2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Kernel-based learning of stable nonlinear state-space models

M.F. Shakib[1,2], R. Tóth[3], A.Y. Pogromsky[2], A. Pavlov[4], and N. van de Wouw[2]

Abstract— **This paper presents a kernel-based learning approach for black-box nonlinear state-space models with a focus on enforcing model stability. Specifically, we aim to enforce a stability notion called convergence which guarantees that, for any bounded input from a user-defined class, the model responses converge to a unique steady-state solution that remains within a positively invariant set that is user-defined and bounded. Such a form of model stability provides robustness of the learned models to new inputs unseen during the training phase. The problem is cast as a convex optimization problem with convex constraints that enforce the targeted convergence property. The benefits of the approach are illustrated by a simulation example.**

I. Introduction

Given the complexity of today's engineering systems, deriving dynamic models from first-principle laws can be a challenging task. Data-driven modeling approaches offer an alternative by using data to construct dynamic models directly. Classically, system identification methods identify black-box linear time-invariant (LTI) models from system data [1]. For most of the widely used LTI identification methods, the stability property of the identified models is not guaranteed [2] and, therefore, dedicated methods have been developed to enforce stability, see, e.g., [3], [4]. To further increase model flexibility, data-driven black-box *nonlinear modeling* has become increasingly popular in recent years [5], [6], [7]. However, nonlinear models can exhibit complex dynamics (e.g., multiple attractors) which makes it more challenging to guarantee stability as, e.g., even a small input perturbation can lead to a wildly different, possibly even unbounded, response. Such learned models do not generalize well to input variations and can be dangerous in safety-critical applications. Consequently, there is an increasing need for reliable learning of black-box nonlinear models with *stability guarantees* [8], [9], [10], [11], [12], [13]. Such stability properties provide robustness of the learned nonlinear model to inputs unseen during training and are instrumental for system analysis and control design [8], [9].

[1]Department of Electrical and Electronic Engineering, Imperial College London, London, United Kingdom (m.shakib@imperial.co.uk)

[2]Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands ({m.f.shakib; a.pogromsky; n.v.d.wouw}@tue.nl)

[3]Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands, and with the Institute for Computer Science and Control, Budapest, Hungary (r.toth@tue.nl)

[4]Department of Geoscience and Petroleum, NTNU, Trondheim, Norway (alexey.pavlov@ntnu.no)

This research was partially supported by the Engineering and Physical Sciences Research Council (grant number: EP/W005557/1) and by the Eötvös Loránd Research Network (grant number: SA-77/2021).

Kernel-based methods [14] are a particular class of learning methods for black-box nonlinear state-space modeling. These methods include regularization networks [15], support vector machines [16], and Gaussian process regression [17], and are, under certain conditions, *universal approximators* [15], [18]. However, the complex function description of kernel-based nonlinear state-space models complicates the analysis of model properties, such as stability. For example, the fixed points of state-space models whose dynamics are described by the widely used squared-exponential kernel can, in general, not be found analytically [19].

Kernel-based learning methods that *enforce* some form of model stability during the learning process are proposed in [20], [21] for the *autonomous* case, i.e., models without external (time-varying) inputs. For models with external inputs, [8] proposes a method that enforces *global contraction* for kernel-based nonlinear *input-output* models. Methods that enforce model stability for other model classes, such as recurrent equilibrium network models [6], polynomial state-space models [22], [23], and Lur'e-type models [9], [12], have also been proposed. However, none of these methods is directly applicable to *kernel-based state-space* models. Furthermore, all of these methods enforce a global form of model stability, i.e., for *any* bounded input and the complete state space. In practice, however, measured system data does not cover the entire input and state space and the implicit assumption that stability properties hold globally is often not true or cannot be verified. Therefore, enforcing a *regional* stability form rather than a *global* one can be beneficial for two reasons. Firstly, it allows for the observed stability property to be preserved exactly. Secondly, it increases the modeling flexibility, since global stability constraints can be more stringent than region-based ones.

This paper presents a learning method for *non-autonomous* kernel-based discrete-time state-space models that enforces the *convergence* stability property, thereby providing robustness to changes in the initial condition and the input. This stability notion guarantees the boundedness, uniqueness, and asymptotic stability of the steady-state solution [24], [25], [26]. Our proposed approach uses a *region-based* version of convergence, namely convergence on *compact invariant sets*, see Figure 1 for a graphical illustration. This is conceptually different from current methods in the literature, e.g., [8], [6], [22], [23], [9], [12], as it gives the user the freedom to enforce the stability property only in a desired region of the input and state space, and is naturally suitable for learning as the system data is observed only in (compact) sets. The compact invariant set can be chosen to be larger than the region in which the system data is collected to enable safe

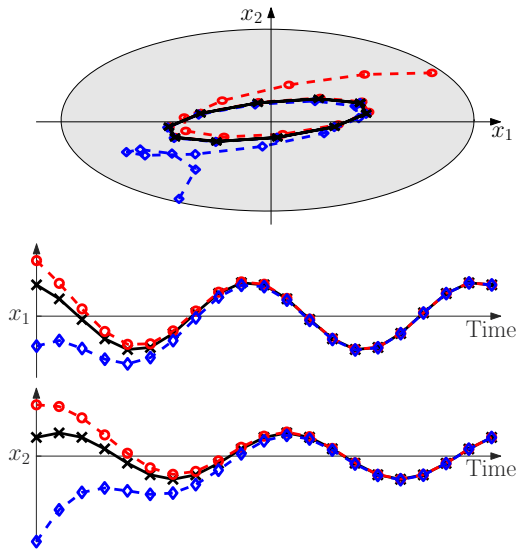

Fig. 1. Illustration of the convergence in compact sets property for a two-dimensional discrete-time state-space model. Top: the phase plane where the positively invariant set is represented by the grey area, the measured data by the black crosses and the time series of model simulations with two different initial conditions by the red circles and blue diamonds. Bottom: time series of the measured data in the black crosses and time series of model simulations with two different initial conditions in the red circles and blue diamonds. It can be seen that the effect of the initial conditions fades out thanks to the convergence property. There are no guarantees for solutions starting outside the positively invariant set.

model extrapolation. The learning problem is to minimize the *regularized equation error criterion* constrained to the set of models that are convergent in a user-defined compact set. Crucially, we show that this optimization problem is jointly convex in the criterion and the constraints. Using a simulation example, we explicitly show that learning without enforcing such model stability can lead to unstable model responses. Such unfavorable scenarios are prevented by the approach proposed in this paper.

The remainder of this paper is organized as follows. Section II formally introduces the learning problem. Section III presents the proposed learning approach. Section IV describes the results of a simulation study. Finally, Section V presents the conclusions of the paper.

Notation: The symbols $\mathbb{R}, \mathbb{R}_+, \mathbb{C}$, and $\mathbb{Z}$ denote the set of real numbers, non-negative real numbers, complex numbers, and integers, respectively. The symbol I_n denotes the $n \times n$ identity matrix and the symbol 0_n denotes the zero vector of dimension $n \times 1$. A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is called positive (negative) definite, denoted as $A \succ 0$ ($A \prec 0$), if all its eigenvalues are strictly positive (negative). For a vector $x \in \mathbb{R}^n$, its Euclidean norm is denoted by $|x|$. Given a matrix $P \succ 0$ and a vector $x \in \mathbb{R}^n$, $|x|_P$ denotes $\sqrt{x^\top P x}$.

II. Problem Statement

A. Data-generating system

Consider the discrete-time data-generating system represented by the following set of nonlinear difference equations:

$$\tilde{x}_{k+1} = f(\tilde{x}_k, u_k), \tag{1a}$$
$$\tilde{y}_k = h(\tilde{x}_k, u_k) + e_k, \tag{1b}$$

where, at time instance $k \in \mathbb{Z}$, the state is denoted by $\tilde{x}_k \in \mathbb{R}^n$, the input by $u_k \in \mathbb{R}^m$, and the output by $\tilde{y}_k \in \mathbb{R}^l$. The noise $e_k \in \mathbb{R}^l$, is assumed to be independent and identically distributed white noise with a zero-mean normal distribution which has a finite diagonal covariance matrix Σ_e. This corresponds to an output error (OE) type of noise structure for (1). The mapping f is called the state-transition map and the mapping h is called the output map. Without loss of generality, we assume that the origin is a fixed point for the zero input, i.e., $0_n = f(0_n, 0_m)$. The class of bounded inputs U_c is defined as follows for any constant $c \in \mathbb{R}_+$:

$$U_c := \{\{u_k\}_{k \in \mathbb{Z}} \mid u_k \in \mathcal{U}_c, \forall\, k \in \mathbb{Z}\} \tag{2}$$

with $\mathcal{U}_c$ being a ball around the origin defined as follows:

$$\mathcal{U}_c := \{u \in \mathbb{R}^m \mid |u| \leq c\}. \tag{3}$$

Solutions of the system (1a) for inputs $u \in U_c$ are all sequences $\{\tilde{x}_k, u_k\}_{k=k_0}^{\infty} \in \mathbb{R}^{n+m}$ that satisfy (1a) with $\tilde{x}_{k_0} \in \mathbb{R}^n$. If no confusion arises, such a solution is denoted by $\tilde{x}$.

It is assumed that the data-generating system (1a) exhibits a strong form of model stability for any input from an a priori known input class U_c. Hereto, the notion of *global* convergence, defined in [27], is adapted to convergence *on compact sets* and is defined as follows.

Definition 1: The discrete-time nonlinear system (1a) is said to be exponentially convergent in a set $\mathcal{X} \subset \mathbb{R}^n$ for a class of inputs U_c if, for every $u \in U_c$,

- there exists a solution $\bar{x}$, called the steady-state solution, that is defined and lies in $\mathcal{X}$ for all $k \in \mathbb{Z}$;
- the steady-state solution $\bar{x}$ is exponentially stable for any initial condition in $\mathcal{X}$, i.e., there exist scalars $\tau \in \mathbb{R}_+$ and $0 \leq \rho < 1$ such that for any $\tilde{x}_0 \in \mathcal{X}$, the solution $\tilde{x}$ satisfies:

$$|\tilde{x}_k - \bar{x}_k|^2 \leq \tau \rho^k |\tilde{x}_0 - \bar{x}_0|^2, \forall\, k \geq 0. \tag{4}$$

Assumption 1: The data-generating system (1a) is exponentially convergent on the user-defined convergence region $\mathcal{X}$ for the user-defined class of inputs U_c.

For the zero input, i.e., $u_k = 0$, for all $k \in \mathbb{Z}$, the origin of an exponentially convergent system (1) is an exponentially stable fixed-point for any initial condition in $\mathcal{X}$. Furthermore, for any non-constant input sequence from U_c, e.g., periodic input, the exponentially stable steady-state solution $\bar{x}$ is in general also non-constant, e.g., periodic. Consequently, the effect of initial conditions fades out, which can be exploited in a learning setting that uses *steady-state* response data to avoid the dependency on initial conditions, as is done in [9].

Remark 2: With some adaptations, the approach proposed in this paper can be extended to include systems with process noise. In that setting, the approach in [28] enables the estimation of the noise realization e and a sample-based estimate of the covariance matrix Σ_e directly from the input-output data. Consequently, the noise sequence can be treated as an additional input during the subsequent estimation of f and h. However, in this paper, we focus on enforcing stability

and due to space limitations this extension is not presented.

B. Model class

For the estimation of f and h, a reproducing kernel Hilbert space (RKHS) based modeling approach is taken, where basis functions are defined by a so-called *kernel* function. Consider the following model class:

$$\begin{aligned} Ex_{k+1} &= \hat{f}(x_k, u_k) \coloneqq \sum_{i=1}^{N_s} \alpha_i K_i^{\alpha}(x_k, u_k), \\ y_k &= \hat{h}(x_k, u_k) \coloneqq \sum_{i=1}^{N_s} \beta_i K_i^{\beta}(x_k, u_k), \end{aligned} \tag{5}$$

where the functions $K_i^{\alpha}(x_k, u_k) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $K_i^{\beta}(x_k, u_k) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ are *kernel slices*, generated by symmetric, positive definite kernel functions. The weights $\alpha_i \in \mathbb{R}^n, \beta_i \in \mathbb{R}^l$, for all $i \in \{1, \dots N_s\}$, with N_s the number of kernel slices used, and the matrix $E \in \mathbb{R}^{n \times n}$, which is restricted to be non-singular, are model parameters. These parameters are collected in a parameter vector $\theta \in \mathbb{R}^{n_\theta}$ with $n_\theta \coloneqq n^2 + (n + l)N_s$. An example of a kernel function is the squared-exponential (SE) kernel function [14]:

$$K^{\alpha}(z^a, z^b) \coloneqq \exp\left(-\frac{|z^a - z^b|^2}{2\ell}\right), \tag{6}$$

where the kernel width $\ell > 0$ is a tuneable hyper-parameter and $z^a, z^b \in \mathbb{R}^p$ and p is a positive integer. For the problem at hand, the kernel slices in (5) are then defined as follows:

$$K_i^{\alpha}(x, u) \coloneqq K^{\alpha}\left(\begin{bmatrix} x_i^s \\ u_i^s \end{bmatrix}, \begin{bmatrix} x \\ u \end{bmatrix}\right), \tag{7}$$

where $x_i^s \in \mathbb{R}^n, u_i^s \in \mathbb{R}^m, i = 1, \dots, N_s$, are so-called *pseudo inputs* [29] or *inducing variables* [30]. The pseudo inputs facilitate efficient sparse implementation [29], [30] and are also part of the tuneable hyper-parameters. Note that the model parametrization θ is *linear*. As we will show later, the proposed approach in Section III guarantees that the matrix E in (5) is non-singular and exploits this matrix for the convexification of the proposed approach.

The choice of the kernel function defines the resulting function space in which $\hat{f}$ and $\hat{h}$ are searched. The approach in this paper can be applied to almost any kernel function, e.g., the linear, polynomial, spline, and wavelet kernels [14], or the SE kernel function as in (6). The only requirement is that the kernel K^{α} is differentiable with respect to x over the convergence region $\mathcal{X}$. Furthermore, the kernel functions that generate the kernel slices in $\hat{f}$ and $\hat{h}$ in (5) can be selected independently. The kernel functions, together with their hyper-parameters and pseudo input locations define a possibly infinite set of basis functions [31].

C. Learning problem

The dataset, denoted by $\mathcal{D}$, contains N samples of the input $\tilde{u}$, the output $\tilde{y}$, and the state $\tilde{x}$ of the data-generating system (1), and is defined as follows:

$$\mathcal{D} = \{u_k, \tilde{y}_k, \tilde{x}_k\}_{k=1}^{N}. \tag{8}$$

It is assumed that the data is generated from zero initial condition. The availability of the state sequence $\tilde{x}$ is exploited in the proposed solution in Section III. In practice, however, full-state measurements are not always available. A compatible state sequence can then be estimated using a kernelized version of canonical correlation analysis (CCA), as outlined in [32], [28]. Because this estimated state sequence comes in an unknown state basis that may be nonlinearly transformed, a nonlinear output mapping $\hat{h}$ should also be learned. It is also worth noting that, under certain conditions, the state estimation approach in [28] is statistically consistent.

The convergence region $\mathcal{X}$ to be enforced during learning, is defined as the convex hyperellipsoidal set:

$$\mathcal{X} \coloneqq \{x \in \mathbb{R}^n \mid x^\top X x \leq 1\}, \tag{9}$$

characterized by the user-defined matrix $0 \prec X \in \mathbb{R}^{n \times n}$. Note that the matrix X can always be chosen such that $\mathcal{X}$ contains the state sequence $\tilde{x}$ in the dataset $\mathcal{D}$ such that $\{\tilde{x}_k\}_{k=1}^{N} \in \mathcal{X}$. However, the set $\mathcal{X}$ can be chosen larger to allow for extrapolation without instability problems.

Consider the regularized equation error criterion:

$$\begin{aligned} J(\theta, \mathcal{D}) \coloneqq & \frac{\gamma_f}{2(N-1)} \sum_{k=1}^{N-1} \left\| E\tilde{x}_{k+1} - \hat{f}(\tilde{x}_k, u_k) \right\|_2^2 \\ & + \frac{\gamma_h}{2N} \sum_{k=1}^{N} \left\| \tilde{y}_k - \hat{h}(\tilde{x}_k, u_k) \right\|_2^2 \\ & + \frac{1}{2} \sum_{i=1}^{n} \left\| \hat{f}_{(i)} \right\|_{\mathcal{H}}^2 + \frac{1}{2} \sum_{i=1}^{p} \left\| \hat{h}_{(i)} \right\|_{\mathcal{H}}^2, \end{aligned} \tag{10}$$

where $\|\cdot\|_{\mathcal{H}}^2 = \langle \cdot, \cdot \rangle$ is the squared Hilbert-space norm, defined for functions in an RKHS, and $\hat{f}_{(i)} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the i-th element of the vector-valued function $\hat{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ (and $\hat{h}_{(i)}$ is defined similarly). The RKHS is defined through the kernel function and its hyper-parameters, see [31] for details. Essentially, (10) is the one-step-ahead prediction error criterion with additional regularization terms (the last two terms in (10)) to avoid overfitting and to control the bias-variance tradeoff via the regularization-parameters $\gamma_f, \gamma_h \in \mathbb{R}_+$. The criterion in (10) is quadratic and, thanks to (5) and the properties of the RKHS norm, it is convex in the model parameter vector θ characterizing $\hat{f}$ and $\hat{h}$. Based on the criterion (10), the convergence region $\mathcal{X}$, and the input class U_c, the learning problem is formalized as follows.

Problem 3: Consider the data-set $\mathcal{D}$ in (8), the convergence region $\mathcal{X}$ in (9), and the class of inputs U_c in (2). Find a model of the form (5) such that the regularized equation error criterion J in (10) is minimized and such that the learned model (5) is exponentially convergent on the convergence region $\mathcal{X}$ for the class of inputs U_c.

In practice, checking whether the true system is convergent is a non-trivial task. Therefore, Problem 3 can be interpreted in two different ways. Firstly, if the data-generating system is convergent, e.g., inferred by analyzing its responses, then Problem 3 preserves this property for the learned model. Secondly, if it is unknown whether the data-generating sys-

tem is convergent, then Problem 3 enforces the convergence property because it is a favorable model property for using the model, e.g., for reliable model simulation for new inputs.

III. Convergent models by convex learning

In this section, the learning problem is formulated as a constrained optimization problem for a given, fixed set of kernel hyper-parameters, regularization parameters, and pseudo-input locations. The constrained optimization problem is formulated jointly convex in the error criterion and the constraints that enforce the convergence property.

Using the convex convergence region $\mathcal{X}$ defined in (9), a test for the convergence property on $\mathcal{X}$ for models of the form (5) is presented in the next theorem using the notation:

$$A(x,u) \coloneqq \sum_{i=1}^{N_s} \alpha_i \frac{\partial K_i^\alpha}{\partial x}(x,u). \tag{11}$$

Theorem 4: Consider the model (5), the convergence region $\mathcal{X}$ in (9), and the class of inputs U_c in (2) defined through $\mathcal{U}_c$ in (3). Assume that $A(x,u)$ exists for all $(x,u) \in (\mathcal{X},\mathcal{U}_c)$. If there exists a matrix $P \in \mathbb{R}^{n\times n}$ such that:

$$P \succ 0, \tag{12a}$$

$$\begin{bmatrix} E+E^\top-P & A^\top(x,u) \\ A(x,u) & P \end{bmatrix} \succ 0, \tag{12b}$$

$$\begin{bmatrix} E+E^\top-X & \hat{f}(x,u) \\ \hat{f}^\top(x,u) & 1 \end{bmatrix} \succeq 0, \tag{12c}$$

for all $(x,u) \in (\mathcal{X}\times\mathcal{U}_c)$. Then, the matrix E is non-singular and the model (5) is exponentially convergent in the set $\mathcal{X}$ under any input from U_c. As a consequence, any two solutions starting from $x_0^a, x_0^b \in \mathcal{X}$, with the same input $u \in U_c$ will remain in $\mathcal{X}$, i.e., $x_k^a, x_k^b \in \mathcal{X}, \forall\, k \geq 0$. Furthermore, there exist scalars $\tau \in \mathbb{R}_+$ and $0 \leq \rho < 1$, such that any two solutions starting from $x_0^a, x_0^b \in \mathcal{X}$ with the same input $u \in U_c$ remain in $\mathcal{X}$ for $k \geq 0$ and converge exponentially to each other, i.e.,

$$|x_k^a - x_k^b|^2 \leq \tau\rho^k |x_0^a - x_0^b|^2, \forall\, k \geq 0. \tag{13}$$

Proof: The proof is omitted for the sake of brevity. ■

The condition (12b) enforces (exponential) *incremental stability* on $(\mathcal{X},\mathcal{U}_c)$, while the condition (12c) enforces *positive invariance* of the set $\mathcal{X}$ for inputs from U_c. The latter property of the set $\mathcal{X}$ for the model (5) implies that there exists a solution $\bar{x}$ that lies in $\mathcal{X}$ for all $k \in Z$, see [33, Lemma 2]. As a consequence, (exponential) *convergence* according to Definition 1 is guaranteed. The conditions of Theorem 4 are convex in the matrix P and the parameter vector θ as these appear linearly in the conditions (12).

For most kernel choices, the function $\hat{f}$ in (5) is non-convex in x and u. Consequently, the conditions of Theorem 4 must be verified for all $(x,u) \in (\mathcal{X},\mathcal{U}_c)$. For some specific parametrization of the state-transition map $\hat{f}$ in (5), the conditions of Theorem 4 can be verified efficiently, for example using a polynomial basis function expansion for $\hat{f}$ and the sum-of-squares programming techniques, see [22], [23]. Unfortunately, that approach does not apply to kernel-based modeling using a generic class of kernels, as is the case considered in this paper. To make the conditions computationally tractable, the verification is performed on a grid where the sets $\mathcal{X}^{\mathrm{g}}, \mathcal{U}_c^{\mathrm{g}}$ denote the gridded version of the convergence region $\mathcal{X}$ and the input space $\mathcal{U}_c$, respectively. This is further motivated by the observation that the matrices in the conditions of Theorem 4 depend continuously on (x,u) and the observation that many kernels, including the SE kernel in (6), are smooth, i.e., infinitely many times differentiable. The grid density trades off the risk of violating the constraints (12) for some $(x,u) \in (\mathcal{X},\mathcal{U}_c)$ against numerical complexity.

In addition to the parameter vector θ, another parameter vector is introduced that contains all the remaining parameters. This vector is denoted by $\phi \in \mathbb{R}^{n_\phi}$ with $n_\phi \coloneqq n_{\mathrm{hyp}} + 2 + (n+m)N_s$, and contains (i) the n_{hyp} number of kernel hyper-parameters; (ii) the regularization parameters γ_f, γ_h in (10); and (iii) the N_s number of pseudo input locations $(x_i^s, u_i^s), i = 1,\ldots,N_s$.

Given the conditions of Theorem 4 and the gridded sets $\mathcal{X}^{\mathrm{g}}, \mathcal{U}_c^{\mathrm{g}}$, the model set $\Theta(\phi)$ is defined as follows:

$$\begin{aligned} \Theta(\phi) \coloneqq \{\theta \in \mathbb{R}^{n_\theta} \mid &(12) \text{ is feasible} \\ &\forall\,(x,u) \in (\mathcal{X}^{\mathrm{g}},\mathcal{U}_c^{\mathrm{g}})\}\,. \end{aligned} \tag{14}$$

The set $\Theta(\phi)$ encodes the convergence property such that any candidate model $\theta \in \Theta(\phi)$ satisfies the conditions of Theorem 4 on the grid $(\mathcal{X}^g,\mathcal{U}_c^g)$. It is assumed that this grid-based test, for a sufficiently dense grid, gives exponential convergence on the non-grid-based convergence region $\mathcal{X}$ with the non-grid-based input space $\mathcal{U}_c$. The dependence of the set Θ on ϕ is via the constraints (12b) and (12c), both of which are inherently dependent on ϕ.

With some abuse of notation, the equation error criterion (10) is written as $J(\theta,\phi)$ with arguments θ and ϕ. Using this notation, the following constrained optimization problem is formulated for any given fixed choice of parameters ϕ:

$$\hat{\theta}_\phi = \underset{\theta\in\Theta(\phi)}{\arg\min}\, J(\theta,\phi), \tag{15}$$

which is jointly convex in the parameter vector θ in the criterion as well as in its constraints for any fixed ϕ.

Remark 5: For any $\phi \in \mathbb{R}^{n_\phi}$, and any selected kernel function, the set $\Theta(\phi)$ in (14) is non-empty because model (5) with $E = X \succ 0, \alpha_i = 0_n, \beta_i = 0_l, i = 1,\ldots,N_s$, satisfies the conditions of Theorem 4 for $P = X$. In fact, such a model is *globally* convergent.

Remark 6: A choice of the vector ϕ can be interpreted as a choice of the *model class*. For example, changing the kernel hyper-parameters or the pseudo input locations will result in different functions generated by the kernels. In the RKHS literature, a variety of hyper-parameter tuning approaches have been developed, i.e., for the tuning of ϕ, see, e.g., [17], [29], [30]. The vector θ can be interpreted as the parametrization of the *specific model*. The joint learning of ϕ and θ could be cast as a two-level optimization problem similar to [34]. However, in this paper, we focus on the learning of θ only with guaranteed model stability.

Remark 7: The computational complexity of the learning problem depends on the dimensions of the LMIs in Theorem 4 and the number of to-be-learned parameters n_θ. Consider the case where g_x points are used for each of the n state components and g_u points are used for each of the m input components. The dimension of the LMIs in Theorem 4 is $g_x^n g_u^m(3n+1) \times g_x^n g_u^m(3n+1)$ (note that the LMI (12a) is implied by the satisfaction of (12b)). The number of parameters to learn is $n_\theta = n^2 + (n+l)N_s$, where N_s is the number of pseudo input locations and l is the number of outputs. In numerical case studies, the learning problem for a second-order model ($n = 2$), for $g_x = 16, g_u = 20$ grid points (320 points in total), and $N_s = 20$ pseudo input locations can be successfully solved within minutes on a modern laptop. A topic for future study is to investigate the implementation limits in terms of n, g_x, g_u, and N_s. We are working on extensions that do not require gridding and are therefore more computationally efficient.

IV. Numerical case study

This section presents a *simulation case study* that highlights the benefits of models learned using the strategy proposed in this paper. The main contribution of this paper is the learning of the state-transition mapping with the convergence property enforced. Therefore, in this example, we focus only on the learning of the state-transition mapping. The predictive quality of the learned mapping is evaluated using the best-fit rate (BFR) of the simulated model response. The BFR is defined as follows:

$$\text{BFR} := 100\% \cdot \max\left(0, 1 - \frac{\|\tilde{x} - x\|_2}{\|\tilde{x} - \text{mean}(\tilde{x})\|_2}\right), \quad (16)$$

where $\tilde{x}$ is the (measured) response of the data-generating system and x is the model response computed by the forward simulation. The BFR can be calculated for both the training and test datasets.

Consider the following data-generating nonlinear system:

$$x_{k+1} = \begin{cases} u_k - 2x_\star^3 + 3x_\star^2 x_k & \text{if } x_k > x_\star, \\ u_k + x_k^3 & \text{if } |x_k| \le x_\star, \\ u_k + 2x_\star^3 + 3x_\star^2 x_k & \text{if } x_k < -x_\star, \end{cases} \quad (17)$$

$$y_k = x_k, \quad (18)$$

where $x_\star = 0.5$. The dataset $\mathcal{D}$ of length $N = 100$ is generated using an input sequence drawn from a normal distribution and scaled such that $|u_k| \le 1$, for all $k \in \{1, \ldots, N\}$, thus $u \in U_c$ with $c = 1$. Using this dataset, two models are trained using the third-order polynomial kernel and a number of $N_s = 10$ pseudo inputs. Firstly, in a traditional manner, a model of the form (5) is learned in an unconstrained manner according to [17] (containing both θ and ϕ and using $E = \mathcal{I}_n$), hence not guaranteed to be convergent. This model is called the *non-convergent* model with estimated parameters ψ_{nc}. Secondly, using the proposed approach in this paper, a model of the form (5) is learned for the set $\mathcal{X}$ in (9) with $X = 1$, i.e., $\mathcal{X} := \{x \in \mathbb{R}^n | |x| \le 1\}$, and for the class of inputs U_c as in (2) with $c = 1$. This model is called the *convergent* model with estimated parameters ψ_c. To implement the LMI constraints characterizing the set $\Theta(\phi)$ in (14), the input and state-space are gridded equidistantly to give 11 and 10 points in $\mathcal{X}$ and $\mathcal{U}_c$, respectively. The responses of the learned models to the input in the estimation dataset are depicted in Figure 2. Although both models perform well, the non-convergent model performs slightly better on the estimation data, which is also evident by its BFR of 95.4% versus 91.0% for the convergent model. This accuracy result is expected as the proposed approach sacrifices accuracy for guaranteed model stability.

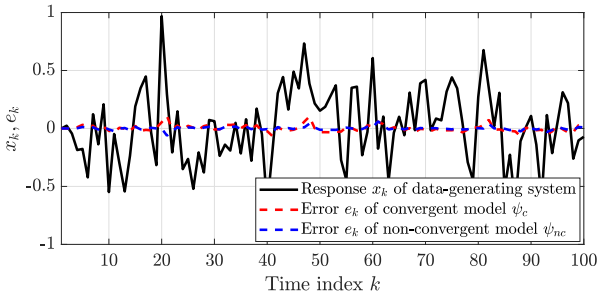

Fig. 2. Response of the data-generating system and the simulation error of the learned convergent and non-convergent models for the input in the estimation dataset.

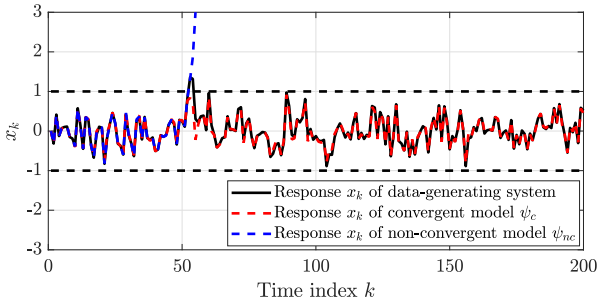

Fig. 3. Bounded response of the data-generating system for the test input. The response of the learned convergent model remains within $|x| \le 1$, while the response of the learned non-convergent model becomes unbounded.

Both models are subjected to a test input drawn from the same distribution as the estimation input, i.e., from the same class of inputs U_c with constant $c = 1$. Figure 3 shows that the response of the non-convergent model to this new input becomes unbounded, even though the response of the data-generating system remains bounded. However, the response of the convergent model remains within $\mathcal{X}$, i.e., within $|x| \le 1$, as enforced. The BFR for the convergent model is 78.7% on the test data.

For scalar models, the conditions of Theorem 4 can be guaranteed only if the Jacobian of the state-transition mapping $E^{-1}\hat{f}$ with respect to x remains in absolute value below 1 for all $(x, u) \in (\mathcal{X}, \mathcal{U}_c)$. Figure 4 shows that this necessary condition is violated for the non-convergent model for zero input, while it is satisfied by the convergent model.

This example illustrates the potential instability hazards of learned models with traditional approaches, even in a noiseless scenario. Furthermore, it shows that the proposed approach guarantees bounded model responses inside the

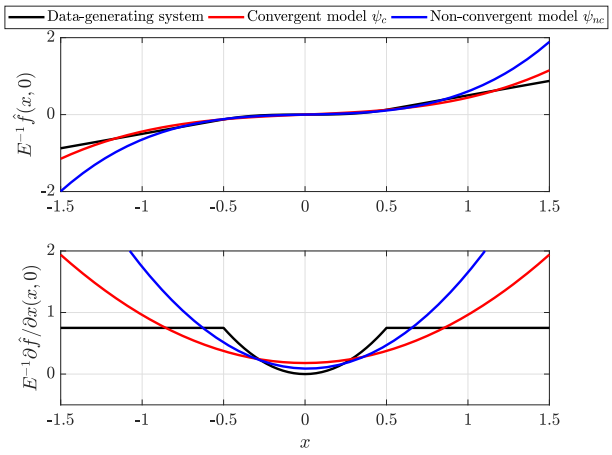

Fig. 4. The state-transition mapping (top panel) of the data-generating system, the learned convergent model, and the learned non-convergent model for zero input, together with their Jacobian (bottom panel) for zero input.

convergence region $\mathcal{X}$ for any input from the input class U_c. The enforced convergence property thus provides robustness of the learned model to new inputs.

V. Conclusions

This paper presents an approach for learning kernel-based stable nonlinear state-space models. The solutions of the resulting models are guaranteed to remain inside a user-defined positively invariant set for a class of user-defined inputs. Consequently, models learned by the proposed approach safely generalize to unseen scenarios, which ensures the robustness of the learned models to new inputs. The benefits of the approach are illustrated by a simulation example, in which the model learned using the proposed approach generalizes favorably to new inputs, while models learned using traditional methods produce unstable, unbounded responses for unseen scenarios.

References

[1] L. Ljung, *System identification: theory for the user*. Prentice-hall, 1987.

[2] L. Ljung, "Perspectives on system identification," *Annual Reviews in Control*, vol. 34, no. 1, pp. 1–12, 2010.

[3] J. Umenberger, J. Wågberg, I. R. Manchester, and T. B. Schön, "Maximum likelihood identification of stable linear dynamical systems," *Automatica*, vol. 96, pp. 280–292, 2018.

[4] L. Ljung and T. Söderström, *Theory and practice of recursive identification*. MIT press, 1983.

[5] K. Berntorp, "Online Bayesian inference and learning of Gaussian-process state–space models," *Automatica*, vol. 129, p. 109613, 2021.

[6] M. Revay, R. Wang, and I. R. Manchester, "Recurrent equilibrium networks: Unconstrained learning of stable and robust dynamical models," in *Proc. of the Conference on Decision and Control*, pp. 2282–2287, 2021.

[7] G. Beintema, R. Tóth, and M. Schoukens, "Nonlinear state-space identification using deep encoder networks," in *Proc. of Machine Learning Research (3rd Annual Learning for Dynamics & Control Conference)*, (Zurich, Switzerland), pp. 1–10, 2021.

[8] H. J. van Waarde and R. Sepulchre, "Kernel-based models for system analysis," *IEEE Transactions on Automatic Control*, 2022.

[9] M. F. Shakib, A. Y. Pogromsky, A. Pavlov, and N. van de Wouw, "Computationally efficient identification of continuous-time Lur'e-type systems with stability guarantees," *Automatica*, vol. 136, p. 110012, 2022.

[10] N. Takeishi and Y. Kawahara, "Learning dynamics models with stable invariant sets," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 9782–9790, 2021.

[11] S. Jin, Z. Wang, Y. Ou, and W. Feng, "Learning accurate and stable dynamical system under manifold immersion and submersion," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3598–3610, 2019.

[12] M. Shakib, A. Pogromsky, A. Pavlov, and N. van de Wouw, "Fast identification of continuous-time Lur'e-type systems with stability certification," *IFAC-PapersOnLine*, vol. 52, no. 16, pp. 227–232, 2019.

[13] J. Duan, Y. Ou, J. Hu, Z. Wang, S. Jin, and C. Xu, "Fast and stable learning of dynamical systems based on extreme learning machine," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 6, pp. 1175–1185, 2017.

[14] B. Schölkopf, A. J. Smola, F. Bach, *et al.*, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[15] G. De Nicolao and G. F. Trecate, "Consistent identification of NARX models via regularization networks," *IEEE Transactions on Automatic Control*, vol. 44, no. 11, pp. 2045–2049, 1999.

[16] J. A. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. P. Vandewalle, *Least squares support vector machines*. World scientific, 2002.

[17] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, USA: The MIT Press, 2005.

[18] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657–682, 2014.

[19] T. Beckers and S. Hirche, "Stability of Gaussian process state space models," in *Proc. of the European Control Conference*, pp. 2275–2281, 2016.

[20] M. Khosravi and R. S. Smith, "Nonlinear system identification with prior knowledge on the region of attraction," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 1091–1096, 2020.

[21] S. M. Khansari-Zadeh and A. Billard, "Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions," *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 752–765, 2014.

[22] M. H. Abbasi, L. Iapichino, W. Schilders, and N. van de Wouw, "A data-based stability-preserving model order reduction method for hyperbolic partial differential equations," *Nonlinear Dynamics*, vol. 107, no. 4, pp. 3729–3748, 2022.

[23] J. Umenberger and I. R. Manchester, "Specialized interior point algorithm for stable nonlinear system identification," *IEEE Transactions on Automatic Control*, 2018.

[24] B. P. Demidovich, "Lectures on Stability Theory," tech. rep., Nauka, Moscow, 1967.

[25] A. Pavlov, N. van de Wouw, and H. Nijmeijer, *Uniform output regulation of nonlinear systems: a convergent dynamics approach*. Springer Science & Business Media, Birkhäuser Boston, 2006.

[26] A. Pavlov, A. Pogromsky, N. van de Wouw, and H. Nijmeijer, "Convergent dynamics, a tribute to Boris Pavlovich Demidovich," *Systems & Control Letters*, vol. 52, no. 3, pp. 257 – 261, 2004.

[27] A. Pavlov and N. van de Wouw, "Steady-state analysis and regulation of discrete-time nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1793–1798, 2012.

[28] M. Shakib, R. Tóth, A. Pogromsky, A. Pavlov, and N. van de Wouw, "State-space kernelized closed-loop identification of nonlinear systems," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1126–1131, 2020.

[29] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," *Advances in neural information processing systems*, vol. 18, 2005.

[30] M. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Proc. of the international conference on Artificial intelligence and statistics*, pp. 567–574, 2009.

[31] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American mathematical society*, vol. 68, no. 3, pp. 337–404, 1950.

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

[33] A. Pavlov and N. van de Wouw, "Convergent discrete-time nonlinear systems: The case of PWA systems," in *Proc. of the American Control Conference*, pp. 3452–3457, 2008.

[34] D. Khandelwal, *Automating data-driven modelling of dynamical systems: an evolutionary computation approach*. Springer Nature, Cham, Switzerland, 2022.