B} := (B_1, \dots, B_{n_p})$, C , and $\theta := (\theta_1, \dots, \theta_{n_p})$ by solving the optimization problem

$$\min_{\mathbf{A}, \mathbf{B}, C, \theta, x_0} \frac{1}{N} \sum_{t=0}^{N-1} \|y_t - Cx_t\|_2^2 \quad (4)$$

$$\text{s.t. } x_{t+1} = A(p(x_t, \theta))x_t + B(p(x_t, \theta))u_t, \quad t \in \mathbb{I}_0^{N-1},$$

in which also the initial state is optimized. Unfortunately, a constrained controller for a model obtained from (4) is not guaranteed to exist, since the output \hat{y} of (2) might not match the plant output y exactly. We ameliorate this using a *control-oriented regularization* [1] based on the state-observer model

$$z^+ = A(p(z))z + B(p(z))u + L(p(z))w, \quad (5)$$

where we parameterize the disturbance as $w := y - Cz$ and the observer gain as $L(p) := \sum_{i=1}^{n_p} p_i L_i$. We now recall [1, Prop. 1, Prop. 2] to derive an uncertain model based on (5).

Proposition 1: (i) Suppose that the behavior of system (1) is described by the model

$$\hat{x}^+ = A(p(\hat{x}))\hat{x} + B(p(\hat{x}))u \quad y \in C\hat{x} + \mathbb{V}, \quad (6)$$

for some $\mathbb{V} \subset \mathbb{R}^{n_y}$, and there exists some set $\mathcal{E} \subseteq \mathbb{R}^{n_x}$ that satisfies $\hat{x}_0 - z_0 \in \mathcal{E} \Rightarrow \hat{x}_t - z_t \in \mathcal{E}$ for all $t > 0$ when (5) and (6) are excited by the same inputs. Defining

$$\mathbb{W} := C\mathcal{E} \oplus \mathbb{V}, \quad (7)$$

and denoting the set of states reached by (5) at time t from some z_0 for all possible $w \in \mathbb{W}$ sequences and given input sequence as Z_t , it follows that $\hat{x}_0 - z_0 \in \mathcal{E} \Rightarrow y_t \in CZ_t \oplus \mathbb{W}$ for all $t > 0$ for all $v \in \mathbb{V}$ sequences, such that

$$z^+ \in A(p(z))z + B(p(z))u \oplus L(p(z))\mathbb{W}, \quad y \in Cz \oplus \mathbb{W} \quad (8)$$

is a valid uncertain model of (6); (ii) Suppose there exists a set $X \subseteq \mathbb{R}^{n_x}$ and control law $\mu_c: X \rightarrow \mathbb{U}$ verifying

$$A(p(z))z + B(p(z))\mu_c(z) \oplus L(p(z))\mathbb{W} \subseteq X, \quad \forall z \in X, \quad (9a)$$

$$CX \oplus \mathbb{W} \subseteq \mathbb{V}. \quad (9b)$$

Then from any $z_0 \in X$ and $\hat{x}_0 \in z_0 \oplus \mathcal{E}$, the control input $u_t = \mu_c(z_t)$ ensures $z_t \in X$ and $y_t \in \mathbb{Y}$ for all $t \geq 0$. \square

The first result in Proposition 1 states that the uncertain system (8) can be *forward simulated* to bound the plant output if (7) holds. This is a standard robust control approach using the observer invariant set \mathcal{E} [14]. Through a suitable choice of \mathbb{V} , any system (1) (potentially also including bounded noise) can be represented as (6). The second result states that if the state of (8) can be persistently maintained inside X using a control law μ_c , then it can be used to regulate the underlying plant output inside Y . As per (9), X is an RCI set for (8). In the sequel, we modify Problem (4) to (i) Compute observer gains $\mathbf{L} := (L_1, \dots, L_{n_p})$, and (ii) Guarantee the existence of a set X verifying (9). Denoting $\Theta := (\mathbf{A}, \mathbf{B}, C, \mathbf{L}, \theta)$, we formulate the modified problem as

$$\min_{\Theta, x_0} \frac{1}{N} \sum_{t=0}^{N-1} \|y_t - Cx_t\|_2^2 + \tau \mathbf{r}(\Theta) \quad (10)$$

$$\text{s.t. } x_{t+1} = A(p(x_t, \theta))x_t + B(p(x_t, \theta))u_t, \quad t \in \mathbb{I}_0^{N-1},$$

where the *control-oriented regularization* $\mathbf{r}(\Theta)$ is such that

$$\mathbf{r}(\Theta) < \infty \Rightarrow \exists X \subseteq \mathbb{R}^{n_x} \text{ verifying (9),} \quad (11a)$$

$$\text{Small } \mathbf{r}(\Theta) \Rightarrow \exists \text{ Large } X \subseteq \mathbb{R}^{n_x} \text{ verifying (9),} \quad (11b)$$

and $\tau \geq 0$ is the regularization constant. While the requirement (11b) is informally stated, we will formalize it in the sequel. The goal of Problem (10) is to identify a model Θ , while also maximizing the size of the corresponding RCI set. In this context, we say a model Θ_1 is *less conservative* than Θ_2 if they admit RCI sets satisfying $\mathbf{r}(\Theta_1) < \mathbf{r}(\Theta_2)$.

B. Control-oriented regularization

We now recall the approach of [1] which formulates $\mathbf{r}(\Theta)$ as inverse of the size of the largest RCI set admitted by Θ . This formulation requires: (i) a characterization of the disturbance set \mathbb{W} ; (ii) a description of the RCI set X ; and (iii) a suitable measure to quantify the size of an RCI set.

1) *Characterization of \mathbb{W} :* We utilize a simple data-driven characterization of \mathbb{W} , since the sets \mathcal{E} and \mathbb{V} required to compute it as (7) are unknown. Assuming access to a dataset $\mathcal{D}^w := \{(y_t, u_t), t \in \mathbb{I}_0^{N^w-1}\}$ from (1), we simulate (5) from the origin using \mathcal{D}^w (with $w_t = y_t - Cz_t$), and denote the resulting state sequence as $\{z_t^w, t \in \mathbb{I}_0^{N^w}\}$. Defining the sampled disturbances as $\mathcal{W} := \{y_t - Cz_t^w \mid t \in \mathbb{I}_0^{N^w-1}\}$, we denote $\bar{w} := \max_{w \in \mathcal{W}} w$ and $\underline{w} := \min_{w \in \mathcal{W}} w$, along with

$c_w := 0.5(\bar{w} + \underline{w})$ and $\epsilon_w := \frac{1}{2}(\bar{w} - \underline{w})$. Then, we characterize the set \mathbb{W} for given Θ as

$$\mathbb{W} := \{w \mid |w - c_w| \leq \kappa \epsilon_w\}, \quad (12)$$

where $\kappa > 0$ is a user-specified inflation parameter to account for finite data. Note that \mathbb{W} is an inflated bounding box of the sampled disturbances \mathcal{W} built using the dataset \mathcal{D}^w . We refer to [1, Prop. 4] for lower-bounds on $\kappa > 1$ to verify (7).

2) *Characterization of X*: We work with polytopic RCI sets X parameterized with given matrix $F \in \mathbb{R}^{f \times n_x}$ as

$$X \leftarrow X(q) := \{x \mid Fx \leq q\},$$

and enforce configuration-constraints \mathcal{C} [13] which dictate

$$q \in \mathcal{C} := \{q \mid Eq \leq 0\} \Rightarrow X(q) = \text{CH}\{V_j q, j \in \mathbb{I}_1^y\},$$

where $V_j \in \mathbb{R}^{n_x \times f}$ are vertex maps. We refer to [13] for details about computing E and $V := (V_1, \dots, V_v)$ given F . To enforce the RCI constraints in (9), the approach of [1] exploits the parameterization of the scheduling function in (3), which implies $p(z) \in \mathcal{P}$ for all $z \in \mathbb{R}^{n_x}$ such that

$$(A(p), B(p), L(p)) \in \Delta := \text{CH}\{(A_i, B_i, L_i), i \in \mathbb{I}_1^{n_p}\} \quad (13)$$

for all p . Then, an RCI set for the uncertain linear system

$$z^+ = Az + Bu + Lw, \quad (A, B, L) \in \Delta, w \in \mathbb{W} \quad (14)$$

is RCI for uncertain system (8). To enforce that $X(q)$ is an RCI set for (14), we use the following result from [13], where we denote $U_j := e_j^\top \otimes \mathbb{I}_{n_u}$ for $j \in \mathbb{I}_1^v$, and assume that the output constraint set $\mathbb{Y} := \{y \mid H^y y \leq h^y\}$.

Proposition 2: The set $X(q)$ is an RCI set for (14) with constraints $Cz \oplus \mathbb{W} \subseteq \mathbb{Y}$ and $u \in \mathbb{U}$ if there exists some $v \in \mathbb{R}^{v \cdot n_u}$ such that $(q, v) \in \mathbb{S}$, where we define

$$\mathbb{S} := \left\{ \begin{pmatrix} q \\ v \end{pmatrix} \mid \begin{array}{l} \forall (i, j) \in \mathbb{I}_1^{n_p} \times \mathbb{I}_1^v, Eq \leq 0, U_j v \in \mathbb{U}, \\ F(A_i V_j q + B_i U_j v) + d_i \leq q, \\ H^y (C V_j q + c_w) + \kappa |H^y| \epsilon_w \leq h^y \end{array} \right\}, \quad (15)$$

with $d_i := F L_i c_w + \kappa |F L_i| \epsilon_w$ for all $i \in \mathbb{I}_1^{n_p}$. \square

Essentially, Proposition 2 together with (13) state that

$$\exists v : (q, v) \in \mathbb{S} \Rightarrow X(q) \text{ satisfies (9)}, \quad (16)$$

with $\mu_c(z)$ being a vertex control law defined by v [13].

3) *Size of the RCI set*: We define the size of an RCI set based on the ability of the system to perform safe output tracking. Denoting the vertices of \mathbb{Y} as $\{y_k, k \in \mathbb{I}_1^{v_y}\}$ and the mean values of the matrices $(A_i, B_i), i \in \mathbb{I}_1^{n_p}$ as (\bar{A}, \bar{B}) , the size of $X(q)$ is modeled as

$$\begin{aligned} d(\bar{A}, \bar{B}, C, q) &:= \min_{z, u} \sum_{k=1}^{v_y} \sum_{t=1}^M \|y_k - C z_t^k\|_2^2 \quad (17) \\ \text{s.t. } & z_0^k = 0, z_{t+1}^k = \bar{A} z_t^k + \bar{B} u_t^k, u_t^k \in \mathbb{U}, \\ & F z_t^k \leq q, \forall (k, t) \in \mathbb{I}_1^{v_y} \times \mathbb{I}_0^{M-1}, \end{aligned}$$

where (z, u) denote the state and input trajectories. The value $d(\bar{A}, \bar{B}, C, q)$ captures how close the output $y = Cz$ of the nominal system $z^+ = \bar{A}z + \bar{B}u$ can be driven from the origin

to the vertices of \mathbb{Y} in M -steps while belonging inside $X(q)$. Note that for any feasible vectors q_1 and q_2 , the inequality

$$X(q_1) \subseteq X(q_2) \Rightarrow d(\bar{A}, \bar{B}, C, q_2) \leq d(\bar{A}, \bar{B}, C, q_1) \quad (18)$$

holds since the optimizers of the latter problem are feasible for the former. Using the ingredients in (12), (15) and (17), the approach of [1] models the control-oriented regularization function $r(\Theta)$ in Problem (10) as the value of the QP

$$r(\Theta) := \inf_{(q, v) \in \mathbb{S}} d(\bar{A}, \bar{B}, C, q). \quad (19)$$

As per (16) and (18), $r(\Theta)$ satisfies (11), making it a suitable control-oriented regularization. However, an RCI set $X(q)$ obtained by solving (19) might be unnecessarily *small*, because the uncertain LTI system (14) encapsulating (8) might be too conservative. In the next section, we derive an uncertain LTI approximation that is less conservative, and formulate a corresponding control-oriented regularization.

Remark 2: The choice of function d should reflect the goals of the control design procedure, e.g., (i) regulation around an output setpoint, with d modeled to minimize the volume of the RCI set about that point [15]; (ii) tuned tube-based model predictive control schemes [16] for stabilization, etc. Future research can focus on deriving such formulations.

III. CONTROL-ORIENTED REGULARIZATION WITH REDUCED CONSERVATIVENESS

While the uncertain LTI system (14) encapsulates the non-linear system (8), it is possible that the scheduling function $p(z)$ with $z \in X(q)$ does not cover the entire set \mathcal{P} , such that $(A(p(z)), B(p(z)), L(p(z)))$ only evolves in some $\tilde{\Delta} \subseteq \Delta$. We exploit this observation to derive a tightened multiplicative uncertainty $\tilde{\Delta}$, and present an approach to compute an RCI set for (14) with Δ replaced by $\tilde{\Delta}$.

A. RCI sets with reduced conservativeness

We now present new conditions for a set $X(q)$ to be an RCI set for (8), which are based on an LTI system with tightened multiplicative uncertainty $\tilde{\Delta} \subseteq \Delta$.

Proposition 3: Suppose the scheduling variable satisfies

$$0 \leq a_i \leq p_i(z), \quad \forall z \in X(q), i \in \mathbb{I}_1^{n_p} \quad (20)$$

for given $q \in \mathcal{C}$. If $X(q)$ is RCI for the uncertain LTI system

$$z^+ = Az + Bu + Lw, \quad (A, B, L) \in \tilde{\Delta}, w \in \mathbb{W}, \quad (21)$$

where $\tilde{\Delta} := \text{CH}\{(\tilde{A}_i, \tilde{B}_i, \tilde{L}_i), i \in \mathbb{I}_1^{n_p}\}$ is defined with

$$(\tilde{A}_i, \tilde{B}_i, \tilde{L}_i) := \left(1 - \sum_{j=1}^{n_p} a_j \right) (A_i, B_i, L_i) + \sum_{j=1}^{n_p} a_j (A_j, B_j, L_j) \quad (22)$$

then $X(q)$ verifies the inclusion in (9).

Proof: The proof follows if for all $z \in X(q)$, we have $(A(p(z)), B(p(z)), L(p(z))) \in \tilde{\Delta}$. To show this, we define

$$\tilde{p}_i(z) = \frac{p_i(z) - a_i}{1 - \sum_{j=1}^{n_p} a_j}, \quad \forall i \in \mathbb{I}_1^{n_p}.$$

Since $\sum_{j=1}^{n_p} a_j \leq \sum_{j=1}^{n_p} p_j(z) \leq 1$ holds, we have $\tilde{p}(z) \in \mathcal{P}$ if $z \in X(q)$. Then, $A(p(z))$ can be written for $z \in X(q)$ as

$$\begin{aligned} A(p(z)) &= \sum_{i=1}^{n_p} \left(\tilde{p}_i(z) + a_i - \tilde{p}_i(z) \sum_{j=1}^{n_p} a_j \right) A_i \\ &= \sum_{i=1}^{n_p} \left(\tilde{p}_i(z) + a_i \sum_{j=1}^{n_p} \tilde{p}_j(z) - \sum_{j=1}^{n_p} a_j \tilde{p}_i(z) \right) A_i \\ &= \sum_{i=1}^{n_p} \left(1 - \sum_{j=1}^{n_p} a_j \right) \tilde{p}_i(z) A_i + \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} a_j \tilde{p}_i(z) A_i = \sum_{i=1}^{n_p} \tilde{p}_i(z) \tilde{A}_i, \end{aligned}$$

where the second equality follows from $\tilde{p}(z) \in \mathcal{P}$, and the third by interchanging the summation order. Hence, we have that $(A(p(z)), B(p(z)), L(p(z))) \in \tilde{\Delta}$ if $z \in X(q)$. ■

The next result derives a set similar to \mathbb{S} for (21).

Corollary 1: The set $X(q)$ is an RCI set for (21) with constraints $\mathcal{C}z \oplus \mathbb{W} \subseteq \mathbb{Y}$ and $u \in \mathbb{U}$ if there exists some $v \in \mathbb{R}^{v \cdot n_u}$ such that $(q, v) \in \tilde{\mathbb{S}}$, where we define

$$\tilde{\mathbb{S}} := \left\{ \begin{pmatrix} q \\ v \end{pmatrix} \left| \begin{array}{l} \forall (i, j) \in \mathbb{I}_1^{n_p} \times \mathbb{I}_1^1, \exists a \in [0, p(z)], \forall z \in X(q), \\ F(\tilde{A}_i(a)V_jq + \tilde{B}_i(a)U_jv) + \tilde{d}_i(a) \leq q, \\ H^y(CV_jq + c_w) + \kappa|H^y|e^w \leq h^y, \\ U_jv \in \mathbb{U}, Eq \leq 0 \end{array} \right. \right\},$$

with $\tilde{d}_i(a) := F\tilde{L}_i(a)c_w + \kappa|F\tilde{L}_i(a)|\epsilon_w$ for $i \in \mathbb{I}_1^{n_p}$, and the matrices $(\tilde{A}_i, \tilde{B}_i, \tilde{L}_i)$ depend on a_i through (22). □

Proof: The proof follows from Proposition 2, after observing that \mathbb{S} is RCI for (21) for any a verifying (20), with (A_i, B_i, L_i) replaced by $(\tilde{A}_i(a), \tilde{B}_i(a), \tilde{L}_i(a))$. ■

Using $\tilde{\mathbb{S}}$, we formulate the regularization function as

$$r(\Theta) := \inf_{(q,v) \in \tilde{\mathbb{S}}} d(\bar{A}, \bar{B}, C, q), \quad (23)$$

where (\bar{A}, \bar{B}) are mean values of $(\tilde{A}_i(a), \tilde{B}_i(a))$, $i \in \mathbb{I}_1^{n_p}$. We always have $\mathbb{S} \subseteq \tilde{\mathbb{S}}$, since $\tilde{\mathbb{S}}$ formulated with feasible value $a = 0$ equals \mathbb{S} . Hence, (23) is less conservative than (19). Unfortunately, $\tilde{\mathbb{S}}$ is no longer a polytope since it includes the robust nonlinear constraint (20), such that Problem (23) is a nonlinear robust optimization problem instead of a QP. We now develop an algorithm to solve Problem (23).

B. Solving Problem (23)

We use the following observation to solve Problem (23): Given sets X_1 and X_2 such that $X_1 \subseteq X_2$, if $p_i(z) \geq a_i$ holds for all $z \in X_2$, then $p_i(z) \geq a_i$ follows for all $z \in X_1$. To exploit this observation, given $q \in \mathcal{C}$ we define the set

$$\mathbb{B}(q) := \{x : |x_i - \mu_i(q)| \leq \sigma_i(q) + \zeta, i \in \mathbb{I}_1^{n_x}\}, \quad (24)$$

where we define $\mu_i(q) := \frac{1}{2}(\max_j V_{j,i}q + \min_j V_{j,i}q)$ and $\sigma_i(q) := \frac{1}{2}(\max_j V_{j,i}q - \min_j V_{j,i}q)$, with $V_{j,i}$ denoting row i of V_j , and $\zeta > 0$. The set $\mathbb{B}(q)$ is a bounding box of $X(q)$ inflated by $\zeta > 0$. Then, we define the polytope

$$\hat{\mathbb{S}}(\tilde{q}, a) := \left\{ \begin{pmatrix} q \\ v \end{pmatrix} \left| \begin{array}{l} \forall (i, j) \in \mathbb{I}_1^{n_p} \times \mathbb{I}_1^1, X(q) \subseteq \mathbb{B}(\tilde{q}), \\ F(\tilde{A}_i(a)V_jq + \tilde{B}_i(a)U_jv) + \tilde{d}_i(a) \leq q, \\ H^y(CV_jq + c_w) + \kappa|H^y|e^w \leq h^y, \\ U_jv \in \mathbb{U}, Eq \leq 0 \end{array} \right. \right\},$$

where the parameters $a \in \mathbb{R}^{n_p}$ depend on \tilde{q} as

$$0 \leq a_i \leq p_i(z), \quad \forall z \in \mathbb{B}(\tilde{q}), i \in \mathbb{I}_1^{n_p}. \quad (25)$$

Algorithm 1 Evaluate $r(\Theta; q_0)$

Require: $\Theta, (c_w, \epsilon_w), q_0 \in \mathcal{C}, \hat{k} > 0$

- 1: **For** $k = 0, 1, 2, \dots, \hat{k} - 1$ **do**
 - 2: $a \leftarrow \text{BoundProp}(q_k, \Theta)$, evaluate $(\tilde{A}_i(a), \tilde{B}_i(a), \tilde{L}_i(a))$
 - 3: $q_{k+1}, v_{k+1} \leftarrow \arg \inf_{(q,v) \in \hat{\mathbb{S}}(q_k, a)} d(\bar{A}, \bar{B}, C, q)$
 - 4: $r_{k+1} \leftarrow d(\bar{A}, \bar{B}, C, q_{k+1})$
 - 5: **return** $r_{\hat{k}}, q_{\hat{k}}$
-

The polytope $\hat{\mathbb{S}}(\tilde{q}, a)$ is formulated by replacing condition (20) over a in \mathbb{S} with $X(q) \subseteq \mathbb{B}(\tilde{q})$. Since (25) implies $a \in [0, p(z)]$ for all $z \in X(q)$, it follows that $\hat{\mathbb{S}}(\tilde{q}, a) \subseteq \mathbb{S}$. If \tilde{q} is high enough such that the constraint $X(q) \subseteq \mathbb{B}(\tilde{q})$ is inactive for all feasible q , the inclusions $\mathbb{S} = \hat{\mathbb{S}}(\tilde{q}, 0) \subseteq \hat{\mathbb{S}}(\tilde{q}, a) \subseteq \mathbb{S}$ follow for any a verifying (25), which implies that an RCI set computed by optimizing over $\hat{\mathbb{S}}(\tilde{q}, a)$ is less conservative than that obtained by optimizing over \mathbb{S} . Utilizing $\hat{\mathbb{S}}(\tilde{q}, a)$, we formulate Algorithm 1 to solve Problem (23). Starting from some $q_0 \in \mathcal{C}$, we compute bounds a verifying (25) using a BoundProp methodology presented next. Then, we compute $(\tilde{A}_i(a), \tilde{B}_i(a), \tilde{L}_i(a))$ used to define a QP in Step 3 that optimizes over $\hat{\mathbb{S}}(q_k, a)$ constructed using $\mathbb{B}(q_k)$. Then, updating q_k to q_{k+1} , we repeat the procedure for \hat{k} number of steps. The key feature of Algorithm 1 is that the output $r_{\hat{k}}$ is differentiable in Θ , such that it can be plugged into a gradient-based solver to tackle Problem (10). Future research can study the theoretical properties of this algorithm.

C. Interval bound propagation

We use interval bound propagation (IBP) [17] to compute a verifying (25) in Step 3 of Algorithm 1, assuming that the FNNs $\mathcal{N}(\cdot; \theta_i)$ are defined using monotonic activations. While IBP computes a exploiting this monotonicity, alternative verification approaches can be used to handle nonmonotonic activations, see, e.g., [18]–[20]. However, they might be computationally expensive. The IBP approach, while limited to monotonic activations, is computationally inexpensive. It is formed by observing that the interval $z \in [\underline{z}, \bar{z}]$ projected through the linear map $h(z) = Wz + b$ results in $h(z) \in [W\mu + b - |W|\Sigma, W\mu + b + |W|\Sigma]$ with $\mu = 0.5(\bar{z} + \underline{z})$ and $\Sigma = 0.5(\bar{z} - \underline{z})$, while through a monotonically increasing nonlinear map $h(z)$ results in $h(z) \in [h(\underline{z}), h(\bar{z})]$. Defining $\mathbf{N}(z; \theta_i) := e^{\mathcal{N}(z; \theta_i)}$, a composition of the propagations can be used to compute the bounds

$$\mathbf{N}(z; \theta_i) \in [\underline{\mathbf{N}}_i, \bar{\mathbf{N}}_i], \quad \forall z \in \mathbb{B}(q). \quad (26)$$

Proposition 4: Given $q \in \mathcal{C}$, suppose (26) holds. Then,

$$a_i := \frac{\underline{\mathbf{N}}_i}{\left(\underline{\mathbf{N}}_i + \sum_{j \in \mathbb{I}_1^{n_p} \setminus i} \bar{\mathbf{N}}_j \right)}$$

verifies (25) with $\tilde{q} = q$ for all $i \in \mathbb{I}_1^{n_p}$.

Proof: We observe that (26) satisfies the inequalities $0 \leq \underline{\mathbf{N}}_i \leq \bar{\mathbf{N}}_i$, and the optimizer of $\min_{x,y} \frac{x}{x+y}$ s.t. $(x, y) \in [\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}]$ with $0 \leq \underline{x} \leq \bar{x}$ and $0 \leq \underline{y} \leq \bar{y}$ is (\underline{x}, \bar{y}) . ■

Algorithm 2 Solve Problem (10)

Require: $(\mathcal{D}, \mathcal{D}^w)$, (F, V, E) , $\kappa > 1$, $\zeta > 0$, $\hat{k}, \hat{l} > 0$, Initial parameters $\Theta_0, x_{0,0}, q_0 \in \mathcal{C}$.

- 1: **For** $l = 0, 1, 2, \dots, \hat{l} - 1$ **do**
 - 2: $\Theta_{l+1}, x_{0,l+1} \leftarrow \text{Optimizer}(\nabla \mathcal{J}(\Theta_l, x_{0,l}; q_l))$
 - 3: $r_{l+1}, q_{l+1} \leftarrow r(\Theta_{l+1}; q_l)$
 - 4: **return** Θ_l
-

IV. CONCURRENT SYNTHESIS ALGORITHM

We now develop Algorithm 2 to solve Problem (10), in which we utilize the fact that Algorithm 1 is parametric in q_0 . Towards its development, for a given $\tilde{q} \in \mathcal{C}$, we define

$$\mathcal{J}(\Theta, x_0; \tilde{q}) := \frac{1}{N} \sum_{t=0}^{N-1} \|y_t - Cx_t\|_2^2 + \tau r(\Theta; \tilde{q}),$$

where $r(\Theta; \tilde{q})$ is evaluated using Algorithm 1. We use this function as an alias for the objective of Problem (10). We initialize the algorithm with $q_0 \in \mathcal{C}$ being the optimizer of Problem (19) with $\Theta = \Theta_0$. In Step 3, with the last updated RCI set parameter q_l , we compute the gradients of $\mathcal{J}(\Theta, x_0; q_l)$, that we pass to an optimizer such as Adam [21] to compute Θ_{l+1} and $x_{0,l+1}$. Then, we update q_{l+1} using the updated model parameters Θ_{l+1} starting from q_l using Algorithm 1. To compute the initial Θ_0 and configuration triplet (F, V, E) , we use the approach in [1, Section IV].

V. NUMERICAL EXAMPLE

We consider data from an oscillator with dynamics given by $1.5\ddot{y} + \dot{y} + y + 1000y^3 = u$, with input $u \in [-0.5, 0.5][N]$ the force applied, and output y [m] the position. While we focus on the benefits of using the approach of this paper for concurrent synthesis, we refer to [1] for insights regarding the qLPV parameterization. We implement Algorithm 2 using `jax` [22], and utilize the differentiable QP solver `qpax` [23] to implement of Algorithm 1. We evaluate the quality of a model using the Best Fit Rate (BFR) score [24, Section 3.3]¹.

1) **Concurrent identification:** Using randomly sampled inputs in $[-0.5, 0.5]$, we build the training dataset \mathcal{D} with 10000 points, disturbance dataset \mathcal{D}^w with 2000 points, and test dataset with 10000 points sampled at 0.1s time intervals. We parameterize (2) with $n_x = 2$, $n_p = 6$, and a single-hidden-layer FNN $\mathcal{N}(x; \theta_i)$ with 3 monotonic activation units $\text{elu}(x) + 1$. We compute an initial qLPV model parameterized as (2) using the `jax-sysid` [24] toolbox on \mathcal{D} , which achieves BFR scores of 85.887 on \mathcal{D} , 85.97038 on \mathcal{D}^w , and 86.8900 on the test set. We use this model to define Θ_0 , in which we set each observer gain $L_i = 0$. Using [1, Proposition 5], we compute a feasible triplet (F, V, E) with $f = v = 4$ for $X(q)$. We select $M = 5$ in (17), $\tau = 0.0005$ in (10), $\kappa = 1.01$ in (12), and compute q_0 by solving Problem (19) to initialize Algorithm 2. We consider the following benchmarks: (i) **Sequential model-based synthesis:** We compute the parameters of (2) starting from Θ_0 using the `jax-sysid` toolbox, then utilize \mathcal{D}^w to identify \mathbb{W} and

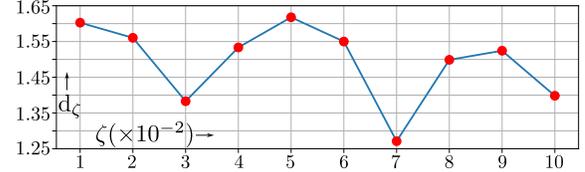


Fig. 1: Output of Algorithm 1 for different ζ , simulated using models Θ_ζ identified by Algorithm 2. Observe that for chosen ζ , we obtain d_ζ significantly lesser than $d_{\text{seq}} = 22.6924$ and $d_{\text{base}} = 5.0936$.

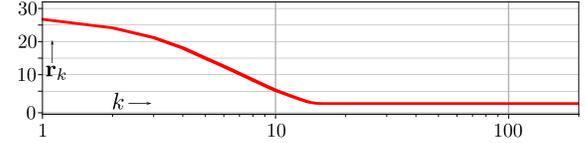


Fig. 2: Iterations of Algorithm 1 for $\Theta = \Theta_{0.07}$ and $\zeta = 0.07$.

compute the maximal RCI set [25] for (14) inside the tightened output constraint set $Cx \in \mathbb{Y} \ominus \mathbb{W}$. Then, selecting (F, q) such that $\{x \mid Fx \leq q\}$ is the maximal RCI set, we evaluate $d_{\text{seq}} := d(\overline{A}, \overline{B}, C, q)$ defined in (17). The identified model achieves BFR scores of 90.7979 on \mathcal{D} , 92.3137 on \mathcal{D}^w , and 90.8584 on the test set, along with $d_{\text{seq}} = 22.6924$. (ii) **Baseline concurrent synthesis:** We follow the approach of [1] to formulate the control-oriented regularization as (19). At the solution, we re-solve (19), and denote its optimal value as d_{base} . The identified model achieves BFR scores of 90.6734 on \mathcal{D} , 92.0751 on \mathcal{D}^w , and 90.6593 on the test set, along with $d_{\text{base}} = 5.0936$ indicating the benefits of utilizing concurrent synthesis for reducing RCI set conservativeness.

To compare against these benchmarks, we simulate Algorithm 2 with ζ uniformly spaced in $[0.01, 0.1]$, and fix $\hat{k} = 1$ such that we perform one iteration of Algorithm 1 per iteration of Algorithm 2. We report that while $\hat{k} > 1$ can be chosen to simulate Algorithm 1, it often results in Algorithm 2 converging to suboptimal points. A study of escaping such minima is a subject of future study. Using the model parameters Θ_ζ computed by Algorithm 2 for given ζ , we recompute the largest RCI set by solving Problem (23) with Algorithm 1 with q_0 as the optimizer of Problem (19). We denote the output of Algorithm 1 as $d_\zeta = r_{\hat{k}}$, with $\hat{k} = 200$. We report no reduction in r when Algorithm 1 is applied to the models identified using the benchmark approaches, since these models were not optimized for bound propagation. In Figure 1, we plot d_ζ for different values of ζ . We observe a reduction in conservativeness, indicated by smaller values than d_{seq} and d_{base} , validating our approach to compute models that admit RCI sets with reduced conservativeness. We report that for all models Θ_ζ , we obtain BFR scores in the interval $[90.7303, 90.7371]$ over \mathcal{D} , $[92.4489, 92.4633]$ over \mathcal{D}^w , and $[90.8314, 90.8553]$ over the test set, indicating that a reduction in conservativeness without degrading the model quality. In Figure 2, we show the value of r_k over iterations of Algorithm 1 with $\Theta = \Theta_{0.07}$ and $\zeta = 0.07$, in which we observe monotonic convergence.

2) **Controller synthesis:** We use the parameters $\Theta_{0.07}$ for controller synthesis. While any robust controller that regulates (9) (or (21)) in $X(q_{\hat{k}})$ can be used, we consider a simple

¹Code available on github.com/samku/Con-qLPV

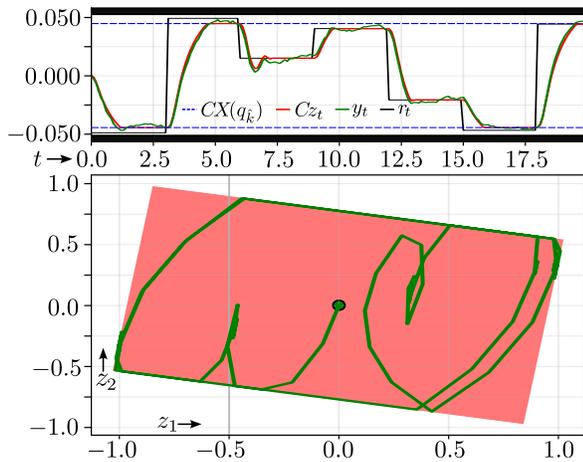


Fig. 3: Closed-loop trajectories using tracking controller (27). The black region in top figure denotes boundaries of Y .

tracking controller formulated as the QP

$$u(z, y, r) = \arg \min_{u \in U} \|Cz^+ - r\|_2^2 \quad (27)$$

$$\text{s.t. } z^+ = A(p(z))z + B(p(z))u + L(p(z))(y - Cz) \in X(q_k),$$

which consumes the current state $z = z_t$ of (5) and current output $y = y_t$ of the plant, and output reference $r = r_t$. For sufficiently large values of $\kappa > 1$ in (12), Problem (27) is recursively feasible. A study of stability properties, which involves the synthesis of ISS-Lyapunov functions to quantify the effect of $w = y - Cz$ on the closed-loop performance, are a subject of future research. Note that (u, z^+) can be penalized for controller tuning. In Figure 3, we plot closed-loop trajectories with piecewise constant references obtained by solving (27). In Figure 3 (top plot), trajectories in the output space are plotted. Observe that the plant output attempts to track the reference signal r_t while satisfying $y_t \in Y$. Also plotted are the model output $Cz_t \in CX(q_k)$. In Figure 3 (bottom plot), the RCI set $X(q_k)$ is plotted, with the state trajectory $z_t \in X(q_k)$ (in green) of the system in (5).

VI. CONCLUSIONS

We extended the concurrent synthesis approach of [1] to identify qLPV models with control synthesis guarantees through the introduction of a novel control-oriented regularization function. In Proposition 3, we derived conditions on existence of RCI sets based on a linear system with tightened multiplicative uncertainty, and in Corollary 1, we derived a new set of configuration-constrained RCI set parameters that we used to formulate the new regularization function. We then developed Algorithm 1 to evaluate the function. Our numerical example demonstrates reduced conservativeness compared to benchmark approaches. Future research will focus on *a*) analyzing Algorithm 1; *b*) efficient approaches to estimate *a*) verifying (25) avoiding monotonicity assumptions; and *c*) using the framework to for real-world systems.

REFERENCES

[1] S. K. Mulagaleti and A. Bemporad, “Combined learning of linear parameter-varying models and robust control invariant sets,” *arXiv preprint arXiv:2411.18166*, 2024.

[2] L. Ljung, C. Andersson, K. Tiels, and T. B. Schön, “Deep learning and system identification,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1175–1181, 2020. 21st IFAC World Congress.

[3] R. L. Kosut, M. K. Lau, and S. P. Boyd, “Set-membership identification of systems with parametric and nonparametric uncertainty,” *IEEE Trans. Autom. Control*, vol. 37, no. 7, pp. 929–941, 1992.

[4] M. Milanese and C. Novara, “Set membership identification of nonlinear systems,” *Automatica*, vol. 40, no. 6, pp. 957–975, 2004.

[5] M. Lauricella and L. Fagiano, “Set membership identification of linear systems with guaranteed simulation accuracy,” *IEEE Trans. Autom. Control*, vol. 65, no. 12, pp. 5189–5204, 2020.

[6] Y. Chen, H. Peng, J. Grizzle, and N. Ozay, “Data-driven computation of minimal robust control invariant set,” in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 4052–4058, 2018.

[7] S. K. Mulagaleti, A. Bemporad, and M. Zanon, “Data-driven synthesis of robust invariant sets and controllers,” *IEEE Control Systems Letters*, vol. 6, pp. 1676–1681, 2022.

[8] M. Zanon and S. Gros, “Safe reinforcement learning using robust MPC,” *IEEE Trans. Autom. Control*, vol. 66, no. 8, pp. 3638–3652, 2021.

[9] M. Lovera, M. Bergamasco, and F. Casella, *LPV Modelling and Identification: An Overview*. Springer Berlin Heidelberg, 2013.

[10] S. Z. Rizvi, J. M. Velni, F. Abbasi, R. Tóth, and N. Meskin, “State-space LPV model identification using kernelized machine learning,” *Automatica*, vol. 88, pp. 38–47, 2018.

[11] C. Verhoek, G. I. Beintema, S. Haesaert, M. Schoukens, and R. Tóth, “Deep-learning-based identification of LPV models for nonlinear systems,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 3274–3280, 2022.

[12] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Springer International Publishing, 2015.

[13] M. E. Villanueva, M. A. Müller, and B. Houska, “Configuration-constrained tube MPC,” *Automatica*, vol. 163, p. 111543, 2024.

[14] D. Mayne, S. Raković, R. Findeisen, and F. Allgöwer, “Robust output feedback model predictive control of constrained linear systems,” *Automatica*, vol. 42, no. 7, pp. 1217–1222, 2006.

[15] P. Trodden, “A one-step approach to computing a polytopic robust positively invariant set,” *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4100–4105, 2016.

[16] F. Badalamenti, S. K. Mulagaleti, A. Bemporad, B. Houska, and M. E. Villanueva, “Configuration-constrained tube MPC for tracking,” *IEEE Control Systems Letters*, vol. 8, pp. 1174–1179, 2024.

[17] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli, “On the effectiveness of interval bound propagation for training verifiably robust models,” *arXiv preprint arXiv:1810.12715*, 2018.

[18] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C.-J. Hsieh, and J. Z. Kolter, “Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification,” in *Advances in Neural Information Processing Systems*, vol. 34, pp. 29909–29921, 2021.

[19] M. Fazlyab, M. Morari, and G. J. Pappas, “Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming,” *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 1–15, 2022.

[20] B. Paulsen and C. Wang, “Linsyn: Synthesizing tight linear bounds for arbitrary neural network activation functions,” in *Tools and Algorithms for the Construction and Analysis of Systems: 28th International Conference, TACAS 2022, (Berlin, Heidelberg)*, p. 357–376, Springer-Verlag, 2022.

[21] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2017.

[22] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, “JAX: composable transformations of Python+NumPy programs,” 2018.

[23] K. Tracy and Z. Manchester, “On the differentiability of the primal-dual interior-point method,” *arXiv preprint arXiv:2406.11749*, 2024.

[24] A. Bemporad, “An L-BFGS-B approach for linear and nonlinear system identification under ℓ_1 and group-lasso regularization,” *IEEE Trans. Autom. Control*, pp. 1–8, 2025.

[25] D. Bertsekas, “Infinite time reachability of state-space regions by using feedback control,” *IEEE Trans. Autom. Control*, vol. 17, no. 5, pp. 604–613, 1972.