Learning Stable Gaussian Process State Space Models

Jonas Umlauft[1], Armin Lederer, Sandra Hirche[1]

Abstract— **Data-driven nonparametric models gain importance as control systems are increasingly applied in domains where classical system identification is difficult, e.g., because of the system's complexity, sparse training data or its probabilistic nature. Gaussian process state space models (GP-SSM) are a data-driven approach which requires only high-level prior knowledge like smoothness characteristics. Prior known properties like stability are also often available but rarely exploited during modeling. The enforcement of stability using control Lyapunov functions allows to incorporate this prior knowledge, but requires a data-driven Lyapunov function search. Therefore, we propose the use of Sum of Squares to enforce convergence of GP-SSMs and compare the performance to other approaches on a real-world handwriting motion dataset.**

I. Introduction

Accurate identification of dynamical systems is fundamental for the application of control engineering. Classical control uses system identification methods and observations of the system's behavior to identify parametric models. The identification strongly depends on the choice of the model structure as otherwise the corresponding set of parameters cannot be determined. The potential model candidates are unknown, especially for complex, possibly stochastic, systems, e.g., human behavior. Therefore, identification using data-driven models has recently emerged, because the application of control engineering reaches more towards areas where no analytic description of the system dynamic exists.

Practitioners, e.g., in robotics, successfully employ autonomous dynamical systems for encoding desired trajectories based on observed motions from humans. Given training data points of point-to-point motions, a stable dynamical system is learned, which guarantees convergence to the goal point during reproduction [1]. The application of nonparametric data-driven models is very promising in such domains as knowledge about the complexity of the motion is missing a priori. Nevertheless, some fundamental properties of the dynamical system must be given as otherwise generalization outside of the observed dataset is not possible [2]. Here, we consider sufficiently *smooth behavior* and the *convergence* of the dynamical system as prior knowledge, which both hold for many real-world systems.

From a system identification point of view, subspace methods as proposed in [3] allow to incorporate stability assumptions. But classical methods rely on a particular parametric model structure, e.g., ARMA models in the linear case or Hammerstein models for nonlinear inputs [4]. However, for a good performance, the model structure should be known. Another issue is the quantification of the possible ignorance of the model outside of the training dataset. Some identification methods use stochastic embedding [5] to model the bias error as realization of a stochastic process, but the benefits from a Bayesian modeling approach are not exploited. Therefore, the behavior in areas without observations is not properly defined and must be handled with care. Gaussian processes (GPs) allow to quantify the model fidelity as they also encode the uncertainty directly in the stochastic process (without separate model error estimation). The work in [6] considers kernel-based approaches like the GP for identification and reviews recent work at the intersection to machine learning.

Previous work in [7], [8] considers GPs for system identification due to their beneficial properties regarding the bias variance trade-off, the strong background of Bayesian nonparametrics models and the implied smoothness (when using an appropriate kernel). The stability of GP-SSMs is first investigated in [9], [10] but without elaborating how convergence can be enforced. Alternatively, [1] uses Gaussian Mixture Models for learning stable dynamics, while [11] develops a generic framework for stabilizing learned dynamical systems using control Lyapunov functions. However, both methods do not consider the uncertainty in the model.

The contribution of this paper is a novel approach to learning stochastically stable GP-SSMs. First, we propose the use of Sum of Squares (SOS) as a flexible control Lyapunov function and demonstrate its superiority over previous approaches in terms of computational efficiency and flexibility. Second, we present methods to enforce asymptotic stability for deterministic GP-SSM and global uniform boundedness for the stochastic GP-SSM under the SOS Lyapunov function. We demonstrate the results on an experimental handwriting motion dataset.

The remainder of this paper is structured as follows: After formulating the considered problem in Section II, Section III presents the main result. It is followed by numerical evaluation in Section IV and a conclusion.

II. Problem Formulation

Consider a discrete-time system with a continuous-valued state[1] $\boldsymbol{x}_k \in \mathcal{X} \subseteq \mathbb{R}^n$ driven by an unknown stochastic process

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{G}(\boldsymbol{x}_k)\boldsymbol{\omega}_k, \quad \boldsymbol{\omega}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n), \quad (1)$$

[1] Member of the Chair of Information-oriented Control, Department of Electrical and Computer Engineering, Technical University of Munich, D-80290 Munich, Germany [jonas.umlauft,armin.lederer,hirche]@tum.de

[1]**Notation:** Lower/upper case bold symbols denote vectors/matrices, respectively. $\boldsymbol{I}_n$ denotes the $n \times n$ identity matrix and $\boldsymbol{A} \succ 0$ the positive definiteness of matrix $\boldsymbol{A}$. $\mathbb{E}[\cdot]$ denotes the expected value, var$[\cdot]$ the variance of a random variable. $\mathbb{R}_+^0$, $\mathbb{R}_+$ denote all real positive numbers with and without the zero, respectively. The norm $\|\cdot\|$ denotes Euclidean norm.

where $\boldsymbol{\omega}_k$ is a random variable from the probability space $(\Omega, \boldsymbol{F}, \mathcal{P})$ with sample space Ω, the corresponding σ-algebra $\boldsymbol{F}$ and the probability measure $\mathcal{P}$. Assume that $\boldsymbol{f} : \mathcal{X} \to \mathcal{X}$ is infinitely differentiable as well as $\boldsymbol{G} : \mathcal{X} \to \mathbb{R}^n_+$ which maps to the diagonal elements of positive definite matrices. It is assumed, that true dynamical system $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k)$ is globally uniformly asymptotically stable and the real stochastic process (1) is globally uniformly bounded, which is defined as follows:

Definition 1 (Uniform Boundedness [12]): The trivial solution of (1) is uniformly bounded if, for every $\alpha > 0$ there exists $\beta = \beta(\alpha) > 0$ such that

$$\|\boldsymbol{x}_k\| < \beta, \quad \text{almust surely}, \tag{2}$$

when $\|\boldsymbol{x}_0\| < \alpha$ for a $k \geq 0$. It is globally uniformly bounded if α is arbitrary large.

We assume that consecutive measurements of the state are taken, thus N data pairs are given in the training set $\mathcal{D} = \{(\bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_{i+1})\}_{i=1}^N$. The goal is to find a process model $\boldsymbol{f}_{\boldsymbol{\psi}}$ with parameters $\boldsymbol{\psi}$, which estimates the real process (1) based on data $\mathcal{D}$ using the above stated prior assumption on convergence and smoothness.

Based on this definition, finding the model $\boldsymbol{f}_{\boldsymbol{\psi}}$ is formalized as a constrained likelihood optimization

$$\boldsymbol{\psi}^* = \arg\max_{\boldsymbol{\psi}} \sum_{i=1}^{N} \log \mathcal{P}(\bar{\boldsymbol{x}}_{i+1} | \bar{\boldsymbol{x}}_i, \boldsymbol{f}_{\boldsymbol{\psi}}), \tag{3a}$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = \boldsymbol{f}_{\boldsymbol{\psi}}(\boldsymbol{x}_k) \text{ is globally uniformly bounded}, \tag{3b}$$

where $\boldsymbol{f}_{\boldsymbol{\psi}}$ is an infinitely mean-square differentiable stochastic process. The optimization ensures, that the model $\boldsymbol{f}_{\boldsymbol{\psi}}$ best possibly corresponds to the real process (1), when using the likelihood (3a) as a measure for similarity. The constraint (3b) incorporates the prior assumption on boundedness of the underlying process. This paper aims to develop a method for synthesizing the model $\boldsymbol{f}_{\boldsymbol{\psi}}$ using a Bayesian data-driven approach which is computationally tractable.

III. Stochastic Stability with SOS Control Lyapunov Functions

Our general idea is to learn a GP-SSM $\boldsymbol{f}_{\boldsymbol{\psi}_{\mathcal{GP}}}(\boldsymbol{x}_k)$ and a SOS control Lyapunov function $V(\boldsymbol{x}_k)$ separately using the available training data. Then, the overall model is synthesized

$$\boldsymbol{f}_{\boldsymbol{\psi}}(\boldsymbol{x}_k) = \boldsymbol{f}_{\boldsymbol{\psi}_{\mathcal{GP}}}(\boldsymbol{x}_k) + \boldsymbol{u}(\boldsymbol{x}_k), \tag{4}$$

where $\boldsymbol{u}(\boldsymbol{x}_k)$ is an internal stabilizing command computed from the control Lyapunov function.

A. Learning Gaussian Process State Space Models

A Gaussian process is a stochastic process which assigns to any finite subset $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\} \subset \mathbb{R}^n$ in a continuous input domain a joint Gaussian distribution. It is often considered as a distribution over functions [13], denoted by $f(\boldsymbol{x}) \sim \mathcal{GP}(m_{\mathcal{GP}}(\boldsymbol{x}), k_{\mathcal{GP}}(\boldsymbol{x})$. It is fully specified by its mean function $m_{\mathcal{GP}}(\boldsymbol{x})\colon \mathcal{X} \to \mathbb{R}$ and covariance function $k_{\mathcal{GP}}(\boldsymbol{x}, \boldsymbol{x}')\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. If no prior knowledge about the function is available the mean function is commonly set to zero, but our approach equivalently works with nonzero priors. A widely used covariance function is the squared exponential kernel with automatic relevance determination

$$k_{\mathcal{GP}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')^\intercal \boldsymbol{L}^{-2}(\boldsymbol{x} - \boldsymbol{x}')\right). \tag{5}$$

The variables $\boldsymbol{L} = \operatorname{diag}(l_1, \ldots, l_n)$ and $l_j \in \mathbb{R}_+$, $\forall j = 1, \ldots, n$ and $\sigma_f \in \mathbb{R}^0_+$ are considered hyperparameters of the GP and concatenated in $\boldsymbol{\psi}_{\mathcal{GP}} = [l_1 \ \cdots \ l_n \ \sigma_f]^T$. The kernel (5) is infinitely differentiable and consequently results in a distribution over infinitely mean square differentiable functions [13]. Thus, by the choice of the kernel, the smoothness assumption regarding the true process (1) is matched.

We model the output variables as independent from each other, thus, n independent GPs are employed for a n-dimensional state space, denoted as

$$\boldsymbol{f}_{\boldsymbol{\psi}_{\mathcal{GP}}}(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{k}_{\mathcal{GP}}(\boldsymbol{x}, \boldsymbol{x}')), \tag{6}$$

where $\boldsymbol{k}_{\mathcal{GP}}(\cdot,\cdot) = \begin{bmatrix} k_{\mathcal{GP},1}(\cdot,\cdot) & \ldots & k_{\mathcal{GP},n}(\cdot,\cdot) \end{bmatrix}^\intercal$ is composed of covariance functions with corresponding hyperparameters $\boldsymbol{\psi}_{\mathcal{GP},j}$, $j = 1, \ldots, n$, and

$$\boldsymbol{f}_{\boldsymbol{\psi}_{\mathcal{GP}}}(\boldsymbol{x}) = \begin{cases} f_{\boldsymbol{\psi}_{\mathcal{GP}},1}(\boldsymbol{x}) \sim \mathcal{GP}(0, k_{\mathcal{GP},1}(\boldsymbol{x}, \boldsymbol{x}')) \\ \vdots \qquad\qquad \vdots \\ f_{\boldsymbol{\psi}_{\mathcal{GP}},n}(\boldsymbol{x}) \sim \mathcal{GP}(0, k_{\mathcal{GP},n}(\boldsymbol{x}, \boldsymbol{x}')). \end{cases} \tag{7}$$

The GP-SSM models the difference equation of a stochastic dynamical system mapping from the current state $\boldsymbol{x}_k$ to the normal distribution over the next state $\boldsymbol{x}_{k+1}$, thus

$$\boldsymbol{x}_{k+1} \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{k}_{\mathcal{GP}}(\boldsymbol{x}_k, \boldsymbol{x}'_k)). \tag{8}$$

Given input training data $\{\bar{\boldsymbol{x}}_i\}_{i=1}^N$ and output training data $\{\bar{\boldsymbol{x}}_{i+1}\}_{i=1}^N$ and the current state $\boldsymbol{x}_k$, the mean and variance of the j-th component of the next state $x_{k+1,j}$ is given by

$$\mathbb{E}[x_{k+1,j}] = \boldsymbol{k}_j^\intercal (\boldsymbol{K}_j + \sigma_{\text{on}}^2 \boldsymbol{I}_N)^{-1} \boldsymbol{y}_j =: \mu_j(\boldsymbol{x}_k), \tag{9}$$

$$\operatorname{var}[x_{k+1,j}] = k_{\mathcal{GP},j}(\boldsymbol{x}_k, \boldsymbol{x}_k) - \boldsymbol{k}_j^\intercal (\boldsymbol{K}_j + \sigma_{\text{on}}^2 \boldsymbol{I}_N)^{-1} \boldsymbol{k}_j =: \sigma_j^2(\boldsymbol{x}_k), \tag{10}$$

where $\boldsymbol{y}_j = [\bar{x}_{2,j}, \cdots, \bar{x}_{N+1,j}]^\intercal$, $j = 1, \ldots, n$ concatenates the j-th dimension of the output training data $\{\bar{\boldsymbol{x}}_{i+1}\}_{i=1}^N$, $\sigma_{\text{on}} \in \mathbb{R}^0_+$ is observation noise and

$$\boldsymbol{K}_j = \begin{bmatrix} k_{\mathcal{GP},j}(\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_1) & \ldots & k_{\mathcal{GP},j}(\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{x}}_N) \\ \vdots & \ddots & \vdots \\ k_{\mathcal{GP},j}(\bar{\boldsymbol{x}}_N, \bar{\boldsymbol{x}}_1) & \ldots & k_{\mathcal{GP},j}(\bar{\boldsymbol{x}}_N, \bar{\boldsymbol{x}}_N) \end{bmatrix},$$

$$\boldsymbol{k}_j = \begin{bmatrix} k_{\mathcal{GP},j}(\bar{\boldsymbol{x}}_1, \boldsymbol{x}_k) & \ldots & k_{\mathcal{GP},j}(\bar{\boldsymbol{x}}_N, \boldsymbol{x}_k) \end{bmatrix}^\intercal.$$

Following Bayesian inference principle, the optimal hyperparameters regarding the training data are obtained by maximizing the marginal likelihood for every component, i.e. for every $j = 1, \ldots, n$

$$\boldsymbol{\psi}^*_{\mathcal{GP},j} = \arg\max_{\boldsymbol{\psi}_{\mathcal{GP},j}} \log p(\boldsymbol{y}_j | \boldsymbol{X}, \boldsymbol{\psi}_{\mathcal{GP},j}),$$

$$\log p(\boldsymbol{y}_j | \boldsymbol{X}, \boldsymbol{\psi}_{\mathcal{GP},j}) = -\frac{1}{2} \boldsymbol{y}_j^T \boldsymbol{K}_j \boldsymbol{y}_j - \frac{1}{2} \log\det \boldsymbol{K}_j - \frac{N}{2} \log(2\pi), \tag{11}$$

where $\boldsymbol{X} = [\bar{\boldsymbol{x}}_1, \cdots, \bar{\boldsymbol{x}}_N]$ concatenates the input data $\{\bar{\boldsymbol{x}}_i\}_{i=1}^N$. This optimization is nonconvex and commonly solved using gradient-based solvers in the GP-SSM training [13].

B. Learning Sum of Squares Lyapunov function

Sum of Squares, a suitable technique for constructing Lyapunov functions, is defined as follows [14].

Definition 2 (Sum of Squares): For $\boldsymbol{x} \in \mathbb{R}^n$, a multivariate polynomial $p(\boldsymbol{x})$ is a Sum of Squares if there exist some polynomials $r_m(\boldsymbol{x})$, $m = 1 \dots M$ such that

$$p(\boldsymbol{x}) = \sum_{m=1}^{M} r_m^2(\boldsymbol{x}). \tag{12}$$

It is well-known that an equivalent characterization of SOS is given as follows [15].

Lemma 1 (SOS in vector notation): A polynomial $p(\boldsymbol{x})$ of degree $2M$ is a SOS if and only if there exists a positive semidefinite matrix $\hat{\boldsymbol{Q}} \succeq 0$ and a vector of monomials[2] $\boldsymbol{m}(\boldsymbol{x})$ containing monomials of degree $\leq M$ and > 0 such that

$$p(\boldsymbol{x}) = \boldsymbol{m}(\boldsymbol{x})^\intercal \hat{\boldsymbol{Q}} \boldsymbol{m}(\boldsymbol{x}). \tag{13}$$

This lemma crucially simplifies the construction of a SOS to finding the elements of the $M_n \times M_n$ dimensional symmetric matrix $\hat{\boldsymbol{Q}}$. In order to make the SOS a valid Lyapunov candidate, we restrict the matrix to be positive definite $\boldsymbol{Q} \in \mathbb{S}_+$ with

$$\mathbb{S}_+ = \left\{ \boldsymbol{Q} \in \mathbb{R}^{M_n \times M_n} \,|\boldsymbol{Q} \succ 0, \boldsymbol{Q} = \boldsymbol{Q}^\intercal \right\}.$$

1) Learning SOS from Data: Since the deterministic part of the true process $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k)$ is asymptotically stable, there exists a Lyapunov function $V(\boldsymbol{x}_k)$ which reduces in each time step

$$V(\boldsymbol{x}_{k+1}) - V(\boldsymbol{x}_k) < 0. \tag{14}$$

To keep the problem of finding this function tractable, we assume a SOS Lyapunov candidate V of degree $2M$

$$V(\boldsymbol{x}_k) = \boldsymbol{m}(\boldsymbol{x}_k)^\intercal \boldsymbol{Q} \boldsymbol{m}(\boldsymbol{x}_k). \tag{15}$$

Since all which is known about (1), besides the stability, are its realizations in the dataset, those will be used for the Lyapunov function search

$$\boldsymbol{Q}^* = \arg\min_{\boldsymbol{Q} \in \mathbb{S}_+} \sum_{i=1}^{N} g\left(V(\bar{\boldsymbol{x}}_{i+1}) - V(\bar{\boldsymbol{x}}_i)\right), \tag{16}$$

which minimizes the violation of the condition (14) with

$$g(\xi) = \begin{cases} 0 & \text{for } \xi \leq 0 \\ \tilde{g}(\xi) & \text{for } \xi > 0 \end{cases} \tag{17}$$

and $\tilde{g}(\xi) \colon \mathbb{R}_+ \to \mathbb{R}_+$ is chosen as a strictly monotone increasing convex function. With this choice, the optimization can be solved efficiently also for large degrees of the SOS since it is a convex problem.

[2]A monomial of degree M in $\boldsymbol{x} \in \mathbb{R}^n$ is a scalar function $\prod_{j=1}^n x_j^{\alpha_j}$ where α_j are nonnegative integers with $\sum_{j=1}^n \alpha_j = M$. The number of all possible monomials of degree $\leq M$ and > 0 is $M_n = \frac{(n+M)!}{n!M!} - 1$.

Proposition 1: The optimization in (16) is a convex problem if V is a Sum of Squares and the function g is a non-decreasing convex function.

Proof: The Lyapunov function (15) is a linear function of the matrix $\boldsymbol{Q}$. The difference $V(\bar{\boldsymbol{x}}_{i+1}) - V(\bar{\boldsymbol{x}}_i)$ is also linear, thus convex in $\boldsymbol{Q}$. The composition of g and this difference is convex because g is non-decreasing and convex by assumption. As the sum of convex functions is also convex, the objective function is convex. The constraint set $\mathbb{S}_+$ is convex, thus (16) is a convex problem. ■

Remark 1: Even though, we aim to identify a stochastic process, a deterministic Lyapunov criteria on (14) is employed. This is done only in the function search but not in the actual control computation (as shown in the following section) and therefore does not affect the convergence properties of the resulting system. This is also why the parameter choice of the Lyapunov function will only influence the identification precision but not the convergence itself. Considering the observations of the stochastic process as its mean realizations, as in (16), is a reasonable assumption, since the data is the only information available about the system. Because of the stochastic nature, not all data points might adhere to the Lyapunov function which belongs to the stable true dynamics $\boldsymbol{f}$. Therefore, the objective function (16) might not reach zero. However, this not decisive for the convergence properties of the identified system as shown later on.

C. Stabilizing Control

After learning a SOS control Lyapunov function from data, it is employed to compute a stabilizing command $\|\boldsymbol{u}\| < \infty$ for the GP-SSM. We start with a deterministic consideration of the GP-SSM before dealing with the stochastic case.

a) Deterministic Results: For this case, the proposed next state $\tilde{\boldsymbol{x}}_{k+1}$ is the mean prediction $\boldsymbol{\mu} \colon \mathcal{X} \to \mathcal{X}$ of the GP in (9), thus $\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{\mu}(\boldsymbol{x}_k)$, which is possibly not stable. The stabilizing command is found through optimization[3]

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \frac{1}{2} \boldsymbol{u}^\intercal \boldsymbol{u}, \tag{18a}$$

$$\begin{aligned} \text{s.t. } V(\boldsymbol{\mu}(\boldsymbol{x}_k) + \boldsymbol{u}) - V(\boldsymbol{x}_k) < 0 \quad &\text{if } \boldsymbol{x}_k \neq \boldsymbol{0} \\ V(\boldsymbol{\mu}(\boldsymbol{x}_k) + \boldsymbol{u}) - V(\boldsymbol{x}_k) = 0 \quad &\text{if } \boldsymbol{x}_k = \boldsymbol{0}, \end{aligned} \tag{18b}$$

where V is the SOS function determined in (16). This is not a convex problem because of the nonconvex constraint set (18b). Therefore, finding a global minimum cannot be guaranteed. However, this is not critical concerning the stability, because any $\boldsymbol{u}$ in the constraint set leads to a stable system. Reaching a local minimum will only result in behavior different from the training data, but convergence is guaranteed. In case that the stability condition is already satisfied for the uncorrected GP-SSM, the stabilizing command equals $\boldsymbol{0}$, so there is no need to solve this optimization. For the deterministic case, the following conclusion is drawn.

Proposition 2: The system

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_{\boldsymbol{\psi}}(\boldsymbol{x}_k) := \boldsymbol{\mu}(\boldsymbol{x}_k) + \boldsymbol{u}^*(\boldsymbol{x}_k), \tag{19}$$

[3]The dependency of $\boldsymbol{u}$ on $\boldsymbol{x}_k$ was dropped for notational convenience.

where $\boldsymbol{\mu}$ is the mean function of a GP from (9) and the stabilizing command $\boldsymbol{u}^*(\boldsymbol{x}_k)$ obtained from the constrained optimization (18) is uniformly globally asymptotically stable.

Proof: The function V is a SOS and therefore positive definite (with the restriction $\boldsymbol{Q} \in \mathbb{S}_+$) and radially unbounded and therefore a valid Lyapunov candidate. The optimization is feasible, since the solution $\boldsymbol{u} = -\boldsymbol{\mu}(\boldsymbol{x}_k)$ is always in the constraint set because $V(0) - V(\boldsymbol{x}_k) = -V(\boldsymbol{x}_k)$ is negative definite $\forall \boldsymbol{x}_k \in \mathcal{X}$. Also boundedness of $\|\boldsymbol{u}\|$ is full filled here, since deterministic GP-SSMs with squared exponential kernel are bounded [9]. ∎

b) Stochastic Results: For the stochastic case, the proposed state $\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{f}_{\boldsymbol{\psi}_{\mathcal{GP}}}(\boldsymbol{x}_k)$ is a Gaussian distributed vector (from the GP) and the stabilizing command is obtained from the optimization

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \frac{1}{2}\boldsymbol{u}^\intercal \boldsymbol{u}, \tag{20a}$$

$$\begin{aligned} \text{s.t. } \delta V(\boldsymbol{x}_k) < 0 \quad \text{if} \quad \boldsymbol{x}_k \neq \boldsymbol{0}, \\ \delta V(\boldsymbol{x}_k) = 0 \quad \text{if} \quad \boldsymbol{x}_k = \boldsymbol{0}, \end{aligned} \tag{20b}$$

where $\delta V(\boldsymbol{x}_k) = \mathbb{E}[V(\tilde{\boldsymbol{x}}_{k+1} + \boldsymbol{u})|\boldsymbol{x}_k] - V(\boldsymbol{x}_k)$. Since $V(\tilde{\boldsymbol{x}}_{k+1}+\boldsymbol{u})$ is a polynomial in $\tilde{\boldsymbol{x}}_{k+1}$ for the SOS Lyapunov function, the computation of the constraint (20b) reduces to a sum over expectations of multivariate polynomials

$$\begin{aligned} &\mathbb{E}[V(\tilde{\boldsymbol{x}}_{k+1} + \boldsymbol{u})|\boldsymbol{x}_k] = \\ &= \sum_{i,j=1}^{M_n} Q_{ij}\mathbb{E}[m_i(\tilde{\boldsymbol{x}}_{k+1} + \boldsymbol{u})m_j(\tilde{\boldsymbol{x}}_{k+1} + \boldsymbol{u})|\boldsymbol{x}_k], \end{aligned} \tag{21}$$

where $m_i(\cdot), m_j(\cdot)$ are the i,j-th element of the monomial vector $\boldsymbol{m}(\cdot)$ and Q_{ij} the elements of $\boldsymbol{Q}$. These expectations are moments of the random vector $\tilde{\boldsymbol{x}}_{k+1} + \boldsymbol{u}$ and their computation for arbitrary degrees M and dimensions n for normally distributed random vectors are given in [16].

In contrast to the deterministic case, (20) is not feasible $\forall \boldsymbol{x}_k \in \mathcal{X}$, because $\mathbb{E}[V(\tilde{\boldsymbol{x}}_{k+1} + \boldsymbol{u})|\boldsymbol{x}_k] > 0 \ \forall \boldsymbol{x}_k$, even for $\boldsymbol{u} = -\boldsymbol{\mu}(\boldsymbol{x}_k)$. And since $V(\boldsymbol{x}_k) \to 0$ for $\|\boldsymbol{x}_k\| \to 0$, the allowed set $\delta V(\boldsymbol{x}_k) \leq 0$ is empty for some $\boldsymbol{x}_k$ near the origin. Before analyzing the impact of this feasibility problem on convergence properties, we review the Lyapunov based criteria [12] for showing boundedness from Definition 1.

Theorem 1 (Global Uniform Boundedness): The stochastic system of the form (1) is almost surely globally uniformly bounded, if there exists a positive definite, radially unbounded function $V(\boldsymbol{x}_k)$, for which

$$\mathbb{E}[V(\boldsymbol{x}_{k+1})|\boldsymbol{x}_k] - V(\boldsymbol{x}_k) < 0, \quad \forall \|\boldsymbol{x}_k\| > \delta \tag{22}$$

holds for some $\delta \in \mathbb{R}_+$.

To apply this theorem the boundedness of the expectation of the Lyapunov function must be shown for our setting.

Lemma 2: There exists a $c_1 \in \mathbb{R}_+$ which upper bounds the expectation of the Sum of Squares Lyapunov function

$$\mathbb{E}\left[V\left(\tilde{\boldsymbol{x}}_{k+1} + \boldsymbol{u}\right)|\, \boldsymbol{x}_k\right] < c_1, \tag{23}$$

where $\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{f}_{\boldsymbol{\psi}_{\mathcal{GP}}}(\boldsymbol{x}_k)$ is a GP-SSM with a squared exponential kernel and $\boldsymbol{u}$ is obtained from (20).

Proof: The expectations $\mathbb{E}[\tilde{\boldsymbol{x}}_{k+1}]$ and variances $\text{var}[\tilde{\boldsymbol{x}}_{k+1}]$, $j = 1, \ldots, n$, of a GP-SSM with squared exponential kernel are bounded by $\sqrt{N}\sigma_{f,j}^2\|\boldsymbol{h}_j\|$ and $\sigma_{f,j}^2$, respectively. This which was shown in [10] using the abbreviation $\boldsymbol{h}_j = (\boldsymbol{K}_j + \sigma_{\text{on}}^2\boldsymbol{I})^{-1}\boldsymbol{y}_j$. Thus it follows

$$\mathbb{E}\left[\tilde{x}_{k+1,j} + u_j|\, \boldsymbol{x}_k\right] < \sqrt{N}\sigma_{f,j}^2\|\boldsymbol{h}_j\| + u_j \tag{24}$$

$$\text{var}\left[\tilde{x}_{k+1,j} + u_j|\, \boldsymbol{x}_k\right] < \sigma_{f,j}^2. \tag{25}$$

The expectation of the SOS Lyapunov function $\mathbb{E}\left[V(\tilde{\boldsymbol{x}}_{k+1} + \boldsymbol{u})|\, \boldsymbol{x}_k\right]$ is a polynomial in the expectations (24) and the variances (25) as given from equation (21). Hence, it is also bounded. ∎

This leads to the main result on boundedness of the system.

Proposition 3: The system

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_{\boldsymbol{\psi}}(\boldsymbol{x}_k) := \boldsymbol{f}_{\boldsymbol{\psi}_{\mathcal{GP}}}(\boldsymbol{x}_k) + \boldsymbol{u}^*(\boldsymbol{x}_k), \tag{26}$$

with Gaussian process $\boldsymbol{f}_{\boldsymbol{\psi}_{\mathcal{GP}}}(\boldsymbol{x}_k) \sim \mathcal{GP}(\boldsymbol{0}, \boldsymbol{k}_{\mathcal{GP}}(\boldsymbol{x}_k, \boldsymbol{x}_k'))$ with squared exponential kernel as defined in (5) and the stabilizing command $\boldsymbol{u}^*(\boldsymbol{x}_k)$ obtained from the constrained optimization (20) is globally uniformly bounded for a bounded initial value $\boldsymbol{x}_0$. Whenever $\delta V(\boldsymbol{x}_k) > 0$, thus (20) not feasible, $\boldsymbol{u}^* = -\boldsymbol{\mu}(\boldsymbol{x}_k)$ is set.

Proof: From Lemma 2, it can be obtained that

$$\delta V(\boldsymbol{x}_k) < c_1 - V(\boldsymbol{x}_k). \tag{27}$$

Due to the fact that $\lim_{\|\boldsymbol{x}_k\| \to \infty} V(\boldsymbol{x}_k) = \infty$ is unbounded and (27), there must exist a $c_2 \in \mathbb{R}_+$ such that

$$\delta V(\boldsymbol{x}_k) < 0, \quad \forall \|\boldsymbol{x}_k\| > c_2. \tag{28}$$

Therefore, the conditions of Theorem 1 are satisfied. ∎

Thus, there exists a set $\mathcal{B}_{c_2}$ around the origin where (20) is not feasible. But outside of this set, $\mathcal{X} \setminus \mathcal{B}_{c_2}$, the problem is feasible and the stabilizing command ensures decreasing expectation of the Lyapunov function. Therefore, the system is guaranteed to convergence to $\mathcal{B}_{c_2}$ in probability.

IV. Experimental Evaluation

For evaluation purposes, we compare the proposed SOS Lyapunov function, now denoted by $V_{\boldsymbol{Q}}$, the previously proposed weighted sum of asymmetric quadratic functions (WSAQF) V_{WSAQF} [11] and a quadratic Lyapunov function $V_{\boldsymbol{P}}$ based on the LASA handwriting dataset[4]. Since no other approach for learning *stochastic* stable systems is known from literature, we first compare the proposed SOS-based technique in a deterministic setting before showing simulations for the stochastic case.

A. Setup Deterministic

The dataset contains 24 demonstrations of human writing motions in two dimensions. Three repetitions of 150 or 250 steps are taken for each movement, thus $\mathcal{D}$ contains 450-750 elements. In the first step, the hyperparameters $\boldsymbol{\psi}_{\mathcal{GP}}$ for the 2D GP-SSM are trained according to Section III-A. For the deterministic model the mean of the GP-SSM is taken. We compare three different Lyapunov functions.

[4]Available for download at *https://bitbucket.org/khansari/seds*

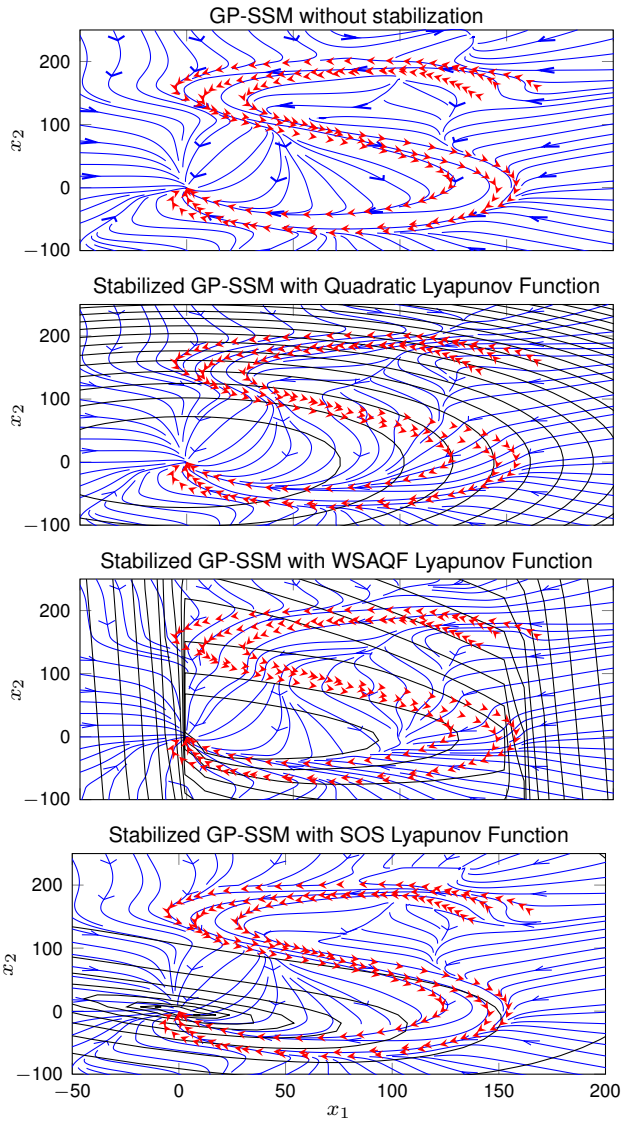

Fig. 1. Result for GP-SSM mean (streamlines in blue) along with the training data (red arrows) for the S-shape motion. First plot depicts the original GP-SSM and the asymptotically stabilized GP-SSMs are shown below. The contour lines of the stabilizing Lyapunov function (Quadratic, WSAQF, SOS) is drawn in black.

- A quadratic Lyapunov function $V_{\boldsymbol{P}}(\boldsymbol{x}_k) = \boldsymbol{x}_k^\intercal \boldsymbol{P} \boldsymbol{x}_k$, where $\boldsymbol{P} \succ 0$ with $n(n+1)/2 = 3$ free parameters (the elements of the Cholesky decomposition of $\boldsymbol{P}$).
- The WSAQF Lyapunov function [11] defined as

$$V_{\text{WSAQF}}(\boldsymbol{x}_k) = \boldsymbol{x}_k^\intercal \boldsymbol{P}_0 \boldsymbol{x} + \sum_{l=1}^{L} \beta_l(\boldsymbol{x}_k) \left(\boldsymbol{x}_k^\intercal \boldsymbol{P}_l (\boldsymbol{x}_k \text{-} \boldsymbol{\xi}_l)\right)^2,$$

$$\text{with} \quad \beta_l(\boldsymbol{x}_k) = \begin{cases} 1 & \text{if } \boldsymbol{x}_k^\intercal \boldsymbol{P}_l (\boldsymbol{x}_k - \boldsymbol{\xi}_l) \geq 0 \\ 0 & \text{otherwise,} \end{cases}$$

which has continuous first order partial derivatives and is positive definite for $\boldsymbol{P}_{0\dots L} \succ 0$. We consider here $L = 3$ resulting in $(L+1)n(n+1)/2 + nL = 18$ free parameters for the Cholesky decompositions of $\boldsymbol{P}_{0\dots L}$ and $\boldsymbol{\xi}_{1\dots L}$. WSAQF have been originally proposed in continuous-time domain, therefore results might vary.

- The proposed SOS Lyapunov function defined in (21) with degree $2M = 4$, resulting in $M_n = 5$ monomials and therefore $M_n(M_n+1)/2 = 15$ free parameters (Cholesky decomposition of $\boldsymbol{Q}$).

The optimization (16) is solved for each Lyapunov function to determine the optimal parameters. For efficient implementation, we perform a Cholesky decomposition $\boldsymbol{L}\boldsymbol{L}^\intercal$ for all positive definite matrices and optimize over the nonzero entries of the lower triangular matrix, which allows to avoid the positive definiteness constraint. To enforce strict positive definiteness of the matrices $\boldsymbol{Q}, \boldsymbol{P}, \boldsymbol{P}_{0\dots L}$ their eigenvalues are enforced to be larger than 0.01 by adding this as a constraint to the numerical optimization. For the optimization (16), we chose $\tilde{g}(\xi) = \xi$ in accordance to Proposition 1. The optimization (16) has, depending on the employed Lyapunov function, a nonconvex or convex objective function (Table I), nevertheless we employed for all the same solver.

For the evaluation of $\boldsymbol{f}_{\boldsymbol{\psi}}$, the correction term $\boldsymbol{u}$ is obtained from solving (18). The constraint can be either convex or nonconvex, making the entire problem convex nor not, as summarized in Table I. For numerical robustness, we enforce the strict negativity in (18b) by

$$V(\boldsymbol{\mu}(\boldsymbol{x}_k) + \boldsymbol{u}) - V(\boldsymbol{x}_k) < -\rho \log(1 + V(\boldsymbol{x}_k))$$

in our implementation, where $\rho = 0.1$ was chosen.

The simulations start from the initial points of the training trajectories until reaching a neighborhood of the origin $\|\boldsymbol{x}_k\| < 10$ or the 500 steps limit. The performance is compared regarding three quality measures:

- The computation time for finding the optimal Lyapunov function (solving (16)) in Matlab on a laptop with Intel Core i7 q740 processor at 1.73 GHz and 4 GB RAM.
- The total area between the training data curve and the curve from the simulated trajectories.
- The average correction effort E_{cor} defined as $E_{\text{cor}} = \sum_{k=1}^{K_f} \|\boldsymbol{u}_k\| \Big/ \sum_{k=1}^{K_f} \|\boldsymbol{x}_k\|$ where K_f is the total number of steps summed over all trajectories and $\boldsymbol{u}_k$ are the corresponding stabilizing commands.

The results are shown graphically for the S-shape motion in Fig. 1 and quantitatively for the entire dataset in Table I. The computation time for the quadratic Lyapunov function is the lowest as it has the least parameters to optimize, but also the lowest precision due to little flexibility. The SOS Lyapunov function outperforms the WSAQF with respect to all measures. The correction effort and the area is the lowest and therefore the stabilized model follows the training data more accurately then all others, while applying fewer corrections.

B. Setup Stochastic

To evaluate the stochastic setting, we simulate as follows.

- GP-SSM without stabilization: Realizations of the trajectories are generated by drawing in each step from

$$\boldsymbol{x}_{k+1} \sim \mathcal{N}\left(\boldsymbol{\mu}(\boldsymbol{x}_k), \boldsymbol{\Sigma}(\boldsymbol{x}_k)\right), \tag{29}$$

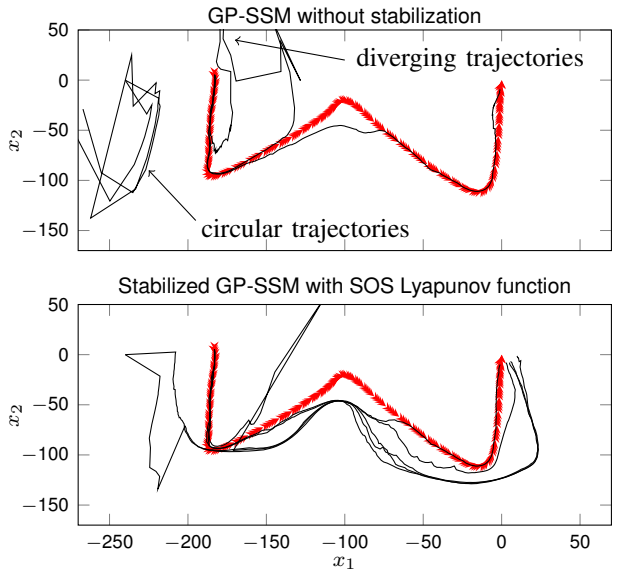

Fig. 2. Stochastic sample paths (black lines) along with the training data (red arrows) for the W-shape for the original and stabilized GP-SSM.

where mean $\mathbb{E}[\boldsymbol{x}_{k+1}] = \boldsymbol{\mu}(\boldsymbol{x}_k)$ and variance $\text{var}[\boldsymbol{x}_{k+1}] = \boldsymbol{\Sigma}(\boldsymbol{x}_k) = \text{diag}\left(\sigma_1^2(\boldsymbol{x}_k), \ldots, \sigma_n^2(\boldsymbol{x}_k)\right)$ are inferred from the GPs according to (9), (10).

- GP-SSM with stabilization through SOS Lyapunov function: At each step $\boldsymbol{u}(\boldsymbol{x}_k)$ is computed according to (20) and the next step is drawn from

$$\boldsymbol{x}_{k+1} \sim \mathcal{N}\left(\boldsymbol{\mu}(\boldsymbol{x}_k) + \boldsymbol{u}(\boldsymbol{x}_k), \boldsymbol{\Sigma}(\boldsymbol{x}_k)\right). \tag{30}$$

The stabilizing command is independent of the realization of $\boldsymbol{x}_{k+1}$ which is unknown when $\boldsymbol{u}$ is computed. It only depends on $\boldsymbol{\mu}(\boldsymbol{x}_k), \boldsymbol{\Sigma}(\boldsymbol{x}_k)$.

For both cases, three trajectories beginning at each starting point of the training data and two trajectories starting outside the training data are generated as shown in Fig. 2. For the original GP-SSM only 1 of 5 path approaches the origin, while all realizations of the stabilized GP-SSM converge to a small set around the origin.

C. Discussion

Regarding flexibility, SOS outperforms the quadratic and WSAQF Lyapunov function on the employed dataset as shown in Table I which leads to higher precision in adherence to training data. Regarding the computational complexity, the search of the quadratic and the SOS Lyapunov function (16) are convex problems, and therefore have significant advantages, over the WSAQF which explains the lower computation times. Computing the stabilizing command for the SOS has a disadvantage compared to WSAQF and quadratic Lyapunov funtions as (18) is not convex. However, this drawback is inherent to flexible Lyapunov candidates: If only convex Lyapunov candidates are permitted, more complex systems cannot be captured. In the stochastic case, sample paths for the original GP-SSM do not converge reliably. In contrast, the proposed stabilized GP-SSM guarantees convergence near the origin.

$V(\boldsymbol{x}_k)$	E_{cor}	Area Error	(16) av. time	(16) convex	(18) convex
Quadratic	0.1523	7.41e+03	**1.06**s	yes	yes
WSAQF	0.0912	4.77e+03	462.64s	no	yes
SOS	**0.0149**	**2.03e+03**	29.89s	yes	no

TABLE I

AVERAGE PERFORMANCE AND PROPERTIES OF QUADRATIC, WSAQF AND SOS LYAPUNOV FUNCTIONS ON THE HANDWRITING DATASET.

V. CONCLUSION

In this paper, we propose a novel approach for learning stable Gaussian process state space models using control Lyapunov functions. In the first step, we propose a Sum of Squares function in the data-driven search for Lyapunov candidates and demonstrate its advantages over existing approaches in terms of computational complexity (convexity of the optimization problem) and its flexibility based on a real world dataset for the deterministic GP-SSM. In the second step, we derive criteria to guarantee global uniform boundedness for the stochastic system and provide simulations.

ACKNOWLEDGMENTS

The European Research Council Starting Grant "Control based on Human Models (conhumo)" supported this work under grant agreement number 337654. Published under ©2018 IEEE with DOI 10.23919/ACC.2017.7963165 [17]

REFERENCES

[1] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with Gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27(5), pp. 943–957, 2011.

[2] D. H. Wolpert, "The supervised learning no-free-lunch theorems," in *Soft Computing and Industry*. Springer, 2002, pp. 25–42.

[3] J. M. Maciejowski, "Guaranteed stability with subspace methods," *Systems & Control Letters*, vol. 26, no. 2, pp. 153–156, 1995.

[4] L. Ljung, *System Identification*. NJ, USA: Prentice Hall PTR, 1998.

[5] G. C. Goodwin, M. Gevers, and B. Ninness, "Quantifying the error in estimated transfer functions with application to model order selection," *IEEE Transactions on Automatic Control*, vol. 37(7), 913-928, 1992.

[6] 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.

[7] J. Kocijan, A. Girard, B. Banko, and R. Murray-Smith, "Dynamic systems identification with Gaussian processes," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 11(4), 411-424, 2005.

[8] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models," in *Advances in Neural Information Processing Systems (NIPS)*, 2005, pp. 1441–1448.

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

[10] ——, "Equilibrium distributions and stability analysis of Gaussian process state space models," in *Conference on Decision and Control (CDC)*, 2016, pp. 6355–6361.

[11] 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*, 62(6), 752-765, 2014.

[12] Y. Li, W. Zhang, and X. Liu, "Stability of nonlinear stochastic discrete-time systems," *Journal of Applied Mathematics*, vol. 2013, 2013.

[13] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, Jan. 2006.

[14] A. Papachristodoulou and S. Prajna, "On the construction of Lyapunov functions using the sum of squares decomposition," in *Conference on Decision and Control (CDC)*, vol. 3, Dec 2002, pp. 3482–3487 vol.3.

[15] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, 2000.

[16] K. Triantafyllopoulos, "Moments and cumulants of the multivariate real and complex Gaussian distributions," University Bristol, 2002.

[17] J. Umlauft, A. Lederer, and S. Hirche, "Learning stable Gaussian process state space models," in *American Control Conference (ACC)*. IEEE, May 2017, pp. 1499–1504.