Mathematics of Control, Signals, and Systems
https://doi.org/10.1007/s00498-025-00427-7

ORIGINAL ARTICLE

A finite-sample generalization bound for stable LPV systems

Dániel Rácz[1,2] · Martin Gonzalez[3] · Mihály Petreczky[4] · András Benczúr[1,5] · Bálint Daróczy[1]

Received: 22 August 2024 / Accepted: 29 October 2025

Abstract
One of the main theoretical challenges in learning dynamical systems from data is providing upper bounds on the generalization error, that is, the difference between the expected prediction error and the empirical prediction error measured on some finite sample. In machine learning, a popular class of such bounds are the so-called Probably Approximately Correct (PAC) bounds. In this paper, we derive a PAC bound for stable continuous-time linear parameter-varying (LPV) systems. Our bound depends on a weighted H_2-like norm of the chosen class of the LPV systems, but does not depend on the time interval for which the signals are considered.

Keywords PAC bounds · Rademacher complexity · Learning theory · Dynamical systems

1 Introduction

LPV [1] systems are a popular class of dynamical systems in control and system identification, e.g., [1–8]. Generally, LPV systems are linear in state, input and output signals, but the coefficients of these linear relationships depend on the *scheduling variables*. These systems are popular due to their ability to model highly nonlinear phenomena while allowing much simpler theoretical analysis. In this work, we consider *continuous-time* LPV systems in state-space form, where the system matrices are affine functions of the scheduling variables.

✉ Dániel Rácz
racz.daniel@sztaki.hun-ren.hu

1 HUN-REN Institute for Computer Science and Control, Budapest, Hungary

2 Eötvös Loránd University, Budapest, Hungary

3 Institut de Recherche Technologique SystemX, Palaiseau, France

4 Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

5 Research Center of Vehicle Industry, Széchenyi István University, Gyor, Hungary

Published online: 08 April 2026

Context Our main contribution is a PAC generalization bound for stable LPV systems. To put our result more into perspective, consider the problem of learning, or the problem of system identification. The two problems are closely related, in fact, they often overlap. The difference between them lies in some technical assumptions which stem from the particular applications which motivate these two fields. Throughout this paper, these differences are usually not relevant, therefore unless stated otherwise, the term *learning* will refer both to *machine learning* and *system identification*. However, whenever the difference between those two fields is relevant, we will discuss them separately. In order to make the discussion concrete, we will concentrate on LPV models.

In both disciplines, our starting point is a family of parametrized models and the goal is to find a member of this family, which predicts the true output sufficiently accurately for each input and scheduling variable. To this end, we assume that we have a dataset consisting triplets of signals representing the input, the scheduling signal and the corresponding output which are consistent with the underlying physical process we would like to model, all defined on the finite time interval $[0, T]$, and measured before learning. Learning algorithms then choose a concrete element of the parameterized family of LPV systems, for which the prediction error computed on the dataset that is used for learning, called the training data, is small.

To sum it up, in case of learning we choose a model which predicts the training data the best according to some appropriate measure. However, for subsequent use our model should perform well on data which was not used during learning. A widespread measure for the latter is the *true loss*, i.e., the expectation of the prediction error, where the expectation is taken over some distribution of inputs, scheduling signals and outputs, such that the distribution is consistent with the underlying physical process.

Note that even if the distribution of the inputs and scheduling signals is known, the distribution of the outputs remains unknown in the absence of an adequate model for the underlying physical system. That is, the distribution used for computing the true loss is unknown in general, thus in practice the true loss is also unknown.

However, the true loss can be approximated by the *empirical loss*, i.e., the average of prediction errors by a certain model, where the average is taken over the elements of the dataset. In particular, one could use the empirical loss evaluated over some validation or test dataset, as a proxy for the true loss. Furthermore, empirical loss can also be used for learning. In fact, many learning algorithms are based on choosing a member of the parametrized family which minimizes the empirical loss calculated over the training dataset. The intuitive explanation for many learning algorithms lies in the implicit assumption that if the chosen model has a small empirical loss, then its true loss will also be small and it will predict previously unseen data accurately.

It can be shown formally that the empirical loss converges to the true loss as the number of data points grow, e.g., [9, 10]. However, for the purpose of evaluating the true loss based on the empirical loss, it is of interest to have a uniform bound on the difference between the true loss and the empirical loss, with respect to the parameterized family.

Such uniform bounds are referred to as *PAC bounds* in the literature [10, 11] and they are a standard tool for theoretical analysis of statistical learning algorithms.

PAC bounds can be used in several ways [10, 11], without claiming completeness we mention the following applications: (1) to evaluate the performance of identified models based on their performance on a dataset (the training or the validation set); (2) to analyze the effect of various quantities appearing in the bound (norms, VC-dimension, number of data points) on the generalization capability of the learnt models—in particular, characterizing the amount of data necessary for learning adequate models; (3) to select parametrizations which minimize the PAC bound and hence are more likely to allow learning models which generalize well and avoid overfitting.

Contribution In this paper, we derive a particular PAC bound for LPV systems which is based on the Rademacher complexity of the model family.

The derived bound is of the magnitude $O(c/\sqrt{N})$, where c is a constant that depends on the parametrization, and N is the cardinality of an arbitrary finite sample used to evaluate or identify the model. More accurately, the constant c depends on the maximal H_2 norm of the elements of the parametrization. In contrast to related results, the constant c, and thus the bound, does not depend on the length of the time interval $[0, T]$.

Motivation: machine learning There is consensus in the machine learning community on the usefulness of PAC bounds. Continuous-time dynamical systems, e.g., continuous-time Recurrent Neural Networks (RNNs, [12]), Neural Controlled Ordinary Differential Equations (NCDEs, [13]) or structured State-Space Models (SSMs, [14]) are becoming increasingly popular in machine learning. Since LPV systems include bilinear systems [15] as a special case, in principle they could be used as universal approximators for sufficiently smooth dynamical systems [16], including important subclasses of RNNs and NCDEs. Finally, LPV systems include linear state-space models, which are crucial ingredients of SSMs, and subclasses of RNNs, hence PAC bounds for LPV systems are expected to be useful for PAC bounds for NCDEs, RNNs and SSMs.

Motivation: system identification PAC bounds are expected to have the same applications for system identification as for learning, i.e., they will enable a sharper theoretical analysis of system identification algorithms, by making explicit relationship between the empirical and true losses as functions of the number of data points and various properties of the parametrization. In addition, existing theoretical guarantees for learning algorithms of LPV systems from the system identification literature tend to focus on the discrete-time case and they usually provide asymptotic guarantees only [9]. To the best of our knowledge, there are no finite-sample bounds for learning continuous-time LPV systems. The results of this paper represent a first step toward finite-sample bound for LPV systems, despite they are not directly applicable to many classical system identification problems.

Indeed, in order to be able to apply classical PAC bounds based on Rademacher complexity [10], we assume that the data come from several independent experiments and that continuous-time signals are available. While learning from several independently sampled time signals is not unheard of in system identification literature [17–20], it is more common to assume that both the training and the validation sets originate from a single time series, which represents the time sampled version of a single, continuous-time signal. However, we conjecture that the results of the paper could be extended to cover both challenges. The main technical contribution of the paper is a bound on

the Rademacher complexity of LPV models, and there are extension of PAC bounds to the case of a single time-series sampled from a mixing process [21, 22]. In other words, our conjecture is that the results of the current paper could be combined with [21, 22] to extend it beyond the case of multiple independent experiments. In addition, ideas from [23] using Bernstein polynomials could be used to handle time-sampling.

Related work PAC bounds for discrete-time linear systems were explored in system identification in [17, 24], but not for LPV systems in state-space form and continuous-time. PAC bounds for a class of autoregressive discrete-time systems were developed using Rademacher complexity in [25], but the derived results do not apply to continuous-time LPV state-space representations. There are several PAC bounds available for continuous-time RNNs and nonlinear systems [23, 26–30], but these are exponential in the integration time and they do not require stability. In [31–33] bounds for discrete-time RNNs were obtained via estimating the Rademacher complexity, but the bounds grow at least linearly with the number of time-steps. In [28, 29] PAC bounds for NCDEs were derived using Rademacher complexity as well, but the bounds grow exponentially with the integration interval. The closest result to our work is [30], which consider input-affine nonlinear systems and propose a PAC bound based on Rademacher complexity. Again, as stability was not taken into account, the bound is exponential in the length of the integration interval.

PAC-Bayesian bounds for discrete-time linear and classes of nonlinear systems were explored in [34–37]. However, these papers consider PAC-Bayesian, as opposed to PAC bounds, and they do not apply to continuous-time systems. We argue that while PAC-Bayesian bounds offer more flexibility, their tightness is sensitive to the choice of a suitable prior, making their application challenging. In contrast, classical PAC bounds are easier to use, but they tend to be more conservative. We present a more detailed discussion in Sect. 3.2.

The literature on finite-sample bounds for learning discrete-time dynamical systems, e.g., [38–42], is somewhat related, but considers different learning problems in the sense that these handle the matter of learning from a single trajectory. Some papers dealing with finite-sample bounds, e.g., [18–20], do consider learning from multiple trajectories. However, the papers [18–20, 38–42] provide risk bounds only for a specific learning algorithm instead of uniform bounds on the generalization gap. Namely, those papers provide an upper bound on the true loss of a model learnt using a specific learning algorithm. In contrast, we provide an upper bound on the difference between the true loss and the empirical loss, regardless of the origin of the model. Hence, the results of this paper can be used to bound the generalization gap and therefore the true loss of the outcome of any learning algorithm. Moreover, the cited papers do not deal with LPV state-space representations in continuous-time.

Significance and novelty The main novelty of the paper are that (1) our error bound does not depend on the length of the integration interval T; (2) it exploits quadratic stability; (3) it uses a weighted H_2 norm defined via Volterra-kernels to estimate the Rademacher complexity. This is in contrast to [23, 28–30] which used Fliess-series expansions for that purpose. It was precisely the use of Volterra-series and H_2 norms which allowed us to formulate bounds independent of T. The latter is important as dynamical systems are often used for making long-term predictions.

Structure of the paper In Sect. 2, we present our notations and definitions of LPV systems. In Sect. 3, we define the problem of generalization for LPV systems and present a brief general introduction to PAC bounds. In Sect. 4, we propose the main technical tools that allow us to exploit stability in the statistical learning setting, as well as the necessary assumptions needed to prove our theorem. We state our time-independent bound for the generalization gap in Sect. 5. The proof, presented in Sect. 6, is based on bounding the Rademacher complexity of the hypothesis set using the corollary of our stability assumption, namely that the considered LPV systems have finite H_2 norms. Finally, in Sect. 7 we consider a concrete system and a dataset in a numerical example, empirically estimate the elements of the bound and show that the bound in this case is meaningful.

Notations Scalars, vectors and matrices are denoted by simple lowercase, bold lowercase and bold uppercase symbols, respectively. We use round brackets for vectors, matrices and tuples as well, though the nature of the considered objects are always clear from the context. For a matrix $\mathbf{M}$, trace($\mathbf{M}$) refers to its trace, i.e., the sum of the diagonal elements, while $e^{\mathbf{M}}$ denotes the matrix exponential. The $\dot{}$ symbol refers to the partial differential operator w.r.t. the time dimension. The symbol T refers to the fixed integration time and the symbol $\mathcal{E}$ refers to an arbitrary and fixed set of LPV systems. We use $[n] = \{1, \ldots, n\}$ and $[n]_0 = \{0, \ldots, n\}$ for all $n \in \mathbb{N}$. The symbol $I_k = [n_\mathrm{p}]^k$ denotes the set of multi-indices of length k, thus an element $I \in I_k$ is a tuple of the form $I = (i_1, \ldots, i_k)$, $1 \leq i_j \leq n_p$ for $1 \leq j \leq k$. By slight abuse of notation, let I_0 be the singleton set $\{\emptyset\}$. For $t \in \mathbb{R}$ and $\boldsymbol{\tau} = (\tau_k, \ldots, \tau_1)^T \in \mathbb{R}^k$ we use the notation $(t, \boldsymbol{\tau}) = (t, \tau_k, \ldots, \tau_1)^T \in \mathbb{R}^{k+1}$. The symbols L^2 and L^∞ refer to the usual Lebesgue spaces. We denote by $L^2([0, T], \mathbb{R}^{n_\mathrm{in}})$ the space of all measurable functions $f : [0, T] \to \mathbb{R}^{n_\mathrm{in}}$ such that $\|f\|^2_{L^2([0,T],\mathbb{R}^{n_\mathrm{in}})} := \int_0^T \|f(t)\|_2^2 \mathrm{d}t$ is finite. The symbol $\prec$ refers to the Loewner order. For a function f, the symbol $f|_H$ denotes the restriction of f to the set H. For the notations related to probability, see Sect. 3.

2 LPV systems

In this paper, we consider *LPV state-space representations with affine dependence on the parameters (LPV-SSA)*, i.e., systems of the form

$$\Sigma \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{p}(t))\mathbf{x}(t) + \mathbf{B}(\mathbf{p}(t))\mathbf{u}(t) \\ \mathbf{x}(0) = \mathbf{0} \\ \mathbf{y}_\Sigma(t) = \mathbf{C}(\mathbf{p}(t))\mathbf{x}(t) \end{cases} \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_\mathrm{x}}$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^{n_\mathrm{in}}$ is the input and $\mathbf{y}_\Sigma(t) \in \mathbb{R}^{n_\mathrm{out}}$ is the output of the system for all $t \in [0, T]$, and $n_\mathrm{x}, n_\mathrm{in}, n_\mathrm{out} \in \mathbb{N}^+$ are the dimensions of the state, input and outputs spaces, respectively. The vector $\mathbf{p}(t) = (p_1(t), \ldots, p_{n_\mathrm{p}}(t))^T \in \mathbb{V} \subseteq \mathbb{R}^{n_\mathrm{p}}$ is the scheduling variable for $n_\mathrm{p} \in \mathbb{N}^+$. Note, that n_p is the number of scheduling variables in the system, therefore it is an important parameter regarding the complexity of the system. The matrices of the system Σ are assumed to depend

on $\mathbf{p}(t)$ affinely, i.e.,

$$\mathbf{A}(\mathbf{p}(t)) = \mathbf{A}_0 + \sum_{i=1}^{n_\mathrm{p}} p_i(t)\mathbf{A}_i \qquad \mathbf{B}(\mathbf{p}(t)) = \mathbf{B}_0 + \sum_{i=1}^{n_\mathrm{p}} p_i(t)\mathbf{B}_i,$$
$$\mathbf{C}(\mathbf{p}(t)) = \mathbf{C}_0 + \sum_{i=1}^{n_\mathrm{p}} p_i(t)\mathbf{C}_i$$

for matrices $\mathbf{A}_i \in \mathbb{R}^{n_\mathrm{x}\times n_\mathrm{x}}$, $\mathbf{B}_i \in \mathbb{R}^{n_\mathrm{x}\times n_\mathrm{in}}$ and $\mathbf{C}_i \in \mathbb{R}^{n_\mathrm{out}\times n_\mathrm{x}}$, $i = 0, \ldots, n_\mathrm{p}$, which do not depend on time or the scheduling signal. We identify the LPV-SSA Σ with the tuple $\Sigma = (\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i)_{i=0}^{n_\mathrm{p}}$.

A *solution* of Σ refers to the tuple of functions $(\mathbf{u}, \mathbf{p}, \mathbf{x}, \mathbf{y}_\Sigma)$, all defined on $[0, T]$, such that $\mathbf{x}$ is absolutely continuous, $\mathbf{y}_\Sigma$, $\mathbf{u}$ and $\mathbf{p}$ are piecewise continuous and they satisfy (1). These conditions guarantee existence and uniqueness of a solution of (1) and they are sufficiently general to cover many practical cases [1]. Note that the output $\mathbf{y}_\Sigma$ is uniquely determined by $\mathbf{u}$ and $\mathbf{p}$, since the initial state $\mathbf{x}(0)$ is set to zero. To emphasize this dependence, we denote $\mathbf{y}_\Sigma$ by $\mathbf{y}_\Sigma(\mathbf{u}, \mathbf{p})$. Thus, the scheduling signal $\mathbf{p}(t)$ behaves as an external input, too.

For the sake of compactness, we make a series of simplifications. First, as already stated, the initial state is set to zero. This is not a real restriction, as we consider stable systems for which the contribution of the nonzero initial state decays exponentially. Second, we work with systems with scalar output, i.e., let $n_\mathrm{out} = 1$. Therefore, instead of $\mathbf{y}_\Sigma$ we use the notation y_Σ. Third, we assume that the scheduling variables take values in $\mathbb{V} \subseteq [-1, 1]^{n_\mathrm{p}}$. The latter is a standard assumption in the literature [1] and it can always be achieved by an affine transformation, if the scheduling variables take values in a suitable interval.

3 Learning problem and generalization bounds

We now define the learning problem for LPV systems along the lines of classical statistical learning theory [10].

3.1 Learning problem

For the purpose of defining the learning problem, below we define the set of inputs, the set of labels (outputs), the class of models, the elementwise loss function and the true and empirical losses. To this end, let us fix a time interval $[0, T]$ and a set $\mathcal{E}$ of LPV systems of the form (1), both remain fixed during the rest of the paper.

Inputs and outputs Let $\mathcal{U}$, $\mathcal{P}$ and $\mathcal{Y}$ be sets of piecewise continuous functions defined on $[0, T]$ and taking values in $\mathbb{R}^{n_\mathrm{in}}$, $\mathbb{V}$ and $\mathbb{R}^{n_\mathrm{out}}$, respectively. These sets contain the considered input, scheduling and output trajectories. Hereinafter we use the standard terminology of probability theory [43]. Consider the probability space $(\mathcal{U} \times \mathcal{P} \times \mathcal{Y}, \mathcal{B}, \mathcal{D})$, where $\mathcal{B}$ is a suitable σ-algebra and $\mathcal{D}$ is a probability measure on $\mathcal{B}$. For example, $\mathcal{B}$ could be the direct product of the standard cylindrical Borel

σ-algebras defined on the function spaces $\mathcal{U}$, $\mathcal{P}$ and $\mathcal{Y}$. Let us denote by $\mathcal{D}^N$ the N-fold product measure of $\mathcal{D}$ with itself. We use $\mathbb{E}_{(\mathbf{u},\mathbf{p},y)\sim\mathcal{D}}$, $\mathbb{P}_{(\mathbf{u},\mathbf{p},y)\sim\mathcal{D}}$, $\mathbb{E}_{\mathbf{S}\sim\mathcal{D}^N}$ and $\mathbb{P}_{\mathbf{S}\sim\mathcal{D}^N}$ to denote expectations and probabilities w.r.t. the measures $\mathcal{D}$ and $\mathcal{D}^N$, respectively. The notation $\mathbf{S} \sim \mathcal{D}^N$ tacitly assumes that $\mathbf{S} \in (\mathcal{U} \times \mathcal{P} \times \mathcal{Y})^N$, i.e., $\mathbf{S}$ is made of N triplets of input, scheduling and output trajectories. Intuitively, we think of $\mathbf{S}$ as a dataset of size N drawn randomly and independently from the distribution $\mathcal{D}$.

Elementwise loss function Consider an *elementwise loss function* $\ell : \mathbb{R} \times \mathbb{R} \to [0, +\infty)$, which measures the discrepancy between two possible output values. Some of the widespread choices are $\ell(a, b) = \|a - b\|$ or $\ell(a, b) = \|a - b\|^2$.

True and empirical loss, generalization gap The learning objective is to find an LPV system $\Sigma \in \mathcal{E}$ such that the *true risk at time T* (also referred to as *true error*, *true prediction error* or *true loss*), defined as

$$\mathcal{L}(\Sigma) = \mathbb{E}_{(\mathbf{u},\mathbf{p},y)\sim\mathcal{D}}\left[\int_{[0,T]} \ell(y_\Sigma(\mathbf{u}, \mathbf{p})(\tau), y(\tau))d\mu(\tau)\right]$$

is as small as possible. Since the distribution $\mathcal{D}$ is unknown, minimizing the true risk is impossible. Therefore, the true risk is approximated by the *empirical risk at time T* w.r.t. a dataset $\mathbf{S} = \{(\mathbf{u}_i, \mathbf{p}_i, y_i)\}_{1\le i\le N}$ (also referred to as *empirical error*, *empirical prediction error* or *empirical loss*), defined as

$$\mathcal{L}_N^{\mathbf{S}}(\Sigma) = \frac{1}{N}\sum_{i=1}^{N}\int_{[0,T]} \ell(y_\Sigma(\mathbf{u}_i, \mathbf{p}_i)(\tau), y_i(\tau))d\mu(\tau).$$

In the above definitions, we require the measure μ to be σ-finite on the Borel sets generated by $[0, T]$, and to be normalized such that $\mu([0, T]) \le 1$. These definitions are quite general, however, commonly used loss functions in practical applications can be recovered by suitable special cases of μ.

- If μ is the Dirac measure at T, i.e., for any measurable set $H \subseteq [0, T]$ we have $\mu(H) = 1$ if $T \in H$, otherwise $\mu(H) = 0$, then the true and empirical losses only depend on $y_\Sigma(\mathbf{u}, \mathbf{p})(T)$ and $y(T)$. This loss function is usually employed for NeuralODEs [44]. In this case as $\mu(\{T\}) = 1$, we have

$$\begin{aligned}
&\int_{[0,T]} \ell(y_\Sigma(\mathbf{u}, \mathbf{p})(\tau), y(\tau))d\mu(\tau) = \ell(y_\Sigma(\mathbf{u}, \mathbf{p})(T), y(T))\\
&\mathcal{L}(\Sigma) = \mathbb{E}_{(\mathbf{u},\mathbf{p},y)\sim\mathcal{D}}\left[\ell(y_\Sigma(\mathbf{u}, \mathbf{p})(T), y(T))\right]\\
&\mathcal{L}_N^{\mathbf{S}}(\Sigma) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_\Sigma(\mathbf{u}_i, \mathbf{p}_i)(T), y_i(T)).
\end{aligned} \tag{2}$$

- If μ is a sum of Dirac measures, respectively, at $t_1, \ldots, t_{k_T}$ that puts the weight $\frac{1}{k_T}$ on each t_i, then for any measurable set $H \subseteq [0, T]$ we have $\mu(H) = \frac{1}{k_T}\sum_{i=1}^{k_T} \chi_{\{t_i\in H\}}$, where $\chi_{\{t_i\in H\}} = 1$ if $t_i \in H$ and 0 otherwise.

This choice of μ recovers the case when the output of the system is discretized in time and measured only at $0 = t_1, \ldots, t_{k_T} = T$, we have

$$\begin{aligned}
&\int_{[0,T]} \ell(y_\Sigma(\mathbf{u}, \mathbf{p})(\tau), y(\tau)) d\mu(\tau) = \frac{1}{k_T} \sum_{i=1}^{k_T} \ell(y_\Sigma(\mathbf{u}, \mathbf{p})(t_i), y(t_i)) \\
&\mathcal{L}(\Sigma) = \mathbb{E}_{(\mathbf{u},\mathbf{p},y)\sim\mathcal{D}} \left[\frac{1}{k_T} \sum_{i=1}^{k_T} \ell(y_\Sigma(\mathbf{u}, \mathbf{p})(t_i), y(t_i)) \right] \\
&\mathcal{L}_N^{\mathbf{S}}(\Sigma) = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{k_T} \sum_{i=1}^{k_T} \ell(y_\Sigma(\mathbf{u}_j, \mathbf{p}_j)(t_i), y_j(t_i)).
\end{aligned} \tag{3}$$

- If μ is the normalized Lebesgue measure with density $\frac{1}{T}$, we recover the following integral loss function applied in [45] for NeuralODEs, i.e.,

$$\begin{aligned}
&\int_{[0,T]} \ell(y_\Sigma(\mathbf{u}, \mathbf{p})(\tau), y(\tau)) d\mu(\tau) = \frac{1}{T} \int_0^T \ell(y_\Sigma(\mathbf{u}, \mathbf{p})(\tau), y(\tau)) \mathrm{d}\tau \\
&\mathcal{L}(\Sigma) = \mathbb{E}_{(\mathbf{u},\mathbf{p},y)\sim\mathcal{D}} \left[\frac{1}{T} \int_0^T \ell(y_\Sigma(\mathbf{u}, \mathbf{p})(\tau), y(\tau)) \mathrm{d}\tau \right] \\
&\mathcal{L}_N^{\mathbf{S}}(\Sigma) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{T} \int_0^T \ell(y_\Sigma(\mathbf{u}_i, \mathbf{p}_i)(\tau), y_i(\tau)) \mathrm{d}\tau.
\end{aligned} \tag{4}$$

Note, that for all the cases above, μ is σ-finite and $\mu([0, T]) = 1$.

Remark 1 (Role of scheduling) The discussion above implies that in the learning problem at hand, the scheduling signal $\mathbf{p}$ may depend on $\mathbf{u}$ or y, and the data may be generated by a quasi-LPV system. However, for the models to be learnt the scheduling signal acts as an input. This is consistent with the assumptions in system identification for LPV systems [1]. The dependence of $\mathbf{p}$ on $\mathbf{u}$ and y does not conflict with our proof technique and highlights the potential future research direction toward extending our results to Neural ODEs and other similar structures.

Remark 2 (Role of time sampling) The considered definition of LPV systems include piecewise constant $\mathbf{u}$ and $\mathbf{p}$ corresponding to the case when the observed data are measured only in discrete-time points, i.e., it is sampled in time from a continuous trajectory. The time sampling of the system output is also included, namely in the form of the loss functions described by Equation (3). As we will later see, our results do not depend on the choice of the measure μ or the presence of piecewise constant input and scheduling signals. Therefore, the time sampling has no particular effect on the considered learning framework.

3.2 Probably approximately correct (PAC) bounds: general introduction

In practice, selecting an appropriate model is done by minimizing the empirical risk w.r.t. a so-called training dataset, while the trained model is usually evaluated by computing the empirical risk w.r.t. a separate test dataset. In both cases, we need a bound on the difference between the true and the empirical risks. That is, we need to bound the *generalization gap*, defined as $\sup_{\Sigma\in\mathcal{E}} \left(\mathcal{L}(\Sigma) - \mathcal{L}_N^{\mathbf{S}}(\Sigma)\right)$. *Probably Approximately Correct (PAC)* [10] bounds are bounds of the form

$$\mathbb{P}_{\mathbf{S}\sim\mathcal{D}^N}\left(\forall\Sigma\in\mathcal{E} : \mathcal{L}(\Sigma) - \mathcal{L}_N^{\mathbf{S}}(\Sigma) \leq \frac{R(\delta)}{\sqrt{N}}\right) \geq 1-\delta, \tag{5}$$

where $R(\delta)$ is a constant which depends on the confidence level δ and model family $\mathcal{E}$, and N is the number of data points used to measure the empirical error. Intuitively, such a PAC bound says that for any element of the parameterized family, the generalization gap is smaller than $\frac{R(\delta)}{\sqrt{N}}$ with the probability $1-\delta$. For several model classes [10], it was shown that $R(\delta)$ is $O\left(\log\left(\frac{1}{\delta}\right)\right)$. In this paper, we will present an analytic expression for $R(\delta)$ for a class $\mathcal{E}$ of LPV systems.

Intuitively, the PAC bound (5) implies that for sufficiently large N, the gap between the empirical loss and the true loss is small. However, this gap increases with the confidence level $1-\delta$, as the constant $R(\delta)$ is a decreasing function of δ.

Applications to model validation PAC bounds can be used for validating and learning models as follows. First of all, if we want to evaluate a certain model $\Sigma\in\mathcal{E}$, the PAC bound implies that with probability at least $1-\delta$ over the validation dataset

$$\mathcal{L}(\Sigma) \leq \mathcal{L}_N^{\mathbf{S}}(\Sigma) + \frac{R(\delta)}{\sqrt{N}} \tag{6}$$

holds. This can be used to estimate the confidence interval of the true loss based on the empirical loss and to estimate the number of data points which are necessary to make this confidence interval sufficiently small. Indeed, for any $\epsilon > 0$, by choosing $N > \frac{R^2(\delta)}{\epsilon^2}$, $\mathcal{L}(\Sigma)$ will be in the interval $[0, \mathcal{L}_N^{\mathbf{S}}(\Sigma)+\epsilon]$ with probability $1-\delta$. In particular, if the empirical loss is small and there is enough data points, we can conclude that the true loss will also be small with high probability.

Applications to learning First of all, learning algorithms can be viewed as a map sending the training data $\mathbf{S}$ to a data-dependent element $\hat{\Sigma}$ of the model family $\mathcal{E}$. Then, a PAC bound implies that with probability at least $1-\delta$ over the training samples $\mathbf{S}$,

$$\mathcal{L}(\hat{\Sigma}) \leq \mathcal{L}_N^{\mathbf{S}}(\hat{\Sigma}) + \frac{R(\delta)}{\sqrt{N}}. \tag{7}$$

That is, we get again a confidence interval for the true loss, based on the empirical loss during training. That is, for any $\epsilon > 0$, if $N \geq \frac{R^2(\delta)}{\epsilon^2}$, then $\mathcal{L}(\hat{\Sigma})$ lies in the interval $[0, \mathcal{L}_N^{\mathbf{S}}(\Sigma)+\epsilon]$ with probability $1-\delta$.

Another important application is to estimate the difference between the true loss of the model learnt by Empirical Risk Minimization [10] and the smallest possible true loss. More precisely, let $\hat{\Sigma}$ be the model which minimizes the empirical loss for a given training dataset $\mathbf{S}$, i.e.,

$$\hat{\Sigma} = \arg\min_{\Sigma \in \mathcal{E}} \mathcal{L}_N^{\mathbf{S}}(\Sigma).$$

In the system identification literature, the algorithm returning $\hat{\Sigma}$ is sometimes referred to as Prediction Error Minimization (PEM).

Under suitable assumptions (e.g., bounded loss functions), from the proof of [10, Theorem 26.5 , Part 3] it follows that

$$\mathcal{L}(\hat{\Sigma}) \leq \inf_{\Sigma \in \mathcal{E}} \mathcal{L}(\Sigma) + \frac{R(\delta) + R_2(\delta)}{\sqrt{N}} \tag{8}$$

with probability at least $1 - \delta$ over all the samples $\mathbf{S}$, where $R_2(\delta)$ is a $O(\ln(\frac{2}{\delta}))$ constant which depends on the upper bound of the loss function. The inequality (8) implies that learning algorithms based on minimizing the empirical loss will achieve the smallest possible true loss with arbitrary accuracy for large enough number of data points. That is, by choosing $N > \frac{(R(\delta)+R_2(\delta))^2}{\epsilon^2}$ it follows that

$$\mathcal{L}(\hat{\Sigma}) \leq \inf_{\Sigma \in \mathcal{E}} \mathcal{L}(\Sigma) + \epsilon \tag{9}$$

with probability at least $1-\delta$ over the training data $\mathbf{S}$. The latter property implies *PAC-learnability* [10], and it is a classical sufficient condition for learning to be feasible. In other words, PAC bounds allow us to show the feasibility of the learning problem, see [10] for a more detailed discussion.

PAC bounds versus quality of approximation The PAC bounds (7)–(8), and by extension the parametrization-dependent term $R(\delta)$ on their right-hand side, do not provide an absolute upper bound on how well the learned model approximates the true system of interest (captured by the true loss). Instead, they provide a *relative bound*, i.e., a bound on the difference between the true loss and some other quantity. This quantity can be either the empirical loss or the best achievable true loss $\inf_{\Sigma \in \mathcal{E}} \mathcal{L}(\Sigma)$. The latter is commonly referred to as the *approximation error of the set of parametrizations* $\mathcal{E}$ [10, Section 5.2], and it expresses its ability to represent the true system.

The term $R(\delta)$ provides a lower bound on the number of data points from which the learning problem can be guaranteed to be well behaved. Informally, this means that (i) any learning algorithm produces a model with a small generalization gap (see (5)) and (ii) there exists at least one learning algorithm whose true loss is close to the approximation error (see (8) and (9)).

Intuitively, $R(\delta)$ measures how strongly the input–output behavior of models from $\mathcal{E}$ is determined by their behavior on randomly selected data of a given size. Mathematically, this is formalized by *complexity* measures such as the Vapnik–Chervonenkis

dimension, covering numbers and Rademacher complexity [10], and $R(\delta)$ is typically proportional to these quantities.

Tradeoff between complexity and approximation error There is a well-known tradeoff between the ability of $\mathcal{E}$ to generalize, captured by $R(\delta)$ via some complexity measure, and the approximation error $\inf_{\Sigma\in\mathcal{E}} \mathcal{L}(\Sigma)$ [10, Section 5.2]. For the approximation error to be small, usually $\mathcal{E}$ must contain a large variety of models. However, in this case $R(\delta)$ tends to be large, since with richer model classes it becomes easier to find a model that fits the training data but behaves very differently on unseen data.

Applications to parameter estimation Equation (8) can be used to show that the learning algorithm based on empirical loss minimization is consistent, if viewed as a parameter estimation algorithm for system identification. More precisely, (8) implies that the true loss $\mathcal{L}(\hat{\Sigma})$ converges to the optimal loss $\inf_{\Sigma\in\mathcal{E}} \mathcal{L}(\Sigma)$. It is well-known in system identification that this latter property, under suitable regularity conditions ([9, Chapter 8]), implies the learnt parameter represented by $\hat{\Sigma}$ converges to the parameter $\arg\min_{\Sigma\in\mathcal{E}} \mathcal{L}(\Sigma)$ that minimizes the true loss. In turn, if the true system corresponds to a certain element of the considered parametrizations, then the corresponding parameter is the unique minimizer of $\mathcal{L}(\Sigma)$, and hence $\hat{\Sigma}$ converges to the parameter of the true system.

The regularity conditions mentioned have not been worked out for continuous-time LPV systems, but for discrete-time LTI systems they are known [9, Chapter 8].

PAC-Bayesian bounds An alternative approach for generalization bounds are the so-called PAC-Bayesian bounds [11]. In contrast to PAC bounds, they do not bound the worst-case generalization gap, but rather the worst-case mean generalization gap, where the mean is taken w.r.t. posterior distributions on the model class. The bounds then typically depend on a chosen prior distribution. Although they might result in tighter bounds, attaining a well-tuned prior is a very challenging problem both theoretically and in practice. Therefore, PAC bounds might be a reasonable alternative for PAC-Bayesian ones whenever well-tuned priors may not be available, e.g., discrete-time dynamical systems [46], but we have good estimates of the Rademacher complexity of the hypothesis class, which is the case for this paper.

4 Technical preliminaries and assumptions

We start by presenting a Volterra-series representation of the output of an LPV system of the form (1), which plays a central role in formulating and proving the main result. To this end, we introduce the concept of iterated integrals.

Definition 3 (*Iterated integrals* [15]) For any positive integer k, let

$$\Delta_k^t = \left\{(\tau_k, \ldots, \tau_1)^T \mid t \geq \tau_k \geq \cdots \geq \tau_1 \geq 0\right\} \subset \mathbb{R}^k$$
$$\Delta_k^\infty = \left\{(\tau_k, \ldots, \tau_1)^T \mid \tau_k \geq \cdots \geq \tau_1 \geq 0\right\} \subset \mathbb{R}^k$$

Clearly, $\Delta_k^t \subseteq \Delta_k^\infty$ for all $t \in [0, +\infty)$. In addition, we use the following notation for iterated integrals (for any function f for which the integrals are well defined), for

$t \in [0, +\infty]$

$$\int_{\Delta_k^t} f(\boldsymbol{\tau})\, d\boldsymbol{\tau} \equiv \int_0^t \int_0^{\tau_k} \cdots \int_0^{\tau_2} f(\tau_k, \ldots, \tau_1)\, \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_k.$$

Now we will define the λ-*weighted LPV Volterra-kernels* and the λ-*weighted scheduling-input products* for a system Σ of the form (1), for every $I \in I_k$, $t \in [0, +\infty)$, $i_q, i_r \in [n_\mathrm{p}]_0$, and for every $\mathbf{p} \in \mathcal{P}$, $\mathbf{u} \in \mathcal{U}$ and $\lambda \geq 0$. These objects are the main technical tools that allow us to interpret the stability property of LPV systems within the framework of statistical learning.

Definition 4 (λ-*weighted LPV Volterra-kernels*) The λ-*weighted LPV Volterra-kernels* $w^\lambda_{i_q,i_r,I} : \Delta^\infty_{k+1} \to \mathbb{R}^{1\times n_\mathrm{in}}$ are defined as follows for $\lambda \geq 0$.

For $k = 0$ and $I = \emptyset$, for any $\tau = \tau_1 \in \Delta^\infty_1 = [0, \infty)$, let

$$w^\lambda_{i_q,i_r,\emptyset}(\tau) := \mathbf{C}_{i_q} \mathrm{e}^{\mathbf{A}_0\tau_1} \mathbf{B}_{i_r} \mathrm{e}^{\frac{\lambda}{2}\tau_1}.$$

For $k > 0$, $\boldsymbol{\tau} = (\tau_{k+1}, \ldots, \tau_1)^T \in \Delta^\infty_{k+1}$ and $I = (i_1, \ldots, i_k)$ let

$$w^\lambda_{i_q,i_r,I}(\boldsymbol{\tau}) := \mathbf{C}_{i_q} \mathrm{e}^{\mathbf{A}_0(\tau_{k+1}-\tau_k)} \mathbf{A}_{i_k} \mathrm{e}^{\mathbf{A}_0(\tau_k-\tau_{k-1})} \cdots \mathbf{A}_{i_1} \mathrm{e}^{\mathbf{A}_0\tau_1} \mathbf{B}_{i_r} \mathrm{e}^{\frac{\lambda}{2}\tau_{k+1}}.$$

The Volterra-kernels defined above do not depend on the input or scheduling trajectories, but on the system parameters only. Foreshadowing Definition 7 and Theorem 10, these objects induce a weighted extension of the H_2 norm, which will provide a way to quantize the degree of stability and eventually upper bound the generalization gap of LPV systems. The effect of input and scheduling variables on the output of the considered systems is captured by the following term.

Definition 5 (λ-*weighted scheduling-input products*) Let $p_\emptyset(\tau) := 1$ for $\tau \in [0, T]$ and $p_I(\boldsymbol{\tau}) := \prod_{j=1}^k p_{i_j}(\tau_j + T - \tau_{k+1})$ for $\boldsymbol{\tau} = (\tau_{k+1}, \ldots, \tau_1)^T \in \Delta^T_{k+1}$ and $I = (i_1, \ldots, i_k)$. The λ-*weighted scheduling-input products* $\varphi^\lambda_{i_q,i_r,I} : \Delta^T_{k+1} \to \mathbb{R}^{n_\mathrm{in}}$ are defined as follows for $\lambda \geq 0$.

For $k = 0$ and $I = \emptyset$, for any $\tau = \tau_1 \in \Delta^T_1 = [0, T]$, let

$$\varphi^\lambda_{i_q,i_r,\emptyset}(\tau) := p_{i_q}(T) p_{i_r}(T - \tau_1) \mathbf{u}(T - \tau_1) \mathrm{e}^{-\frac{\lambda}{2}\tau_1}.$$

For $k > 0$, $\boldsymbol{\tau} = (\tau_{k+1}, \ldots, \tau_1)^T \in \Delta^T_{k+1}$, and $I = (i_1, \ldots, i_k)$ let

$$\varphi^\lambda_{i_q,i_r,I}(\boldsymbol{\tau}) := p_{i_q}(T) p_{i_r}(T - \tau_{k+1}) p_I(\boldsymbol{\tau}) \mathbf{u}(T - \tau_{k+1}) \mathrm{e}^{-\frac{\lambda}{2}\tau_{k+1}}.$$

In both of the definitions above, the value k is always equal to the size of the tuple I. The respective domains of these functions are made of vectors of size $k + 1$ (denoted by $\boldsymbol{\tau}$). The λ-weighted LPV Volterra-kernels represent weighted Volterra-kernels of certain bilinear systems, outputs of which determine the output of (1). The λ-weighted

scheduling-input products capture the polynomial relationship between the outputs of these bilinear systems and the scheduling and input signals. The exponential weighting terms were introduced in order to make the series of the L^2 norms of these products square summable. The terms belonging to i_r and i_q are related to the effect of $\mathbf{B}_{i_r}$ and $\mathbf{C}_{i_q}$ on the output of (1). The following Lemma captures this intuition in a rigorous way.

Lemma 6 *For every* $(\mathbf{u}, \mathbf{p}) \in \mathcal{U} \times \mathcal{P}$, $\lambda \geq 0$, *the output of* Σ *at time* T *admits the following representation:*

$$y_\Sigma(\mathbf{u}, \mathbf{p})(T) = \sum_{i_q, i_r = 0}^{n_p} \sum_{k=0}^{\infty} \sum_{I \in I_k} \int_{\Delta_{k+1}^T} w_{i_q, i_r, I}^{\lambda}(\boldsymbol{\tau}) \varphi_{i_q, i_r, I}^{\lambda}(\boldsymbol{\tau}) d\boldsymbol{\tau}.$$

The proof is based on a Volterra-series expansion [15] and can be found in Appendix A. The Lemma states that the output of Σ is an infinite sum of convolutions of the inputs with iterated integrals of the scheduling signal. This observation allows us to represent the output of an LPV system as a scalar product in a suitable Hilbert space which turns out to be the key for the proof of the main result.

Next, we define the λ-*weighted* H_2 *norm*, a variant of the classical H_2 norm, parameterized by a constant $\lambda \geq 0$.

Definition 7 The λ-*weighted* H_2 *norm* of a system Σ of the form (1) is defined as follows for $\lambda \geq 0$.

$$\|\Sigma\|_{\lambda, H_2}^2 := \sum_{i_q, i_r = 0}^{n_p} \sum_{k=0}^{\infty} \sum_{I \in I_k} \int_{\Delta_{k+1}^\infty} \left\| w_{i_q, i_r, I}^{\lambda}(\boldsymbol{\tau}) \right\|_2^2 d\boldsymbol{\tau}.$$

There exist several, non-equivalent definitions of H_2 norms for LPV systems, see for instance [47, Section 2.2], [48, Section 7.4]. These definitions are equivalent when applied to LTI systems, however, even for LTV (Linear Time-Varying) systems they are not equivalent, see [47, Section 2.2] for an overview. When extending these definitions to LPV systems, one can choose one of these definitions of H_2 norms, and then take the maximum of the H_2 norms of the corresponding LTV systems, where the maximum is taken over the scheduling signals (Δ-blocks). This was done for instance in [49, 50]. The norm $\|\Sigma\|_{\lambda, H_2}$ is different from these other H_2 norms for LPV systems. However, as it is shown in Lemma 8, under certain stability conditions this norm exists, and similarly to other H_2 norms, it is an upper bound on peak output under unit energy input.

The definition of $\|\Sigma\|_{\lambda, H_2}$ is similar to the approach in [51], where the average of Volterra-kernels is taken as the definition of the H_2 norm. The H_2 norm in this paper involves a weighting term, hence it can be considered an extension of the norm used in [51]. The introduction of the weighting term is motivated by the need to upper bound the Rademacher complexity of LPV systems. In contrast, the various definitions of the H_2 norm used in the literature were motivated by the need to formalize various control objectives. We point out that the precise relationship between the H_2 norm used in this paper and the various definitions used in the literature requires further research.

While the definition of the H_2 norm $\|\Sigma\|_{\lambda,H_2}$ is different from many other norms defined in the literature, e.g., [47–50], they do share certain characteristics. In particular, it can be shown that they all provide an upper bound on the norm of the input–output operator generated by the system, where the input space is taken to be L^2 and the output space L^∞, see Lemma 8 in the next section. Moreover, when applied to LTI systems with $\lambda = 0$, the above defined norm is the classical H_2 norm.

Intuitively, the norm $\|\Sigma\|_{\lambda,H_2}$ is finite whenever the Volterra-kernels $w^{\lambda}_{i_q,i_r,I}$ are finite energy signals and the sum of their L^2 norms is finite, too. For $\lambda = 0$, the definition $\|\Sigma\|_{\lambda,H_2}$ is a direct extension of the H_2 norm for bilinear systems [52].

4.1 Assumptions

In order to state the main result of the paper, we state several assumptions.

Assumption 1 (*Stability*) There exists $\lambda \geq n_{\mathrm{p}}$ such that for any $\Sigma \in \mathcal{E}$ of the form (1) there exists $\mathbf{Q}_\Sigma \succ 0$ such that

$$\mathbf{A}_0^T\mathbf{Q}_\Sigma + \mathbf{Q}_\Sigma\mathbf{A}_0 + \sum_{i=1}^{n_{\mathrm{p}}} \mathbf{A}_i^T\mathbf{Q}_\Sigma\mathbf{A}_i + \sum_{i=0}^{n_{\mathrm{p}}} \mathbf{C}_i^T\mathbf{C}_i \prec -\lambda\mathbf{Q}_\Sigma. \tag{10}$$

Assumption 2 (*Bounded H_2 norm*) We assume that $\sup_{\Sigma\in\mathcal{E}} \|\Sigma\|_{\lambda,H_2} \leq c_{\mathcal{E}}$.

Assumption 3 (*Bounded signals*) For any $\mathbf{u} \in \mathcal{U}$ and $y \in \mathcal{Y}$, $\|\mathbf{u}\|_{L^2([0,T],\mathbb{R}^{n_{\mathrm{in}}})} \leq c_{\mathcal{U}}$ and $|y(T)| \leq c_{\mathcal{Y}}$.

Assumption 4 (*Lipschitz loss function*) The loss function ℓ is c_ℓ-Lipschitz-continuous, i.e., $|\ell(y_1, y_1') - \ell(y_2, y_2')| \leq c_\ell(|y_1 - y_2| + |y_1' - y_2'|)$ for all $y_1, y_2, y_1', y_2' \in \mathbb{R}$, and $\ell(y, y) = 0$ for all $y \in \mathbb{R}$.

(*Discussion on the assumptions*) Assumption 1 ensures quadratic stability of all LPV systems in the considered set of parametrizations with a decay rate $\frac{\lambda}{2}$, and it ensures the finiteness of H_2 norms as per the following Lemma.

Lemma 8 *If Assumption 1 holds with $\lambda \geq n_p$, then for any $\Sigma \in \mathcal{E}$ and $\mathbf{Q}_\Sigma \succ 0$ from Assumption 1, we have*

$$\|\Sigma\|^2_{\lambda,H_2} \leq \sum_{i_r=0}^{n_p} \mathrm{trace}(\mathbf{B}_{i_r}^T\mathbf{Q}_\Sigma\mathbf{B}_{i_r}) < +\infty.$$

Additionally, for any $\mathbf{u} \in \mathcal{U}$, $\mathbf{p} \in \mathcal{P}$, we have

$$|y_\Sigma(\mathbf{u}, \mathbf{p})(T)| \leq (n_p + 1)\,\|\Sigma\|_{\lambda,H_2}\,\|\mathbf{u}\|_{L^2([0,T],\mathbb{R}^{n_{\mathrm{in}}})}.$$

The proof can be found in Appendix B. Note that the second part of Lemma 8 implies Bounded Input Bounded Output (BIBO) stability: if $\mathbf{u}$ has a bounded L^2

norm on $[0, \infty)$, i.e., $\sup_{T>0} \|\mathbf{u}\|_{L^2([0,T],\mathbb{R}^{n_{\text{in}}})} < +\infty$, then Lemma 8 implies that $\sup_{T>0} |y_\Sigma(\mathbf{u}, \mathbf{p})(T)| < (n_{\text{p}} + 1) \|\Sigma\|_{\lambda, H_2} \sup_{T>0} \|\mathbf{u}\|_{L^2([0,T],\mathbb{R}^{n_{\text{in}}})} < +\infty$.

Assumption 1 is not restrictive as it can be translated into inequalities on the eigenvalues of the system matrices (see Appendix B), which are not too difficult to ensure by choosing a suitable parametrization. For example, in case of deep state-space architectures, the matrix $\mathbf{A}$ is usually chosen to be diagonal [14]. In general, stable parametrizations are quite common in learning [9, 53], and there is a rich literature on stable parametrization of LTI systems, e.g., [54–58]. In particular, stability of the parametrizations is usually assumed in LPV system identification, e.g., [1, 59–61]. Moreover, stable LPV systems contain exponentially stable bilinear systems and it is known that such bilinear systems are universal approximators for nonlinear input–output systems with fading memory [62]. Intuitively, many forms of stability constraints on input–output systems satisfy the definition of fading memory. This is a strong indication that stable LPV systems might be universal approximators for stable systems.

However, we are not aware of any formal result in the literature. Relevant results on the topic can be found in [62–65].

Assumption 2 ensures that the λ-weighted H_2 norm of the elements of class $\mathcal{E}$ are bounded by $c_{\mathcal{E}}$. Note, that Assumption 2 does not automatically follow from Assumption 1. Indeed, Assumption 1 implies that every element of $\mathcal{E}$ has a finite H_2 norm, but it does not guarantee the existence of a uniform upper bound on the H_2 norms of the elements of $\mathcal{E}$. However, by Lemma 8, Assumption 2 follows from Assumption 1 for instance, for continuous parametrizations with a bounded compact parameter set, or when $\mathbf{Q}_\Sigma$ is independent of Σ and the matrices $\mathbf{B}_i$ are bounded for all $\Sigma \in \mathcal{E}$.

Below we present a sufficient condition for computing an upper bound on the H_2 norms of models from $\mathcal{E}$.

Lemma 9 *Assume that there exist positive real numbers $\gamma > 0$ and $\Gamma > 0$ such that $\gamma \geq \Gamma + n_p$ and for every $\Sigma \in \mathcal{E}$ of the form (1)*

$$\Gamma > \sup_{1 \leq i \leq n_p} \|\mathbf{A}_i\|_2^2 n_p \quad \text{and} \quad \left\| \mathrm{e}^{\mathbf{A}_0 t} \right\|_2 \leq \mathrm{e}^{-\frac{\gamma}{2} t}.$$

Then Assumption 1 holds for any $n_p \leq \lambda < \gamma - \Gamma$ and $\mathbf{Q}_\Sigma = \mathbf{I} \frac{K_C^2 (n_{\text{p}}+1)}{\gamma - \lambda - \Gamma}$, and

$$c_{\mathcal{E}} \leq (n_p + 1)^2 K_C^2 K_B^2 n_{\text{in}} \left(\frac{1}{\gamma - \lambda - \Gamma} \right)$$

where $K_{\mathbf{B}} = \sup_{0 \leq i \leq n_p} \|\mathbf{B}_i\|_2$ and $K_{\mathbf{C}} \geq \sup_{0 \leq i \leq n_p} \|\mathbf{C}_i\|_2$, $K_{\mathrm{C}} > 0$,

The proof of Lemma 9 follows from a simple calculation followed by applying Lemma 8. Using the Linear Matrix Inequality (LMI) of Assumption 1 and Lemma 8, it is possible to formulate further LMI-based sufficient conditions for Assumption 2 to hold. However, working out such conditions would go beyond the scope of the paper.

Assumption 3 means that the input signal has finite energy and the output signal is bounded. The assumption on $|y(T)|$ means the true labels are bounded. By Lemma 8,

the first two assumptions along with finite energy inputs already imply that the outputs of the systems from $\mathcal{E}$ are bounded uniformly by a suitable constant. In practice, considering finite energy inputs and bounded outputs is often natural.

Assumption 4 is standard in machine learning, it is satisfied for $\ell(y, y') = |y - y'|$, and even for the square loss, if the latter is restricted to bounded labels.

5 Main result

We are ready to state our main contribution, which is a PAC bound of the type (5).

Theorem 10 (Main result) *Let*

$$c := 2c_\ell \max\{c_{\mathcal{U}}(n_p + 1)c_{\mathcal{E}}, c_{\mathcal{Y}}\}$$
$$R(\delta) := c\left(2 + 4\sqrt{2\log(4/\delta)}\right).$$

Under Assumptions 1–4 , for any $\delta \in (0, 1)$*, we have*

$$\mathbb{P}_{\mathbf{S}\sim\mathcal{D}^N}\left(\forall \Sigma \in \mathcal{E} : \mathcal{L}(\Sigma) - \mathcal{L}_N^{\mathbf{S}}(\Sigma) \le \frac{R(\delta)}{\sqrt{N}}\right) \ge 1 - \delta.$$

Application of PAC to learning and system identification As it was already mentioned in Sect. 1, system identification and learning theory have similar objectives, hence we will use the term learning for both in the discussion below and discuss them separately, whenever the difference is relevant.

Again, as it was mentioned in Sect. 3.2, PAC bounds can be used to bound the true loss of the learned model using its empirical loss on the validation or training data, see Equations (6) and (7). That is, any learning algorithm maps a dataset $\mathbf{S}$ to a model $\hat{\Sigma} = \hat{\Sigma}(\mathbf{S})$. As a PAC bound holds uniformly on all models, with probability at least $1 - \delta$ over $\mathbf{S}$, we get the explicit high-probability bound on the true error

$$\mathcal{L}(\hat{\Sigma}) \le \mathcal{L}_N^{\mathbf{S}}(\hat{\Sigma}) + \frac{R(\delta)}{\sqrt{N}}.$$

This allows us to evaluate the prediction error of the identified model for unseen data. Moreover, for any accuracy $\epsilon > 0$ we can determine the minimum number of data points $N_m = \frac{R^2(\delta)}{\epsilon^2}$ such that if $N \ge N_m$, the true loss of *any* identified model is smaller than ϵ plus the empirical loss. The integer N_m represents the minimal number of data points after which we can view the empirical loss as indicative of the true loss.

Moreover, as it was discussed in Sect. 3.2, a PAC bound from Theorem 10 can be used to relate the true loss of the model learnt by empirical loss minimization and the best achievable true loss, see Equation (8). More precisely, consider the minimal prediction error model $\hat{\Sigma} = \arg\min_{\Sigma\in\mathcal{E}} \mathcal{L}_N^{\mathbf{S}}(\Sigma)$ and the best possible model $\Sigma_\star = \arg\min_{\Sigma\in\mathcal{E}} \mathcal{L}(\Sigma)$, provided they exist. Then using [10, Theorem 26.5] and the upper bound on the Rademacher complexity of $\mathcal{E}$ from the proof of Theorem 10, we can get

the following high-probability upper bounds on the difference between the true loss of the these two models.

Corollary 11 *With the notation of Theorem 10 and with* $R_2(\delta) := c\left(2 + 5\sqrt{2\log(8/\delta)}\right)$, *for any* $\delta \in [0, \delta]$

$$\mathbb{P}_{\mathbf{S}\sim\mathcal{D}^N}\left(\mathcal{L}(\hat{\Sigma}) \leq \mathcal{L}(\Sigma_\star) + \frac{R_2(\delta)}{\sqrt{N}}\right) \geq 1 - \delta.$$

That is, the bound of Corollary 11 is a version of the bound (9). In particular, we can determine the minimal number of data points $N_{min} = \frac{R_2^2(\delta)}{\epsilon^2}$, such that the difference between the performance of the minimal prediction error model and that of the best possible model is below the desired threshold ϵ.

Potential application to parameter estimation As it was mentioned in Sect. 3.2, bounds of the type (9), for instance, the one of Corollary 11, can be used to show consistency of the learning algorithm based on empirical risk minimization, if the latter is viewed as a parameter estimation algorithm. That is, for linear systems it is known that models with a small prediction error tend to be close to the true one, under suitable identifiability assumptions [9, Theorem 8.3]. For LPV systems, this problem requires further research, but we expect similar results.

Comparison of PAC bounds with asymptotic consistency results in system identification Classical results in system identification aim at showing that for large enough N, an identification algorithm which choose models with a small empirical loss small will result in models with a small true loss, e.g., [9, Lemma 8.2] for LTI systems. We are not aware of similar results for continuous-time LPV systems. However, even for LTI systems, Theorem 10 is different from these classical results, as the latter says little about how large N should be so that the empirical loss upper bounds the true loss with a certain accuracy. On the downside, classical results hold for data which originate from a single time-series, while the present paper assumes presence of multiple independently sampled time-series.

Limitation of the i.i.d. assumption In contrast to the widespread practice in system identification, we assume access to several i.i.d. samples of input, output and scheduling signals. While this assumption is somewhat restrictive, it is still applicable in many scenarios. In particular, the i.i.d. assumption is reasonable when learning Neural ODEs [13, 23, 29, 30] and SSMs [14, 66]. Furthermore, deriving PAC bounds in this setting is a first step toward PAC bounds for the case of a single long time signal. Indeed, as it was mentioned before in Sect. 1, it is possible to extend PAC bounds based on Rademacher complexity to the non i.i.d. setting [21, 22]. We conjecture that it would be possible to combine the results of the presented paper with those of [21, 22] in order to extend them to data originating from a single time series.

Sampling, persistence of excitation, etc. Since PAC bounds hold for any identification algorithms, we did not make any assumptions on such, otherwise crucial issues, as persistence of excitation, sampling, etc.

Discussion on the bound: dependence on N and T, and the role of stability The bound in Theorem 10 tends to zero as N grows to infinity and is also independent of the integration time T, a consequence of assuming stability of the models. The latter

is a significant improvement compared to prior work [23, 28–30]. We conjecture that some form of stability is also necessary for time-independent bounds, as intuitively in case of unstable systems small modeling errors may lead to a significant increase of the prediction error in the long run. The bound grows linearly with the maximal H_2 norm of the elements of $\mathcal{E}$, with the maximal possible value of the true outputs, and with the maximal energy of the inputs. Roughly speaking, as the minimal degree of stability over $\mathcal{E}$ increases, the maximal H_2 norm over $\mathcal{E}$ tends to decrease, thus the more stable the considered system class, the tighter the bound.

Relationship between the quality of approximation and the PAC bounds for LPV systems As mentioned in Sect. 3.2, there is generally a trade-off between the generalization ability of $\mathcal{E}$, captured by $R(\delta)$, and the approximation error of $\mathcal{E}$, defined as $\inf_{\Sigma\in\mathcal{E}} \mathcal{L}(\Sigma)$. In the ideal case, $\mathcal{E}$ is expressive enough to closely approximate the true system, yielding $\inf_{\Sigma\in\mathcal{E}} \mathcal{L}(\Sigma) \approx 0$, but not so rich as to include models with a very high H_2 norm (i.e., $c_{\mathcal{E}}$ is not too large). If $c_{\mathcal{E}}$ is too small, then the behavior of models on any concrete dataset used for learning is closer to their behavior on unseen data, so fewer data points are needed to learn a model whose true loss is close to the empirical loss or the approximation error of $\mathcal{E}$. However, in this case, the latter — and hence the quality of approximation provided by any learned model — can itself be large.

An extreme trivial example occurs when the $C(\mathbf{p}(t))$ matrices of all LPV models in $\mathcal{E}$ are zero, so that the maximal H_2 norm $c_{\mathcal{E}}$ is zero. In this case, the bounding term $R(\delta)$ in Theorem 10 becomes zero, and the true loss depends only on the marginal distribution of the true output y, while the empirical loss $\mathcal{L}_N^{\mathbf{S}}(\Sigma)$ for any $\Sigma \in \mathcal{E}$ depends solely on the true outputs in the dataset $\mathbf{S}$. Here, the behavior of any model on any concrete dataset fully determines its behavior outside it, since in both cases the models produce constant zero output.

Conversely, if $\mathcal{E}$ is large enough to contain a good representation of the underlying distribution $\mathcal{D}$, then $c_{\mathcal{E}}$ may be large and one may need more data points to avoid a large generalization gap. As an extreme example, let us assume that the underlying physical process can be exactly modeled by an LPV system $\Sigma_\star$ which satisfies our assumptions, and the H_2 norm of $\Sigma_\star$ is at most $c_\star$. Let $\mathcal{E}$ be the class of all LPV systems satisfying our assumptions, with H_2 norm at most $c_{\mathcal{E}}$ such that $c_\star \leq c_{\mathcal{E}}$. However, lacking knowledge of the true system, we may select $c_{\mathcal{E}}$ to be large. Clearly, the approximation error satisfies $\inf_{\Sigma\in\mathcal{E}} \mathcal{L}(\Sigma) = 0$ since the true system $\Sigma_\star \in \mathcal{E}$, while due to the large number of systems included in $\mathcal{E}$, for any finite sample $\mathbf{S}$ we can pick a model $\Sigma \in \mathcal{E}$ that behaves very differently outside $\mathbf{S}$ than on $\mathbf{S}$. This is reflected by the large value of $c_{\mathcal{E}}$.

In short, our results bound the prediction error of the learned model on new data relative to either the prediction error on the data used for learning or to the best prediction error achievable by the given set of parametrizations. These results do not provide any guarantee about the approximation error. Establishing similar bounds on the latter is a separate problem, which is beyond the scope of this paper.

Multiple outputs Under suitable assumptions on the loss function, which are satisfied for quadratic and 1-norm losses, the results can be extended to multiple outputs by

applying Theorem 10 separately for each output component, with confidence $1-n_{\text{out}}\delta$, see Appendix C for a detailed derivation.

Discretization of the input data According to Remark 2, the bound in Theorem 10 holds for time sampled trajectories of $\mathbf{u}$ and $\mathbf{p}$ as long as the samples interpreted as piecewise constant functions are in $\mathcal{U}$ and $\mathcal{P}$. Furthermore, the loss function described by Equation (3) ensures the theorem holds for time sampled output trajectories.

The discretization algorithm has no particular effect on the bound. Intuitively, by its nature Theorem 10 does not say anything about how accurately we learn the underlying true LPV system. Instead, it gives a bound on the generalization error, i.e., on the difference between the empirical and true errors as defined in Sect. 3. It is a consistency theorem in the sense that it ensures that given enough data points, regardless of the discretization method on the input trajectories (as long as the dataset remains i.i.d.), the model will behave identically on seen and unseen data with high probability. That is, our results suggest the learning algorithms for any sampling are consistent.

An unsatisfactory time-sampling method could cause the following. First, it could increase the smallest possible true loss that is achievable in the considered learning setup, namely $\inf_{\Sigma\in\mathcal{E}} \mathcal{L}(\Sigma)$ in Equations (8) and (9). Second, in our setup, we assume that $\mathcal{D}$ is unknown but it is already implicitly given by the physical constraints of the learning experiment. These physical constraints, including the choice of the sampling time, may influence the unknown distribution $\mathcal{D}$, for example, by restricting it to inputs and scheduling signals which are piecewise constants on sampling intervals. In this case, $\mathcal{L}(\Sigma)$ will capture the prediction error only for such inputs and scheduling. If the loss function ℓ evaluates the prediction error only at sampling times, then the true loss captures the prediction error on unseen data only at sampling times.

Theorem 10 essentially describes what properties of the learning problem influence the additive bounding terms in Equations (8) and (9). Under concrete circumstances, analysis of learnability requires analyzing the smallest achievable true loss, the adequacy of the loss function and the additional bounding terms. The statement in Theorem 10 is about the latter.

Discretized LPV systems It is also possible to discretize the continuous-time LPV system, resulting in a discrete-time LPV. However, discretizing an affine LPV system does not generally preserve the affine dependence [67]. Furthermore, the sufficient condition for stability and the existence of a finite H_2 norm is different in the discrete-time case. While stability and the existence of H_2 norm can be formulated in terms of LMIs for the discrete-time case and we conjecture that the proof technique could also be extended, such an extension is out of the scope of the paper. It is worth noting that the extension of the results of the paper to discrete-time LPV systems would be useful for the class of deep selective State-Space Models (selective SSMs), in particular, for the Mamba architecture. These models have recently gained popularity and they have turned out to be very promising for several text and video processing tasks [14, 68].

6 Proof of Theorem 10

The proof is of two parts. In the first part, we prove Theorem 10 for the special case of true and empirical losses described by Equation (2). In the second part, we show that the general case can be reduced the first part.

First part In this case, the definitions of $\mathcal{L}(\Sigma)$ and $\mathcal{L}_N^{\mathbf{S}}(\Sigma)$ have the forms

$$\mathcal{L}(\Sigma) = \mathbb{E}_{(\mathbf{u},\mathbf{p},y)\sim\mathcal{D}}\left[\ell(y_\Sigma(\mathbf{u},\mathbf{p})(T), y(T))\right]$$
$$\mathcal{L}_N^{\mathbf{S}}(\Sigma) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_\Sigma(\mathbf{u}_i,\mathbf{p}_i)(T), y_i(T)).$$

The main component of the proof is the estimation of the Rademacher complexity of the class of LPV systems.

Recall from [10, Def. 26.1] that the Rademacher complexity of a bounded set $\mathcal{A} \subset \mathbb{R}^m$

is defined as

$$R(\mathcal{A}) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{a\in\mathcal{A}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i a_i\right],$$

where the random variables σ_i are i.i.d such that $\mathbb{P}[\sigma_i = 1] = \mathbb{P}[\sigma = -1] = 0.5$. The Rademacher complexity of a set of functions $\mathcal{F}$ over a set of samples $\mathbf{S} = \{\mathbf{s}_1 \dots \mathbf{s}_m\}$ is defined as $R_{\mathbf{S}}(\mathcal{F}) = R(\{(f(\mathbf{s}_1), \dots, f(\mathbf{s}_m)) \mid f \in \mathcal{F}\})$.

Intuitively, Rademacher complexity measures the richness of a set of functions, see e.g., chapter 26 in [10], and can be used for deriving PAC bounds [10, Theorem 26.5] for general models. Below we restate this result for LPV systems.

Theorem 12 *Let $L_0(T)$ denote the set of functions of the from $(\mathbf{u},\mathbf{p},y) \mapsto \ell(y_\Sigma(\mathbf{u},\mathbf{p})(T), y(T))$ for $\Sigma \in \mathcal{E}$. Let $B(T)$ be such that the functions from $L_0(T)$ all take values from the interval $[0, B(T)]$. Then for any $\delta \in (0,1)$ we have*

$$\mathbb{P}_{\mathbf{S}\sim\mathcal{D}^N}\left(\forall\Sigma \in \mathcal{E} : \mathcal{L}(\Sigma) - \mathcal{L}_N^{\mathbf{S}}(\Sigma) \le 2R_{\mathbf{S}}(L_0(T)) + 4B(T)\sqrt{\frac{2\log(4/\delta)}{N}}\right) \ge 1-\delta.$$

The proof of Theorem 10 follows from Theorem 12, by first bounding the Rademacher complexity of $R_{\mathbf{S}}(L_0(T))$ and then bounding the constant $B(T)$.

Step 1 We show that

$$R_{\mathbf{S}}(L_0(T)) \le \frac{c}{\sqrt{N}}$$

Consider the class $\mathcal{F}$ of output response functions $(\mathbf{u},\mathbf{p}) \mapsto y_\Sigma(\mathbf{u},\mathbf{p})(T)$ for $\Sigma \in \mathcal{E}$ and the corresponding Rademacher complexity $R_{\mathbf{S}}(\mathcal{F})$. By [10, Lemma 26.9] and Assumption 4, $R_{\mathbf{S}}(L_0(T)) \le c_\ell R_{\mathbf{S}}(\mathcal{F})$, hence it is enough to bound $R_{\mathbf{S}}(\mathcal{F})$. For the latter, we need the following Lemma.

Lemma 13 *There exists a Hilbert space $\mathcal{H}$ such that for every $\Sigma \in \mathcal{E}$ there exists $w^{T,\Sigma} \in \mathcal{H}$ and for every $(\mathbf{u}, \mathbf{p}) \in \mathcal{U} \times \mathcal{P}$ there exists $\varphi^{T,\mathbf{u},\mathbf{p}} \in \mathcal{H}$, such that*

$$y_\Sigma(\mathbf{u}, \mathbf{p})(T) = \langle w^{T,\Sigma}, \varphi^{T,\mathbf{u},\mathbf{p}} \rangle_{\mathcal{H}}$$

and

$$\left\| \varphi^{T,\mathbf{u},\mathbf{p}} \right\|_{\mathcal{H}} \leq c_{\mathcal{U}}(n_p + 1)$$
$$\left\| w^{T,\Sigma} \right\|_{\mathcal{H}} \leq \|\Sigma\|_{\lambda,H_2}.$$

Proof Let $\mathcal{V}$ be the vector space consisting of sequences of the form

$$f = \{ f_{i_q,i_r,I} \mid I \in I_k; i_q, i_r \in [n_\mathrm{p}]_0 \}_{k=0}^{\infty}$$

such that $f_{i_q,i_r,I} : \Delta_{k+1}^T \mapsto \mathbb{R}^{1 \times n_\mathrm{in}}$ is measurable. In other words, every element $f \in \mathcal{V}$ is a sequence of functions indexed by i_q, i_r and I which depends on k. The actual ordering of the sequence members is not relevant and is considered the same for all $f \in \mathcal{V}$. Our goal here is to define a suitable mathematical structure that incorporates the weighted Volterra-kernels as well as the weighted scheduling-input products, for all possible values of k. As it turns out, we can define an inner product over the above defined $\mathcal{V}$ in a way that the subset of $\mathcal{V}$ formed by the vectors of finite norm gives rise to a Hilbert space $\mathcal{H}$ in a way that $\mathcal{H}$ contains both the Volterra-kernels and scheduling-input products.

For any $f, g \in \mathcal{V}$ let us define the series

$$\langle f, g \rangle = \sum_{i_q,i_r=0}^{n_\mathrm{p}} \sum_{k=0}^{\infty} \sum_{I \in I_k} \int_{\Delta_{k+1}^T} f_{i_q,i_r,I}(\boldsymbol{\tau}) g_{i_q,i_r,I}(\boldsymbol{\tau})^T d\boldsymbol{\tau}.$$

and for any $f \in \mathcal{V}$, let us denote by $\|f\|^2 = \langle f, f \rangle$. Let $\mathcal{H}$ consists of those elements $f \in \mathcal{V}$ for which the series $\|f\|^2$ is convergent. Then for any $f, g \in \mathcal{H}$, the series $\langle f, g \rangle$ is absolutely convergent. Let us denote its limit by $\langle f, g \rangle_{\mathcal{H}}$. Then $\mathcal{H}$ is a Hilbert space with the scalar product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, and we denote by $\|\cdot\|_{\mathcal{H}}$ the corresponding norm.

Let $\lambda \geq n_\mathrm{p}$ be such that Assumption 1 is satisfied. Let

$$w^{T,\Sigma} = \{ (w^{\lambda}_{i_q,i_r,I})|_{\Delta_{k+1}^T} \mid I \in I_k; i_q, i_r \in [n_\mathrm{p}]_0 \}_{k=0}^{\infty}$$
$$\varphi^{T,\mathbf{u},\mathbf{p}} = \{ (\varphi^{\lambda}_{i_q,i_r,I})^T \mid I \in I_k; i_q, i_r \in [n_\mathrm{p}]_0 \}_{k=0}^{\infty}.$$

We show that $w^{T,\Sigma} \in \mathcal{H}$ and $\varphi^{T,\mathbf{u},\mathbf{p}} \in \mathcal{H}$ by proving that $\left\| w^{T,\Sigma} \right\|_{\mathcal{H}}^2$ and $\left\| \varphi^{T,\mathbf{u},\mathbf{p}} \right\|_{\mathcal{H}}^2$ are finite and bounded as claimed in the Lemma.

By the definition of $w^{T,\Sigma}$ it is clear that

$$\left\| w^{T,\Sigma} \right\|_{\mathcal{H}}^2 \leq \|\Sigma\|_{\lambda,H_2}^2 < +\infty$$

due to Assumption 2.

As for $\varphi^{T,\mathbf{u},\mathbf{p}}$, due to $\mathbf{p}$ taking values in $[-1,1]^{n_\mathrm{p}}$,

$$\left\|\varphi^{\lambda}_{i_q,i_r,I}(\boldsymbol{\tau})\right\|_2^2 \le \|\mathbf{u}(T-\tau_{k+1})\|_2^2\,\mathrm{e}^{-\lambda\tau_{k+1}}$$

for any $I \in I_k$, $\boldsymbol{\tau} = (\tau_{k+1},\dots,\tau_1) \in \Delta^T_{k+1}$, $k \ge 0$. Hence, by setting $\int_{\Delta^t_k} d\boldsymbol{\tau} = 1$ for $k = 0$ and any $t > 0$, we obtain

$$\begin{aligned}\left\|\varphi^{T,\mathbf{u},\mathbf{p}}\right\|_{\mathcal{H}}^2 &= \sum_{i_q,i_r=0}^{n_\mathrm{p}}\sum_{k=0}^{\infty}\sum_{I\in I_k}\int_{\Delta^T_{k+1}}\left\|\varphi^{\lambda}_{i_q,i_r,I}(\boldsymbol{\tau})\right\|_2^2 d\boldsymbol{\tau}\\ &\le (n_\mathrm{p}+1)^2\int_0^T \|\mathbf{u}(T-t)\|_2^2\,\mathrm{e}^{-\lambda t}\left(\sum_{k=0}^{\infty}\sum_{I\in I_k}\int_{\Delta^t_k} d\boldsymbol{\tau}\right)\mathrm{d}t\end{aligned}$$

For iterated integrals we have the well-known inequality (see [15, Chapter 3.1])

$$\int_{\Delta^t_k} d\boldsymbol{\tau} \le \frac{t^k}{k!},$$

therefore (due to $I_k = [n_p]^k$) we have

$$\begin{aligned}&(n_\mathrm{p}+1)^2\int_0^T \|\mathbf{u}(T-t)\|_2^2\,\mathrm{e}^{-\lambda t}\left(\sum_{k=0}^{\infty}\sum_{I\in I_k}\int_{\Delta^t_k} d\boldsymbol{\tau}\right)\mathrm{d}t\\ &\le (n_\mathrm{p}+1)^2\int_0^T \|\mathbf{u}(T-t)\|_2^2\,\mathrm{e}^{-\lambda t}\left(\sum_{k=0}^{\infty}\frac{n_p^k t^k}{k!}\right)\mathrm{d}t\\ &= (n_\mathrm{p}+1)^2\int_0^T \|\mathbf{u}(T-t)\|_2^2\,\mathrm{e}^{t(n_\mathrm{p}-\lambda)}\mathrm{d}t\\ &\le (n_\mathrm{p}+1)^2\,\|\mathbf{u}\|^2_{L^2([0,T],\mathbb{R}^{n_\mathrm{in}})} < (n_\mathrm{p}+1)^2 c_{\mathcal{U}}^2 < +\infty\end{aligned}$$

where the last inequality follows from the choice of $\lambda \ge n_\mathrm{p}$ and Assumption 3.

Finally, by Lemma 6, $y_\Sigma(\mathbf{u},\mathbf{p})(T) = \langle w^{T,\Sigma}, \varphi^{T,\mathbf{u},\mathbf{p}}\rangle_{\mathcal{H}}$. □

Using Lemma 13 and [10, Lemma 26.10] we have

$$R_{\mathbf{S}}(\mathcal{F}) \le \frac{c_{\mathcal{E}}c_{\mathcal{U}}(n_\mathrm{p}+1)}{\sqrt{N}}$$

yielding

$$R_{\mathbf{S}}(L_0(T)) \le c_\ell R_{\mathbf{S}}(\mathcal{F}) \le \frac{c}{\sqrt{N}}.$$

Step 2: Bounding $B(T)$ By Assumption 4, we have

$$|\ell(y_\Sigma(\mathbf{u},\mathbf{p})(T), y(T))| \le 2c_\ell \max\{|y_\Sigma(\mathbf{u},\mathbf{p})(T)|, |y(T)|\} \le 2c_\ell \max\{c_{\mathcal{U}} c_{\mathcal{E}}, c_{\mathcal{Y}}\} \le c$$

The second-to-last inequality follows from applying Lemma 8 and Assumption 3 along with $n_{\mathrm{p}} \ge 1$.

Finally, the first part follows from the bounds obtained in Step 1. and Step 2. together with Theorem 12.

Second part Now we will reduce the general case to the first part. Let F_0 be the set of functions of the form $(\mathbf{u},\mathbf{p},y) \to \int_{[0,T]} \ell(y_\Sigma(\mathbf{u},\mathbf{p})(\tau), y(\tau))\, d\mu(\tau)$ defined analogously to Theorem 12. This is a generalization of $L_0(T)$ defined in Theorem 12, as the image may depend on the whole trajectories y_Σ and y, not just their values at T. We have

$$\begin{aligned}
R_{\mathbf{S}}(F_0) &= \mathbb{E}_\sigma\left[\sup_{\Sigma\in\mathcal{E}} \frac{1}{N}\sum_{i=1}^N \sigma_i \int_{[0,T]} \ell(y_\Sigma(\mathbf{u}_i,\mathbf{p}_i), y_i)(\tau) d\mu(\tau)\right] \\
&\le \mathbb{E}_\sigma\left[\int_{[0,T]} \sup_{\Sigma\in\mathcal{E}} \frac{1}{N}\sum_{i=1}^N \sigma_i \ell(y_\Sigma(\mathbf{u}_i,\mathbf{p}_i), y_i)(\tau) d\mu(\tau)\right] \\
&= \int_{[0,T]} \mathbb{E}_\sigma\left[\sup_{\Sigma\in\mathcal{E}} \frac{1}{N}\sum_{i=1}^N \sigma_i \ell(y_\Sigma(\mathbf{u}_i,\mathbf{p}_i), y_i)(\tau)\right] d\mu(\tau) \\
&\le \mu([0,T]) \sup_{t\in[0,T]} \mathbb{E}_\sigma\left[\sup_{\Sigma\in\mathcal{E}} \frac{1}{N}\sum_{i=1}^N \sigma_i \ell(y_\Sigma(\mathbf{u}_i,\mathbf{p}_i), y_i)(t)\right] \\
&= \mu([0,T]) \sup_{t\in[0,T]} R_{\mathbf{S}}(L_0(t))
\end{aligned}$$

where $L_0(t)$ is defined the same as $L_0(T)$ from Theorem 12 with $T = t$.

As we require μ to be normalized by T, we have $\mu([0,T]) \le 1$.

Let $B([0,T]) = \sup_{t\in[0,T]} \sup_{\mathbf{u},\mathbf{p},y} \ell(y_\Sigma(\mathbf{u},\mathbf{p}), y)(t)$. By Theorem 12 and using the above proven inequality $R_{\mathbf{S}}(F_0) \le \mu([0,T]) \sup_{t\in[0,T]} R_{\mathbf{S}}(L_0(t))$, we have

$$\mathbb{P}_{\mathbf{S}\sim\mathcal{D}^N}\left(\forall\Sigma\in\mathcal{E} : \mathcal{L}(\Sigma) - \mathcal{L}_N^{\mathbf{S}}(\Sigma) \le 2\sup_{t\in[0,T]} R_{\mathbf{S}}(L_0(t)) + 4B([0,T])\sqrt{\frac{2\log(4/\delta)}{N}}\right) \ge 1-\delta$$

and we can bound $R_{\mathbf{S}}(L_0(t))$ for any $t \in [0,T]$ the same way we bound $R_{\mathbf{S}}(L_0(T))$ in the first part, i.e., $R_{\mathbf{S}}(L_0(t)) \le \frac{c}{\sqrt{N}}$.

In other words, we can apply the first part with $T = t$ and as the obtained bound is independent of T, the bounding term for $R_{\mathbf{S}}(L_0(t))$ and hence for $\sup_{t\in[0,T]} R_{\mathbf{S}}(L_0(t))$ is the same as in the first part ($c/\sqrt{N}$) and the statement of Theorem 10. Furthermore, the calculation in Step 2. of the first part holds for $B([0,T])$. This concludes the proof of the second part and the theorem.

7 Numerical example

We considered a parameterized family $\mathcal{E}$ of LPV systems $\Sigma(\boldsymbol{\theta}) = (\mathbf{A}_i(\boldsymbol{\theta}), \mathbf{B}_i(\boldsymbol{\theta}), \mathbf{C}_i(\boldsymbol{\theta}))_{i=0}^{n_\mathrm{p}}$ where $n_\mathrm{p} = 1$, $\Theta \subseteq \mathbb{R}^3$, and for all $\boldsymbol{\theta} = \begin{pmatrix}\theta_1, \theta_2, \theta_3\end{pmatrix}^T \in \Theta$,

$$\mathbf{A}_0(\boldsymbol{\theta}) = \begin{pmatrix} -\frac{1}{\theta_1} & 0 \\ 1 & \frac{-1}{\theta_1} \end{pmatrix}, \quad \mathbf{A}_1(\boldsymbol{\theta}) = \begin{pmatrix} 0 & \theta_2 \\ 0 & 0 \end{pmatrix}, \quad \mathbf{B}_0(\boldsymbol{\theta}) = \begin{pmatrix} \theta_3 \\ 0 \end{pmatrix},$$
$$\mathbf{B}_1(\boldsymbol{\theta}) = \begin{pmatrix}0, 0\end{pmatrix}^T, \quad \mathbf{C}_0(\boldsymbol{\theta}) = \begin{pmatrix}0, 1\end{pmatrix}, \quad \mathbf{C}_1(\boldsymbol{\theta}) = \begin{pmatrix}0, 0\end{pmatrix}$$
$$\Theta = \{\boldsymbol{\theta} \in \mathbb{R}^3 \mid \|\Sigma(\boldsymbol{\theta})\|_{H_2,\lambda} < c_{\mathcal{E}}\},\ \theta_2 = \theta_3, \lambda = 1.2 \text{ and } c_{\mathcal{E}} = 2.$$

We considered the training dataset $\mathbf{S} = \{\mathbf{u}_i, p_i, y_i\}_{i=1}^N$, and the validation dataset $\mathbf{S}_{\mathrm{val}} = \{\mathbf{u}_i, p_i, y_i\}_{i=1}^M$, $M = 10^4$ where $\mathbf{u}_i$, p_i are signals defined on $[0, T] = [0, 0.45]$, $\mathbf{u}_i$, p_i are right continuous and they are constant on each interval $[kT_s, (k+1)T_s)$, $k = 0, \ldots, k_\star$, with $T_s = 0.01$ and $k_\star = 45$. Moreover, the values $p_i(kT_s)$ and $\mathbf{u}_i(kT_s)$, $i = 1, \ldots, N$, $k = 1, \ldots, k_\star$ were sampled independently from the uniform distribution on $[0, 3]$ and $[0, 30]$, respectively, and their means were substracted to render them zero mean. The signal y_i is the output of the system $\Sigma(\boldsymbol{\theta}_\star)$, where $\boldsymbol{\theta}_\star = \begin{pmatrix}0.1 & -1.87, & -153.15\end{pmatrix}^T$, to which we added a zero mean Gaussian white noise with variance 0.05. Note, that the data generator $\Sigma(\boldsymbol{\theta}_\star)$ does not belong to the set of parameterizations $\mathcal{E}$.

We apply the following learning algorithm to the parametrized family $\mathcal{E}$. Notice, that for $\theta \in \Theta$ the output $y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)$ satisfies

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(t) = -\frac{2}{\theta_1}\frac{\mathrm{d}}{\mathrm{d}t} y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(t) - \left(\frac{1}{\theta_1^2} - \theta_2\right) y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(t) + \theta_2 \mathbf{u}(t) \tag{11}$$

Indeed, it follows that if $\mathbf{x}(t) = \begin{pmatrix}x_1(t)\ x_2(t)\end{pmatrix}^T$ is the state of $\Sigma(\boldsymbol{\theta})$ from the zero initial state, corresponding to input $\mathbf{u}$ and scheduling p, then $\mathbf{x}_2(t) = y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(t)$ and $\frac{\mathrm{d}}{\mathrm{d}t} y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(t) = \mathbf{x}_1(t) - \frac{1}{\theta_1} y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(t)$. By taking the derivative of the latter equation, using the equation for $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}_1(t)$ determined by $\Sigma(\boldsymbol{\theta})$ and $\mathbf{x}_1(t) = \frac{\mathrm{d}}{\mathrm{d}t} y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(t) + \frac{1}{\theta_1} y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(t)$, (11) follows.

If we apply the first-order Euler approximation scheme, then

$$\frac{\mathrm{d}}{\mathrm{d}t} y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(t)|_{t=kT_s} \sim \frac{1}{T_s}\left(y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)((k+1)T_s) - y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(kT_s)\right)$$
$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(t)|_{t=kT_s} \sim \frac{1}{T_s^2}\Big(y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)((k+2)T_s) - 2y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)((k+1)T_s) +$$
$$+ y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(kT_s)\Big)$$

and therefore, from (11), by standard algebraic manipulations, we obtain the following approximate input–output representation

$$y_{\Sigma(\boldsymbol{\theta})}((k+2)T_s) \sim \boldsymbol{\phi}_{k+2}(\boldsymbol{\theta})\mathbf{a}(\boldsymbol{\theta}) \tag{12}$$

or, alternatively,

$$y_{\Sigma(\boldsymbol{\theta})}((k+2)T_s) = \boldsymbol{\phi}_{k+2}(\boldsymbol{\theta})\mathbf{a}(\boldsymbol{\theta}) + v_k \tag{13}$$

where the term v_k depends on $\mathbf{u}$, p, $\boldsymbol{\theta}$ and the approximation error $\frac{\mathrm{d}}{\mathrm{d}t} y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(t)|_{t=kT_s} - \frac{1}{T_s}\left(y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)((k+1)T_s) - y_{\Sigma(\boldsymbol{\theta})}(\mathbf{u}, p)(kT_s)\right)$, and

$$\boldsymbol{\phi}_{k+1}(\boldsymbol{\theta}) = \left(y_{\Sigma(\boldsymbol{\theta})}((k+1)T_s),\ y_{\Sigma(\boldsymbol{\theta})}(kT_s),\ \ \mathbf{u}(kT_s)\right),$$

while the function $\mathbf{a}(\boldsymbol{\theta})$ is defined by

$$\mathbf{a}(\boldsymbol{\theta}) = \left(2(1-\tfrac{T_s}{\theta_1}),\ -(1-\tfrac{T_s}{\theta_1})^2,\ T_s^2\theta_2,\ \theta_2 T_s^2\right)^T.$$

That is, if the data is generated by an element $\Sigma(\boldsymbol{\theta}_\star)$ of $\mathcal{E}$, then it should satisfy the linear regression

$$y_i(kT_s) = \boldsymbol{\phi}_{i,k}\mathbf{a}(\boldsymbol{\theta}_\star) + v_{i,k}, \quad k \geq 2,$$

where $v_{i,k}$ is an error term introduced by Euler discretization and

$$\boldsymbol{\phi}_{i,k} = \left(y_i((k-1)T_s),\ y_i((k-2)T_s),\ y_i((k-2)T_s)p_i((k-2)T_s),\ \mathbf{u}_i((k-2)T_s)\right).$$

Finally, we compute an estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}_\star$ by solving the least squares minimization problem

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left(\frac{1}{N}\sum_{i=1}^{N}\sum_{k=2}^{k_\star} \|y_i(kT_s) - \boldsymbol{\phi}_{i,k}\mathbf{a}(\boldsymbol{\theta})\|_2^2\right) \tag{14}$$

for $k = 1, \ldots, k_\star$, $i = 1, \ldots, N$, $k_\star = 45$.

Note that in Equation (14) the arg min should be taken over $\boldsymbol{\theta} \in \Theta$. However, finding the least square estimate $\hat{\boldsymbol{\theta}}$ together with the constraint $\boldsymbol{\theta} \in \Theta$ is very difficult. On the other hand, the problem formulation implies $\boldsymbol{\theta}_\star \in \Theta$ and we expect the least squares estimator $\hat{\boldsymbol{\theta}}$ to be close to $\boldsymbol{\theta}_\star$ due to the formulation of the least squares problem, implying also $\hat{\boldsymbol{\theta}} \in \Theta$. The latter is numerically verified for each N, hence we have $\hat{\boldsymbol{\theta}} \in \Theta$.

We chose $\delta = 0.05$ and we computed $R(\delta)$. We then computed $\mathcal{L}_N^{\mathbf{S}}(\Sigma(\hat{\boldsymbol{\theta}}))$ and $\mathcal{L}(\Sigma(\hat{\boldsymbol{\theta}}))$ for the case described by Equation (2) in Sect. 3, using the norm as loss $\ell(y, y') = \|y - y'\|_1$, and by approximating $\mathcal{L}(\Sigma(\hat{\boldsymbol{\theta}}))$ by the empirical prediction error

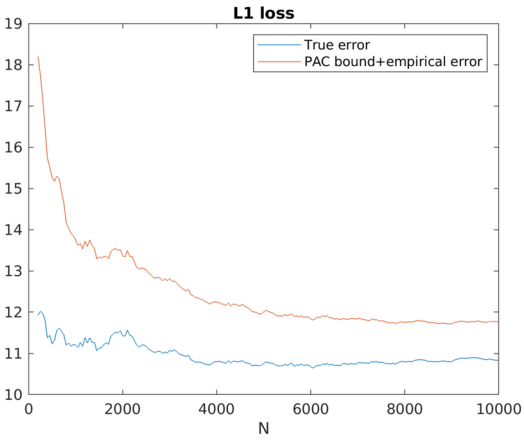

Fig. 1 Difference between the estimated and the true error. The estimated error is the sum of the empirical error measured on a finite set with N elements and the PAC bound

$\mathcal{L}_M^{\mathbf{S}_{\text{val}}}(\Sigma(\hat{\boldsymbol{\theta}}))$ on the validation dataset. As Figure 1 shows, $\mathcal{L}(\Sigma(\hat{\boldsymbol{\theta}})) \leq \mathcal{L}_N^{\mathbf{S}}(\Sigma(\hat{\boldsymbol{\theta}})) + \frac{R(\delta)}{\sqrt{N}}$ holds, and we can see that the maximum of the true loss is strictly greater, than the minimum of the bound, as, e.g., the bound for $N = 10000$ is strictly smaller, than the true loss $\mathcal{L}(\Sigma(\hat{\boldsymbol{\theta}}))$ for $N = 200$. In other words, the bound is non-vacuous. The MATLAB code of the example can be found at https://github.com/danielracz/lpv_mcss.

8 Conclusion

In this paper, we examined LPV systems within the confines of statistical learning theory and derived a PAC bound on the generalization error under stability conditions. The central element of the proof is the application of Volterra-series expansion in order to upper bound the Rademacher complexity of LPV systems. Further research is directed toward extending these methods to more general models, possibly exploiting the powerful approximation properties of LPV systems.

Appendix A: Proof of Lemma 6

Proof Consider the family of bilinear systems indexed by $i_q, i_r \in [n_{\text{p}}]$ and a fixed $t \in [0, T]$, defined as

$$\begin{aligned} \dot{\mathbf{s}}(\tau) &= \left(\mathbf{A}_0 + \frac{\lambda}{2}\mathbf{I} + \sum_{i=1}^{n_{\text{p}}} p_i(\tau + T - t)\mathbf{A}_i\right)\mathbf{s}(\tau) \\ \mathbf{s}(0) &= \mathbf{B}_{i_r}\mathbf{u}(T-t)\mathrm{e}^{-\frac{\lambda}{2}t} \\ y_{i_q,i_r}^{\mathbf{s},t}(\tau) &= \mathbf{C}_{i_q}\mathbf{s}(\tau). \end{aligned} \tag{A1}$$

From the Volterra-series representation [15] of bilinear systems we have

$$y^{\mathbf{s},t}_{i_q,i_r}(\tau) = \left[w_{i_q,i_r,\emptyset}(\tau) + \sum_{k=1}^{\infty}\sum_{I\in I_k}\int_{\Delta_k^\tau} w^{\lambda,\tau}_{i_q,i_r,I}(\boldsymbol{\upsilon})p_I((t,\upsilon^T)^T)d\boldsymbol{\upsilon}\right]\mathbf{u}(T-t)\mathrm{e}^{-\frac{\lambda}{2}t} \tag{A2}$$

where $\tau \in [0,t]$, p_I is as in Sect. 4 and $w^{\lambda,\tau}_{i_q,i_r,I}(\boldsymbol{\upsilon}) = w^{\lambda}_{i_q,i_r,I}((\tau,\boldsymbol{\upsilon}^T)^T)$, for all $I \in I_k$ and $\boldsymbol{\upsilon} \in \Delta_k^\tau$, $k > 0$. By [1, Chapter 3.3.1.1] we have

$$y_\Sigma(\mathbf{u},\mathbf{p})(T) = \int_0^T \mathbf{C}(\mathbf{p}(T))\boldsymbol{\Phi}(T,T-t)\mathbf{B}(\mathbf{p}(T-t))\mathbf{u}(T-t)\mathrm{d}t \tag{A3}$$

where $\boldsymbol{\Phi}(r,r_0)$ is the fundamental matrix of the system

$$\dot{\mathbf{z}}(r) = \mathbf{A}(\mathbf{p}(r))\mathbf{z}(r).$$

Since the fundamental matrix $\boldsymbol{\Phi}_\lambda(r,r_0)$ of a slightly different system, defined as

$$\dot{\mathbf{z}}(r) = (\mathbf{A}(\mathbf{p}(r)) + \frac{\lambda}{2}\mathbf{I})\mathbf{z}(r),$$

satisfies

$$\boldsymbol{\Phi}_\lambda(r,r_0) = \mathrm{e}^{\frac{\lambda}{2}(r-r_0)}\boldsymbol{\Phi}(r,r_0),$$

we have

$$\begin{aligned}\mathbf{s}(\tau) &= \boldsymbol{\Phi}_\lambda(T-t+\tau,T-t)\mathbf{B}_{i_r}\mathbf{u}(T-t)\mathrm{e}^{-\frac{\lambda}{2}t}\\ &= \mathrm{e}^{-\frac{\lambda}{2}(t-\tau)}\boldsymbol{\Phi}(T-t+\tau,T-t)\mathbf{B}_{i_r}\mathbf{u}(T-t).\end{aligned} \tag{A4}$$

Using the definitions of $\mathbf{C}(\mathbf{p}(T))$ and $\mathbf{B}(\mathbf{p}(T-t))$ from Sect. 2, we can substitute Equation (A4) into Equation (A3) and obtain

$$y_\Sigma(\mathbf{u},\mathbf{p})(T) = \int_0^T \sum_{i_q,i_r=0}^{n_\mathrm{p}} p_{i_q}(T)p_{i_r}(T-t)y^{\mathbf{s},t}_{i_q,i_r}(t)\,\mathrm{d}t. \tag{A5}$$

Substituting Equation (A2) into Equation (A5) together with $\{(t,\boldsymbol{\upsilon}) \mid t \in [0,T], \boldsymbol{\upsilon} \in \Delta_k^t\} = \Delta_{k+1}^T$ yields the result. □

Appendix B: Proof of Lemma 8 and bounding the H_2 norms

Proof Let Σ be a fixed system satisfying Assumption 1 and let $\mathbf{Q} = \mathbf{Q}_\Sigma$. Then $\mathbf{A}_0^T\mathbf{Q} + \mathbf{Q}\mathbf{A}_0 \prec -\lambda\mathbf{Q}$, and hence $\mathbf{A}_0 + \frac{\lambda}{2}\mathbf{I}$ is Hurwitz [69]. Let

$$\mathbf{S} = \mathbf{A}_0^T\mathbf{Q} + \mathbf{Q}\mathbf{A}_0 + \sum_{i=1}^{n_\mathrm{p}} \mathbf{A}_i^T\mathbf{Q}\mathbf{A}_i + \sum_{i=1}^{n_\mathrm{p}} \mathbf{C}_i^T\mathbf{C}_i + \mathbf{C}_0^T\mathbf{C}_0 + \lambda\mathbf{Q}.$$

Then $\mathbf{S} \prec 0$ and hence $\mathbf{S} = -\mathbf{V}\mathbf{V}^T$ for some $\mathbf{V}$. Define

$$\tilde{\mathbf{C}} = \left(\mathbf{C}_0^T | \ldots | \mathbf{C}_{n_\mathrm{p}}^T | \mathbf{V}\right)^T \quad \tilde{\mathbf{A}} = \mathbf{A}_0 + 0.5\lambda\mathbf{I} \quad \mathbf{N}_i = \mathbf{A}_i, i \in [n_\mathrm{p}].$$

Then

$$\tilde{\mathbf{A}}^T\mathbf{Q} + \mathbf{Q}\tilde{\mathbf{A}} + \sum_{i=1}^{n_\mathrm{p}} \mathbf{N}_i^T\mathbf{Q}\mathbf{N}_i + \tilde{\mathbf{C}}^T\tilde{\mathbf{C}} = \mathbf{0}$$

and hence, by [52, Theorem 6], for any choice of the matrix $\mathbf{G} = \left(\mathbf{G}_1 | \ldots | \mathbf{G}_{n_\mathrm{p}}\right)$, the bilinear system

$$S(\mathbf{G}) \begin{cases} \dot{\mathbf{z}(t)} = \tilde{\mathbf{A}}\mathbf{z}(t) + \sum_{i=1}^{n_\mathrm{p}} \left(\mathbf{N}_i\mathbf{z}(t)p_i(t) + \mathbf{G}_i p_i(t)\right) \\ \mathbf{z}(0) = \mathbf{0} \\ \tilde{\mathbf{y}}(t) = \tilde{\mathbf{C}}\mathbf{z}(t) \end{cases}$$

has a finite H_2 norm $\|S(\mathbf{G})\|_{H_2}$ which satisfies $\|S(\mathbf{G})\|_{H_2}^2 = \mathrm{trace}(\mathbf{G}^T\mathbf{Q}\mathbf{G})$, and which is defined via Volterra-kernels as follows.

For each $i \in [n_\mathrm{p}]$, let

$$\mathbf{g}_{i,\emptyset}^{\mathbf{G}}(t) = \tilde{\mathbf{C}}\mathrm{e}^{\tilde{\mathbf{A}}t}\mathbf{G}_i$$

and for every $I = (i_1, \ldots, i_k) \in I_k, k > 0, \mathbf{t} = (t_{k+1}, \ldots, t_1)^T \in \mathbb{R}^{k+1}$ and $t \in \mathbb{R}$ let

$$\mathbf{g}_{i,I}^{\mathbf{G}}(\mathbf{t}) = \tilde{\mathbf{C}}\mathrm{e}^{\tilde{\mathbf{A}}t_{k+1}}\mathbf{N}_{i_k}\mathrm{e}^{\tilde{\mathbf{A}}t_k} \cdots \mathrm{e}^{\tilde{\mathbf{A}}t_2}\mathbf{N}_{i_1}\mathrm{e}^{\tilde{\mathbf{A}}t_1}\mathbf{G}_i.$$

Then

$$\|S(\mathbf{G})\|_{H_2}^2 = \sum_{i=1}^{n_\mathrm{p}} \sum_{k=0}^{\infty} \sum_{I \in I_k} \int_{[0,+\infty)^{k+1}} \left\|\mathbf{g}_{i,I}^{\mathbf{G}}(\mathbf{t})\right\|_2^2 d\mathbf{t}.$$

Let $\mathbf{G}^{i,j}$ be the matrix of which the ith column is the jth column of $\mathbf{B}_i$ and all the other elements are zero, $i \in [n_\mathrm{p}]$, $j \in [n_\mathrm{in}]$. By choosing $\mathbf{G} = \mathbf{G}^{i_r,j}$, it follows that the jth column of $w_{i_q,i_r,I}^{\lambda}(\boldsymbol{\tau})$ is the (i_q+1)-th row of $\mathbf{g}_{i_r,I}^{\mathbf{G}^{i_r,j}}(\mathbf{t})$ where $\boldsymbol{\tau} \in \Delta_{k+1}^{\infty}$ and $\mathbf{t} = (\tau_{k+1} - \tau_k, \ldots, \tau_2 - \tau_1, \tau_1)^T$, $i_q, i_r \in [n_\mathrm{p}]_0$, $I \in I_k, k \geq 0$. Hence

$$\sum_{j=1}^{n_\mathrm{in}} \left\|\mathbf{g}_{i_r,I}^{\mathbf{G}^{i_r,j}}(\mathbf{t})\right\|_2^2 \geq \sum_{i_q=0}^{n_\mathrm{p}} \left\|w_{i_q,i_r,I}^{\lambda}(\boldsymbol{\tau})\right\|_2^2$$

and by applying a change of variables in the iterated integrals,

$$\sum_{i_q,i_r=0}^{n_{\mathrm{p}}}\sum_{k=0}^{\infty}\sum_{I\in I_k}\int_{\Delta_{k+1}^{\infty}}\left\|w_{i_q,i_r,I}^{\lambda}(\boldsymbol{\tau})\right\|_2^2 d\boldsymbol{\tau}\leq\sum_{i_r=0}^{n_{\mathrm{p}}}\sum_{j=1}^{n_{\mathrm{in}}}\left\|S(\mathbf{G}^{i_r,j})\right\|_{H_2}^2=\sum_{i_r=0}^{n_{\mathrm{p}}}\mathrm{trace}(\mathbf{B}_{i_r}^T\mathbf{Q}\mathbf{B}_{i_r}).$$

In the last step we used the fact that

$$\sum_{j=1}^{n_{\mathrm{in}}}\mathrm{trace}((\mathbf{G}^{i_r,j})^T\mathbf{Q}\mathbf{G}^{i_r,j})=\mathrm{trace}(\mathbf{B}_{i_r}^T\mathbf{Q}\mathbf{B}_{i_r}).$$

Finally, from Lemma 13 it follows that $y_{\Sigma}(\mathbf{u},\mathbf{p})(T)=\langle w^{T,\Sigma},\varphi^{T,\mathbf{u},\mathbf{p}}\rangle_{\mathcal{H}}$ for a suitable Hilbert space. From the definition of $w^{T,\Sigma}$ and $\varphi^{T,\mathbf{u},\mathbf{p}}$ and the proof of Lemma 13 it follows that

$$\begin{aligned}\|w^{T,\Sigma}\|_{\mathcal{H}}&\leq\|\Sigma\|_{\lambda,H_2}\\ \|\varphi^{T,\mathbf{u},\mathbf{p}}\|_{\mathcal{H}}&\leq(n_{\mathrm{p}}+1)\|\mathbf{u}\|_{L^2([0,T],\mathbb{R}^{n_{\mathrm{in}}})}\end{aligned}$$

and hence by the Cauchy-Schwarz inequality the result follows. □

Appendix C: Multi-output case

Let use assume that the elementwise loss functions satisfies the following condition:

$$\ell(\mathbf{y},\tilde{\mathbf{y}})=\sum_{j=1}^{n_{\mathrm{out}}}\ell_j(y^j,\tilde{y}^j),$$

where y^j, $\tilde{y}^j$ denote the jth component of the vectors $\mathbf{y},\tilde{\mathbf{y}}\in\mathbb{R}^{n_{\mathrm{out}}}$, respectively, and ℓ_i are Lipschitz functions with Lipschitz constant c_ℓ. For instance, if $\ell(\mathbf{y},\tilde{\mathbf{y}})=\|\mathbf{y}-\mathbf{y}'\|_1$ is the ℓ_1 loss, or $\ell(\mathbf{y},\mathbf{y}')=\|\mathbf{y}-\mathbf{y}'\|_2^2$ is the classical quadratic loss and $\mathbf{y},\mathbf{y}'$ are bounded, then this assumption is satisfied.

For all $\Sigma\in\mathcal{E}$, let Σ_j be the LPV system which arises from Σ by considering only the jth output, and let $\mathcal{E}_j$ be the class of LPV systems formed by all Σ_j. For every $j\in[n_{\mathrm{out}}]$, consider the dataset $\mathbf{S}_j=\{(\mathbf{u}_i,\mathbf{p}_i,y_i^j)\}_{1\leq i\leq N}$, which is obtained from the dataset $\mathbf{S}$ by taking only the jth component of the true outputs $\{\mathbf{y}_i\}_{i=1}^N$.

Then it follows that

$$\mathcal{L}(\Sigma)=\sum_{j=1}^{n_{\mathrm{out}}}\mathcal{L}(\Sigma_j),$$

where

$$\mathcal{L}(\Sigma_j)=\mathbb{E}_{(\mathbf{u},\mathbf{p},y)\sim\mathcal{D}}\left[\int_{[0,T]}\ell_j(y_{\Sigma_j}(\mathbf{u},\mathbf{p})(\tau),y^j(\tau))\,d\mu(\tau)\right]$$

$$\mathcal{L}_N^{\mathbf{S}^j}(\Sigma_j) = \frac{1}{N}\sum_{i=1}^{N}\int_{[0,T]} \ell_j(y_{\Sigma_j}(\mathbf{u},\mathbf{p})(\tau), y^j(\tau))d\mu(\tau)$$

and

$$\mathcal{L}_N^{\mathbf{S}}(\Sigma) = \sum_{j=1}^{n_{\text{out}}} \mathcal{L}_N^{\mathbf{S}_j}(\Sigma_j).$$

By applying Theorem 10 to $\mathbf{S}_j$ and $\mathcal{E}_j$, $j \in [n_{\text{out}}]$, it follows that

$$\mathbb{P}_{\mathbf{S}\sim\mathcal{D}^N}\left(\forall \Sigma_j \in \mathcal{E}_j : \mathcal{L}(\Sigma_j) - \mathcal{L}_N^{\mathbf{S}_j}(\Sigma_j) \leq \frac{R_j(\delta)}{\sqrt{N}}\right) \geq 1-\delta$$

for all $j \in [n_{\text{out}}]$. Then by using the union bound it follows that

$$\mathbb{P}_{\mathbf{S}\sim\mathcal{D}^N}\left(\forall j \in [n_{\text{out}}],\ \forall \Sigma_j \in \mathcal{E}_j : \mathcal{L}(\Sigma_j) - \mathcal{L}_N^{\mathbf{S}_j}(\Sigma_j) \leq \frac{\max\limits_{1\leq j\leq n_{\text{out}}} R_j(\delta)}{\sqrt{N}}\right) \geq 1-n_{\text{out}}\delta$$

and by using $\mathcal{L}(\Sigma) = \sum_{j=1}^{n_{\text{out}}} \mathcal{L}(\Sigma_j)$ and $\mathcal{L}_N^{\mathbf{S}}(\Sigma) = \sum_{j=1}^{n_{\text{out}}} \mathcal{L}_N^{\mathbf{S}_j}(\Sigma_j)$ it then follows that

$$\mathbb{P}_{\mathbf{S}\sim\mathcal{D}^N}\left(\forall \Sigma \in \mathcal{E} : \mathcal{L}(\Sigma) - \mathcal{L}_N^{\mathbf{S}}(\Sigma) \leq \frac{n_{\text{out}}\max_{1\leq j\leq n_{\text{out}}} R_j(\delta)}{\sqrt{N}}\right) \geq 1-n_{\text{out}}\delta.$$

Author contributions D. R., M. P., B. D. contributed to ideas and the proofs of the results and to the writing of the manuscript. M. G., and A. B. contributed to the discussion of the paper and to the choice of the numerical example. D.R. and M. P. prepared the numerical example. All authors reviewed the manuscript.

Funding Open access funding provided by HUN-REN Institute for Computer Science and Control. This research was supported by the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory, by the C.N.R.S. E.A.I. project "Stabilité des algorithmes d'apprentissage pour les réseaux de neurones profonds et récurrents en utilisant la géométrie et la théorie du contrôle via la compréhension du rôle de la surparamétrisation", and by the E.D.F. project 101103386 "FaRADAI".

Data availability The MATLAB code for the numerical example can be found on GitHub, the link is provided in the manuscript.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. Tóth R (2010) Modeling and identification of linear parameter-varying systems, vol 403. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-13812-6
2. Bamieh B, Giarre L (2002) Identification of linear parameter varying models. Int J Robust Nonlinear Control IFAC-Affil J 12(9):841–853. https://doi.org/10.1002/rnc.706
3. Verdult V, Verhaegen M (2002) Subspace identification of multivariable linear parameter-varying systems. Automatica 38(5):805–814. https://doi.org/10.1016/S0005-1098(01)00268-0
4. Piga D, Cox P, Tóth R, Laurain V (2015) LPV system identification under noise corrupted scheduling and output signal observations. Automatica 53:329–338. https://doi.org/10.1016/j.automatica.2015.01.018
5. dos Santos PL, Perdicoúlis TPA, Novara C, Ramos DE, Rivera JA (2011) Linear parameter-varying system identification. World Scientific, Hackensack. https://doi.org/10.1142/8186
6. Oomen T, Bosgra O (2012) System identification for achieving robust performance. Automatica 48(9):1975–1987. https://doi.org/10.1016/j.automatica.2012.06.011
7. dos Santos PL, Ramos JA, de Carvalho JLM (2008) Identification of LPV systems using successive approximations. In: Proceedings of 47th IEEE conference on decision and control, pp 4509–4515. https://doi.org/10.1109/CDC.2008.4738786
8. Cox P, Petreczky M, Tóth R (2018) Towards efficient maximum likelihood estimation of LPV-SS models. Automatica 97(9):392–403. https://doi.org/10.1016/j.automatica.2018.08.021
9. Ljung L (1998) System identification. In: Signal analysis and prediction. Birkhäuser, Boston, pp 163–173. https://doi.org/10.1007/978-1-4612-1768-8_11
10. Shalev-Shwartz S, Ben-David S (2014) Understanding machine learning: from theory to algorithms. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9781107298019
11. Alquier P (2021) User-friendly introduction to PAC-Bayes bounds. Preprint at arXiv:2110.11216. https://doi.org/10.48550/arXiv.2110.11216
12. Orvieto A, Smith SL, Gu A, Fernando A, Gulcehre C, Pascanu R, De S (2023) Resurrecting recurrent neural networks for long sequences. In: International conference on machine learning. PMLR, pp 26670–26698
13. Kidger P (2022) On neural differential equations. Preprint at arXiv:2202.02435. https://doi.org/10.48550/arXiv.2202.02435
14. Gu A, Dao T (2023) Mamba: linear-time sequence modeling with selective state spaces. Preprint at arXiv:2312.00752. https://doi.org/10.48550/arXiv.2312.00752
15. Isidori A (1985) Nonlinear control systems: an introduction. Springer, Heidelberg. https://doi.org/10.1007/BFb0006368
16. Krener A (1974) Bilinear and nonlinear realization of input-output maps. SIAM J Control 13(4):827–834. https://doi.org/10.1137/0313049
17. Vidyasagar M, Karandikar RL (2006) A learning theory approach to system identification and stochastic adaptive control. Probabilistic and randomized methods for design under uncertainty, 265–302. https://doi.org/10.1016/j.jprocont.2007.10.009
18. Zheng Y, Li N (2020) Non-asymptotic identification of linear dynamical systems using multiple trajectories. IEEE Control Syst Lett 5(5):1693–1698. https://doi.org/10.1109/LCSYS.2020.3042924
19. Sun Y, Oymak S, Fazel M (2020) Finite sample system identification: optimal rates and the role of regularization. In: Proceedings of the 2nd conference on learning for dynamics and control. PMLR, vol 120, pp 16–25
20. Tu S, Frostig R, Soltanolkotabi M (2024) Learning from many trajectories. J Mach Learn Res 25(216):1–109
21. Mohri M, Rostamizadeh, A (2008) Rademacher complexity bounds for non-i.i.d. processes. In: Advances in neural information processing systems, vol 21
22. McDonald DJ, Shalizi CR (2017) Rademacher complexity of stationary sequences. Preprint at arXiv:1106.0730. https://doi.org/10.48550/arXiv.1106.0730

23. Hanson J, Raginsky M, Sontag ED (2021) Learning recurrent neural net models of nonlinear systems. In: Learning for dynamics and control. PMLR, pp 425–435
24. Campi MC, Weyer E (2002) Finite sample properties of system identification methods. IEEE Trans Autom Control 47(8):1329–1334. https://doi.org/10.1109/TAC.2002.800750
25. Massucci L, Lauer F, Gilson M (2021) Regularized switched system identification: a statistical learning perspective. IFAC-PapersOnLine 54(5):55–60. https://doi.org/10.1016/j.ifacol.2021.08.474
26. Koiran P, Sontag ED (1998) Vapnik-Chervonenkis dimension of recurrent neural networks. Discret Appl Math 86(1):63–79. https://doi.org/10.1016/S0166-218X(98)00014-6
27. Kuusela P, Ocone D, Sontag ED (2004) Learning complexity dimensions for a continuous-time control system. SIAM J Control Optim 43(3):872–898. https://doi.org/10.1137/S0363012901384302
28. Fermanian A, Marion P, Vert J-P, Biau G (2021) Framing RNN as a kernel method: a neural ode approach. Adv Neural Inf Process Syst 34:3121–3134
29. Marion P (2024) Generalization bounds for neural ordinary differential equations and deep residual networks. Adv Neural Inf Process Syst 36:48918–48938
30. Hanson J, Raginsky M (2024) Rademacher complexity of neural odes via chen-fliess series. In: 6th annual learning for dynamics & control conference. PMLR, pp 758–769
31. Wei C, Ma T (2019) Data-dependent sample complexity of deep neural networks via Lipschitz augmentation. In: Advances in neural information processing systems, vol 32
32. Joukovsky B, Mukherjee T, Van Luong H, Deligiannis N (2021) Generalization error bounds for deep unfolding RNNs. In: Uncertainty in artificial intelligence. PMLR, vol 161, pp 1515–1524
33. Chen M, Li X, Zhao T (2020) On generalization bounds of a family of recurrent neural networks. In: Proceedings of AISTATS 2020. PMLR, vol 108, pp 1233–1243
34. Alquier P, Wintenberger O (2012) Model selection for weakly dependent time series forecasting. Bernoulli 18(3):883–913. https://doi.org/10.3150/11-BEJ359
35. Alquier P, Li X, Wintenberger O (2013) Prediction of time series by statistical learning: general losses and fast rates. Depend Model 1(2013):65–93. https://doi.org/10.2478/demo-2013-0004
36. Eringis D, Leth J, Tan ZH, Wisniewski R, Petreczky M (2023) PAC-Bayesian bounds for learning LTI-ss systems with input from empirical loss. Preprint at arXiv:2303.16816. https://doi.org/10.48550/arXiv.2303.16816
37. Eringis D, leth, Tan Z-H, Wisniewski R, Petreczky M (2024) PAC-Bayesian error bound, via rényi divergence, for a class of linear time-invariant state-space models. In: Forty-first international conference on machine learning. https://openreview.net/forum?id=a1Olc2QhPv
38. Simchowitz M, Boczar R, Recht B (2019) Learning linear dynamical systems with semi-parametric least squares. In: Conference on learning theory. PMLR, pp 2714–2802
39. Oymak S, Ozay N (2022) Revisiting Ho-Kalman-based system identification: robustness and finite-sample analysis. IEEE Trans Autom Control 67(4):1914–1928. https://doi.org/10.1109/TAC.2021.3083651
40. Tsiamis A, Ziemann I, Matni N, Pappas GJ (2023) Statistical learning theory for control: a finite-sample perspective. IEEE Control Syst Mag 43(6):67–97. https://doi.org/10.1109/MCS.2023.3310345
41. Tsiamis A, Pappas GJ (2019) Finite sample analysis of stochastic system identification. In: 2019 IEEE 58th conference on decision and control (CDC), pp 3648–3654. https://doi.org/10.1109/CDC40024.2019.9029499
42. Foster D, Simchowitz M (2020) Logarithmic regret for adversarial online control. In: Proceedings of the 37th ICML. PMLR, vol 119, pp 3211–3221
43. Billingsley P (2017) Probability and measure. Wiley, New York
44. Chen RT, Rubanova Y, Bettencourt J, Duvenaud DK (2018) Neural ordinary differential equations. Adv Neural Inf Process Syst 31:6572–6583
45. Massaroli S, Poli M, Park J, Yamashita A, Asama H (2020) Dissecting neural ODEs. Adv Neural Inf Process Syst 33:3952–3963
46. Eringis D, Leth J, Tan Z-H, Wisniewski R, Petreczky M (2024) Pac-bayes generalisation bounds for dynamical systems including stable RNNs. In: Proceedings of the AAAI conference on artificial intelligence, vol 38, pp 11901–11909
47. Sznaier M, Amishima T, Parrilo PA, Tierno J (2002) A convex approach to robust h2 performance analysis. Automatica 38(6):957–966. https://doi.org/10.1016/S0005-1098(01)00299-0
48. Veenman J, Scherer CW, Köroğlu H (2016) Robust stability and performance analysis based on integral quadratic constraints. Eur J Control 31:1–32. https://doi.org/10.1016/j.ejcon.2016.04.004

49. Rösinger CA, Scherer CW (2019) A scalings approach to h2-gain-scheduling synthesis without elimination. IFAC-PapersOnLine 52(28):50–57. https://doi.org/10.1016/j.ifacol.2019.12.347
50. Paganini F (1999) Convex methods for robust h/sub 2/ analysis of continuous-time systems. IEEE Trans Autom Control 44(2):239–252. https://doi.org/10.1109/9.746251
51. Gosea IV, Petreczky M, Antoulas AC (2021) Reduced-order modeling of LPV systems in the Loewner framework. In: 2021 60th IEEE conference on decision and control (CDC), pp 3299–3305. https://doi.org/10.1109/CDC45484.2021.9683742 . IEEE
52. Zhang L, Lam J (2002) On $\mathcal{H}_2$ model reduction of bilinear systems. Automatica 38:205–216. https://doi.org/10.1016/S0005-1098(01)00204-7
53. Garnier H, Wang L (2008) Identification of continuous-time models from sampled data. Springer, London. https://doi.org/10.1007/978-1-84800-161-9
54. Ober R (1991) Balanced parametrization of classes of linear systems. SIAM J Control Optim 29(6):1251–1287. https://doi.org/10.1137/0329065
55. Hanzon B, Ober RJ (1998) Overlapping block-balanced canonical forms for various classes of linear systems. Linear Algebra Appl 281(1):171–225. https://doi.org/10.1016/S0024-3795(98)10056-3
56. Peeters RLM, Hanzon B, Olivi, M (2004) Canonical lossless state-space systems: Staircase forms and the schur algorithm. IFAC Proceedings Volumes. In: 2nd IFAC Symposium on System Structure and Control, Oaxaca, Mexico, December 8-10, 2004, vol 37, no 21, pp 117–122. https://doi.org/10.1016/S1474-6670(17)30454-8
57. Antoulas AC (2005) Approximation of large-scale dynamical systems. SIAM, Philadelphia. https://doi.org/10.1137/1.9780898718713
58. Ribarits T (2002) The role of parametrizations in identification of linear dynamic systems. PhD thesis, Vienna University of Technology, Vienna
59. Goos J, Pintelon R (2016) Continuous-time identification of periodically parameter-varying state space models. Automatica 71:254–263. https://doi.org/10.1016/j.automatica.2016.04.013
60. Laurain V, Toth R, Gilson M, Garnier H (2011) Direct identification of continuous-time linear parameter-varying input/output models. IET Control Theory Appl 5(7):878–888. https://doi.org/10.1049/iet-cta.2010.0218
61. Mejari M, Mavkov B, Forgione M, Piga D (2021) An integral architecture for identification of continuous-time state-space LPV models. IFAC-PapersOnLine 54(8):7–12. https://doi.org/10.1016/j.ifacol.2021.08.573
62. Boyd S, Chua L (1985) Fading memory and the problem of approximating nonlinear operators with Volterra series. IEEE Trans Circuits Syst 32(11):1150–1161. https://doi.org/10.1109/TCS.1985.1085649
63. Zang G, Iglesias PA (2003) Fading memory and stability. J Franklin Inst 340(6–7):489–502. https://doi.org/10.1016/j.jfranklin.2003.11.002
64. Grigoryeva L, Ortega J-P (2018) Universal discrete-time reservoir computers with stochastic inputs and linear readouts using non-homogeneous state-affine systems. J Mach Learn Res 19(24):1–40
65. Grigoryeva L, Ortega J-P (2018) Echo state networks are universal. Neural Netw 108:495–508. https://doi.org/10.1016/j.neunet.2018.08.025
66. Gu A, Johnson I, Timalsina A, Rudra A, Ré C (2022) How to train your hippo: state space models with generalized orthogonal basis projections. Preprint at arXiv:2206.12037. https://doi.org/10.48550/arXiv.2206.12037
67. Tóth R, Heuberger PS, Hof PM (2010) Discretisation of linear parameter-varying state-space representations. IET Control Theory Appl 4(10):2082–2096. https://doi.org/10.1049/iet-cta.2009.0572
68. Dao T, Gu A (2024) Transformers are SSMs: generalized models and efficient algorithms through structured state space duality. In: International conference on machine learning. PMLR, pp 10041–10071
69. Sontag ED (1998) Mathematical control theory. Springer, New York. https://doi.org/10.1007/978-1-4612-0577-7