Learning Stable Stochastic Nonlinear Dynamical Systems

Jonas Umlauft [1] **Sandra Hirche** [1]

Abstract

A data-driven identification of dynamical systems requiring only minimal prior knowledge is promising whenever no analytically derived model structure is available, e.g., from first principles in physics. However, meta-knowledge on the system's behavior is often given and should be exploited: Stability as fundamental property is essential when the model is used for controller design or movement generation. Therefore, this paper proposes a framework for learning stable stochastic systems from data. We focus on identifying a state-dependent coefficient form of the nonlinear stochastic model which is globally asymptotically stable according to probabilistic Lyapunov methods. We compare our approach to other state of the art methods on real-world datasets in terms of flexibility and stability.

1. Introduction

An accurate identification of the system dynamics is the first and very crucial step to many modern control methods. Although reinforcement learning also allows model-free search for optimal policies, it is known to be less efficient and difficult to analyze. Therefore, classical control engineers employ system identification techniques to obtain parametric model descriptions of dynamical systems from observation data, e.g., in the linear case ARX and ARMAX models. The identification focuses on model selection, i.e., finding the model structure and the corresponding set of parameters. But often this set of model candidates is difficult to find, especially for complex, possibly non-deterministic, systems (Ljung, 1998). Therefore, the need for data-driven models has emerged recently as control engineering is increasingly applied in areas without analytic description of the dynamical system. We consider the following two application scenarios: First, assume a set of trajectories for a robotic task is given through human demonstrations, e.g., object grasping. The goal is to represent the motion with a dynamical system. To ensure the reproduction terminates at the desired final point (object to grasp), we introduce the stability constraint. Second, consider a dynamical system which is known to be stable, e.g., a pendulum which rests in hanging position. The goal is to identify the dynamics precisely without further physical insights.

Bayesian non-parametric methods, more particularly Gaussian Processes (GPs) where successfully employed by Kocijan et al. (2005) and Wang et al. (2005) for system identification. Other approaches focus on learning switching linear systems (Fox et al., 2009) or employ an EM algorithm (Ghahramani & Roweis, 1999) for nonlinear systems. However, these approaches neglect the prior assumption that the dynamical system is stable, which becomes crucial when the learned model is used as a generative process such as in movement generation for robotics (Ijspeert et al., 2002). If stability is not considered during learning, the identified model suffers from spurious attractors which are not part of the true dynamics or instability.

Only little work has merged the extensive knowledge on stability theory from control engineering with the powerful data-driven approaches for system identification: For example Boots et al. (2008) and Chiuso & Pillonetto (2010) take stability constraints for learning dynamical systems into account but are limited to linear systems. The work by Khansari-Zadeh & Billard (2011) ensures stability of the system by constraining the optimization of a Gaussian Mixture Model (GMM) to stability conditions derived from Lyapunov methods. The work by Paraschos et al. (2013) relies on a phase variable to ensure stability, which makes the approach time-dependent and therefore less robust. Control Lyapunov functions are used by Khansari-Zadeh & Billard (2014) to ensure global stability for the learned system. These approaches partially employ probabilistic models (GP, GMM), but limit the analysis to the deterministic part by only considering the mean regression. By discarding the true underlying probability distribution, information regarding reliability of the model provided by the data is lost. This leads to overconfident conclusions regarding performance or safety on the real system.

Therefore, this work proposes a framework for learning

[1] Chair of Information-oriented Control, Technical University of Munich, Munich, Germany. Correspondence to: Jonas Umlauft <jonas.umlauft@tum.de>.

Proceedings of the 34^{th} *International Conference on Machine Learning*, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

probabilistic nonlinear dynamical systems from observation, which takes the prior assumption of stability into account. The required stochastic stability conditions of the discrete-time Markov processes are derived from Lyapunov theory. We provide simulation results to validate the proposed approach and compare it to previously mentioned methods for identifying dynamical systems.

2. Problem Formulation

We consider an autonomous, dynamical, discrete-time system with continuous-valued state $\boldsymbol{x}_k \in \mathcal{X} = \mathbb{R}^d$. The state evolves according to an unknown stochastic process[1]

$$\boldsymbol{x}_{k+1} = \hat{\boldsymbol{f}}(\boldsymbol{x}_k, \hat{\boldsymbol{\omega}}_k), \tag{1}$$

with initial value $\boldsymbol{x}_0 \in \mathcal{X}$ and $\hat{\boldsymbol{\omega}}_k$ is a random variable from the probability space $(\Omega, \mathcal{F}, \mathcal{P})$ with sample space Ω, the corresponding σ-algebra $\mathcal{F}$ and the probability measure $\mathcal{P}$. Since $\boldsymbol{x}_k \in \mathcal{X}$ is fixed at each step, (1) describes a state dependent distribution over $\boldsymbol{x}_{k+1}$. A realization of $\hat{\boldsymbol{\omega}}_k \in \Omega$, is drawn at every time step, yielding a realization of the next step. As the distribution for $\boldsymbol{x}_{k+1}$ only depends on the state at time step k, $\hat{\boldsymbol{f}}$ is a Markov process, denoted by $\{\boldsymbol{x}_k\}$.

We assume that consecutive measurements of the state are available, thus N data pairs are given in the trainings set $\mathbb{D} = \{\bar{\boldsymbol{x}}_n, \bar{\boldsymbol{x}}_{n+1}\}_{n=1}^N$. Based on these measurements, we model the unknown dynamics $\hat{\boldsymbol{f}}$ including the distribution $\hat{\boldsymbol{\omega}}_k$ using the prior knowledge, that the stochastic process (1) converges to the origin $\boldsymbol{x}_k = \boldsymbol{0}$. The model consists of the mapping $\boldsymbol{f}_{\boldsymbol{\psi}}$ and a encoding of the random variable $\boldsymbol{\omega}$ defined by a finite parameter vector $\boldsymbol{\psi} \in \Psi$. As the model $\boldsymbol{f}_{\boldsymbol{\psi}}$ must best possibly explain the data $\mathbb{D}$, the problem is formulated as constrained likelihood maximization

$$\boldsymbol{\psi}^* = \arg\max_{\boldsymbol{\psi}} \sum_{n=1}^{N} \log \mathcal{P}\left(\bar{\boldsymbol{x}}_{n+1} | \bar{\boldsymbol{x}}_n, \boldsymbol{f}_{\boldsymbol{\psi}}\right), \tag{2a}$$

$$\text{s.t. } \{\boldsymbol{x}_k\} \text{ converges to the origin for } k \to \infty. \tag{2b}$$

As different stochastic stability concepts exist, the *convergence* in (2b) is defined as *convergence with probability one* (w.p.1) (Kushner, 1971):

Definition 1 (Convergence w.p.1)**.** *$\{\boldsymbol{x}_k\}$ converges to the origin w.p.1 if, for each $\epsilon > 0$, $\|\boldsymbol{x}_k\| \geq \epsilon$ only finitely often.*

This also implies the following type of convergence, which might be more intuitive to the reader.

Definition 2 (Convergence in probability)**.** *The chain $\{\boldsymbol{x}_k\}$ converges to the origin in probability if $\mathcal{P}(\|\boldsymbol{x}_k\| \geq \epsilon) \to 0$, for each $\epsilon > 0$.*

We do not consider any control input here, thus the identification takes place for the closed-loop system for a existing controller or an uncontrolled system.

3. Stability Conditions for the Model

3.1. The Model

Consider the state-dependent coefficient form of $\boldsymbol{f}_{\boldsymbol{\psi}}$

$$\boldsymbol{x}_{k+1} = A(\boldsymbol{x}_k)\boldsymbol{x}_k, \tag{3}$$

where, for a fixed $\boldsymbol{x}_k$, A is a random variable from the probability space $(\Omega_A, \mathcal{F}_A, \mathcal{P}_A)$ with the sample space $\Omega_A \subseteq \mathbb{R}^{d \times d}$. The probability density function of A is specified by the vector $\boldsymbol{\theta} \in \Theta$, which is state dependent through $\boldsymbol{\theta}^{\boldsymbol{\psi}} : \mathcal{X} \to \Theta$. This mapping is itself parametrized by a vector $\boldsymbol{\psi}$. At each step, a realization of A, denoted by $\bar{A}$, is drawn and multiplied by the state $\boldsymbol{x}_k$ to proceed by one step. This is visualized in Figure 1 along with the two-layer model structure: The first layer maps current state $\boldsymbol{x}_k \in \mathcal{X}$ onto the parameter $\boldsymbol{\theta} \in \Theta$, denoted by $\boldsymbol{\theta}^{\boldsymbol{\psi}} : \mathcal{X} \to \Theta$. The mapping is parametrized by $\boldsymbol{\psi}$. The second layer is the probability distribution on A which assigns to each element in the sample space Ω_A a probability based on $\boldsymbol{\theta}$.

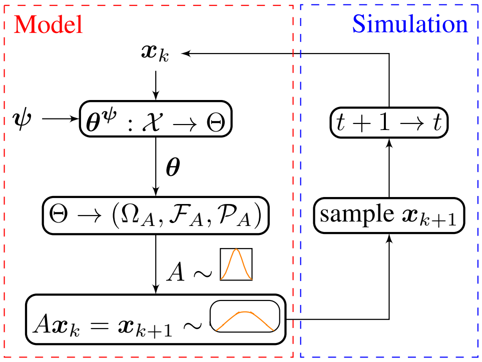
Model
Simulation
$\boldsymbol{x}_k$
$\boldsymbol{\psi}$
$\boldsymbol{\theta}^{\boldsymbol{\psi}} : \mathcal{X} \to \Theta$
$t+1 \to t$
$\boldsymbol{\theta}$
$\Theta \to (\Omega_A, \mathcal{F}_A, \mathcal{P}_A)$
sample $\boldsymbol{x}_{k+1}$
$A \sim$
$A\boldsymbol{x}_k = \boldsymbol{x}_{k+1} \sim$

Figure 1. Illustration of the two layer design of the proposed model and the process of simulation.

To illustrate this multilayer design, we give a brief example in the scalar case $d = 1$: Assume $A(x_k)$ follows, for a given x_k, a Gaussian distribution $A \sim \mathcal{N}(\mu, \sigma)$. Therefore, the parameter vector is $\boldsymbol{\theta}_{\mathcal{N}} = [\mu \ \sigma]^\intercal$ with $\mu \in \mathbb{R}, \sigma \in \mathbb{R}^+$, thus $\Theta_{\mathcal{N}} \subset \mathbb{R} \times \mathbb{R}^+$. The dependency of these parameters on the current state x_k is expressed in $\boldsymbol{\theta}_{\mathcal{N}}^{\boldsymbol{\psi}}$, e.g.,

$$\begin{bmatrix} \mu(x_k) \\ \sigma(x_k) \end{bmatrix} = \boldsymbol{\theta}_{\mathcal{N}}^{\boldsymbol{\psi}}(x_k) = \begin{bmatrix} w x_k \\ z x_k^2 \end{bmatrix}, \tag{4}$$

[1] **Notation:** Bold symbols denote vectors or multivariate functions, capital letters matrices and I_p the $p \times p$ identity matrix. $A \succ 0$ denotes positive definiteness of the matrix A, $\mathbb{E}[\cdot]$ the expected value, $\mathbb{V}[\cdot]$ the variance of a random variable and $\mathbb{C}[\cdot,\cdot]$ the covariance between two random variables, where $\mathbb{C}[\boldsymbol{a}] = \mathbb{C}[\boldsymbol{a}, \boldsymbol{a}]$. $\bar{X}$ denotes a realization of the random variable X. Imitating Matlab indexing, $A_{(:,i)}$ denotes the i-th column, $A_{(j,:)}$ the j-th row and $A_{(1:2,i)}$ the first and second element in the i-the column of A. The i-th entry of the vector $\boldsymbol{x}_k$ is denoted $x_{k,i}$.

where linear dependency of the mean on the state and a quadratic relation between variance and the state is assumed. The parameters defining $\boldsymbol{\theta}^{\boldsymbol{\psi}}_{\mathcal{N}}$ here are $\boldsymbol{\psi} = [w\ z]^\intercal$. Generally, the first layer $\boldsymbol{\theta}^{\boldsymbol{\psi}} : \mathcal{X} \to \Theta$ can be any state of the art parametric regression method which is parametrized by $\boldsymbol{\psi}$. For layer two, any probability distribution with a fixed set of parameters is applicable for A.

Leaving the stochastic aspect aside, model (3) is the state-dependent coefficient (SDC) form which is reached by factorizing a nonlinear system into a linear-like structure. It was shown, that for a any continuous differentiable function f with $f(0) = 0$, their exists a matrix-valued function $A(\boldsymbol{x})$ such that $f(\boldsymbol{x}) = A(\boldsymbol{x})\boldsymbol{x}$, see Cimen (2008). Thus, the SDC form is not limiting the expressive power of our model. It also reflects the setup of many real-world system, e.g., consider an actuator whose output is generally noisy and the magnitude of the noise is dependent on the temperature. By modeling the temperature as a state, the model (3) allows to capture this varying precision of the actuator.

The structure of the model (3) combines two important criteria. First, it provides more flexibility than a linear system with random parameters, so it encodes also nonlinear dynamics.Second, it is simple enough to allow a quadratic Lyapunov function analysis and therefore the derivation of analytic constraints for convergence as needed for the optimization in (2).

3.2. Stability Analysis

For approaching the problem as formulated in Section 2, an analytic condition for the constraint in the optimization problem (2b), given that $\boldsymbol{f}_{\boldsymbol{\psi}}$ is of the form (3), is needed. The literature on stability criteria for dynamical systems is very rich and for nonlinear systems Lyapunov type methods are often used. They are based on the following idea: If there is a function representing the ”energy” in the system (called Lyapunov function) which constantly decreases over time, the state will converge to a ”zero energy” state, the origin. More precise, the Lyapunov function must be positive definite and it must be strictly decreasing over time, except in the origin. Using the stochastic discrete-time version of Lyapunov methods and the Borel-Cantelli Lemma leads to the following conditions for exponential stability (which implies convergence w.p.1 as defined in Definition 1)

Theorem 1 (Exponential Stability, (Kushner, 1971))**.** *Given a positive definite function $V(\boldsymbol{x}_k) \geq 0$ for which*

$$\mathbb{E}\left[V(\boldsymbol{x}_{k+1})|\boldsymbol{x}_k\right] - V(\boldsymbol{x}_k) \leq -\alpha V(\boldsymbol{x}_k),\ \forall \boldsymbol{x}_k \in \mathcal{X} \setminus 0, \tag{5}$$

for some $\alpha > 0$ then

$$\mathbb{E}\left[V(\boldsymbol{x}_{k+m})|\boldsymbol{x}_k\right] \leq (1-\alpha)^m V(\boldsymbol{x}_k) \quad \textit{and} \tag{6}$$

$$V(\boldsymbol{x}_{k+m}) \to 0 \textit{ for } m \to \infty \ (\textit{w.p.1}). \tag{7}$$

For the class of systems in (3) a quadratic function $V(\boldsymbol{x}_k)$ is a proper Lyapunov function to derive sufficient stability constraints for arbitrary distributions on A as shown in the following proposition:

Proposition 1 (Stability of the model (3))**.** *Consider a stochastic process generated from (3) where in each step a realization of A is drawn from sample space $\Omega_A \subset \mathbb{R}^{d\times d}$. The process is globally exponentially stable at $\boldsymbol{x}_k = \mathbf{0}$ if there exists a $P \succ 0$ such that*

$$\mathbb{E}\left[A^\intercal(\boldsymbol{x}_k)\right] P\, \mathbb{E}\left[A(\boldsymbol{x}_k)\right] + Q - (1-\alpha)P \preceq 0,\ \forall \boldsymbol{x}_k \in \mathcal{X}, \tag{8}$$

for some $\alpha > 0$, where Q is defined as

$$Q_{(i,j)}(\boldsymbol{x}_k) = \sum_l P_{(l,:)}\, \mathbb{C}\left[A_{(:,i)}(\boldsymbol{x}_k), A_{(l,j)}(\boldsymbol{x}_k)\right], \tag{9}$$

for any $\boldsymbol{x}_0 \in \mathcal{X}$.

Proof. Considering a quadratic Lyapunov function $V(\boldsymbol{x}_k) = \boldsymbol{x}_k^\intercal P \boldsymbol{x}_k$ with $P \succeq 0$, the inequality from Theorem 1 in (5) is given as

$$\mathbb{E}\left[\boldsymbol{x}_{k+1}^\intercal P \boldsymbol{x}_{k+1}|\boldsymbol{x}_{k+1}\right] - \boldsymbol{x}_k^\intercal P \boldsymbol{x}_k \leq -\alpha \boldsymbol{x}_k^\intercal P \boldsymbol{x}_k,$$

which yields for the stochastic[2] process $\boldsymbol{x}_{k+1} = A(\boldsymbol{x}_k)\boldsymbol{x}_k$

$$\begin{aligned} \boldsymbol{x}_k^\intercal\, \mathbb{E}\left[A^\intercal\right] P\, \mathbb{E}\left[A\right] \boldsymbol{x}_k + \mathrm{Tr}\left(P\, \mathbb{C}\left[A\boldsymbol{x}_k\right]\right) - \\ - (1-\alpha)\boldsymbol{x}_k^\intercal P \boldsymbol{x}_k \leq 0, \quad \forall \boldsymbol{x}_k \in \mathcal{X}. \end{aligned} \tag{10}$$

Now, an expression for the trace is derived as follows

$$\begin{aligned} \mathrm{Tr}\left(P\, \mathbb{C}\left[A\boldsymbol{x}_k\right]\right) &= \mathrm{Tr}\left(P\, \mathbb{C}\left[\sum_i A_{(:,i)} x_{k,i}\right]\right) \\ &= \mathrm{Tr}\left(P \sum_{i,j} x_{k,i} x_{k,j}\, \mathbb{C}\left[A_{(:,i)}, A_{(:,j)}\right]\right) \\ &= \sum_{i,j,l} P_{(l,:)}\, \mathbb{C}\left[A_{(:,i)}, A_{(l,j)}\right] x_{k,i} x_{k,j}, = \boldsymbol{x}_k^\intercal Q \boldsymbol{x}_k \end{aligned}$$

where definition of Q in (9) was substituted. Using this simplification, (10) is rewritten as

$$\boldsymbol{x}_k^\intercal \left(\mathbb{E}\overset{\intercal}{[A]} P\, \mathbb{E}\left[A\right] + Q - (1-\alpha)P\right) \boldsymbol{x}_k \leq 0,$$

which must hold for $\forall \boldsymbol{x}_k \in \mathcal{X}$. To ensure this, the matrix $\mathbb{E}\left[A\right]^\intercal P\, \mathbb{E}\left[A\right] + Q - (1-\alpha)P$ must be negative semi-definite, which concludes the proof. □

[2] The $\boldsymbol{x}_k$ dependency of the random process A as been dropped for notational convenience.

The interpretation of Proposition 1 is analogue to the linear deterministic case $\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k$ which is stable if there exists a matrix P for which $A^\intercal PA - P \prec 0$: In the nonlinear case in (3) the negative definiteness must be fulfilled for $A(\boldsymbol{x}_k)$, $\forall \boldsymbol{x}_k \in \mathcal{X}$. The probabilistic nature of the system (3) in addition requires ”a buffer”, which here is Q. The deterministic case is reconstructed if A has zero variance. The scalar case, considered in the following remark, also allows an intuitive insight to the Proposition 1: There is a trade-off between the magnitude of the expected value and the variance of A as follows:

Remark 1. *In the scalar case*[3]*, i.e. $d = 1$ in (3), with $Q = P\,\mathbb{V}\left[A(x_k)\right]$ condition (8) simplifies for any $P > 0$ to*

$$\mathbb{E}\left[A(x_k)\right]^2 + \mathbb{V}\left[A(x_k)\right] \leq 1 - \alpha, \quad \forall x_k \in \mathcal{X}. \tag{11}$$

4. Stable Learning with Various Distributions

Our learning framework consists of three major steps:

1. Chose any probability distribution for the random variable A in (3) which is given by a fixed set of parameters $\boldsymbol{\theta} \in \Theta$ and whose first two moments are available. It is assumed that subset $\Theta^* \subseteq \Theta$ for which (8) is fulfilled is non-empty, thus $\Theta^* \neq \emptyset$.

2. Chose any parametric regression method to represent the mapping $\boldsymbol{\theta}^{\boldsymbol{\psi}} : \mathcal{X} \to \Theta$. The parameters of this mapping are denoted by $\boldsymbol{\psi} \in \Psi$. The set of all $\boldsymbol{\psi}$ for which all $\boldsymbol{x}_k \in \mathcal{X}$ map to Θ^* is denoted by Ψ^*.

3. The likelihood maximization under constraints

$$\boldsymbol{\psi}^* = \arg \max_{\boldsymbol{\psi} \in \Psi^*} \sum_{n=1}^{N} \log \mathcal{P}\left(\boldsymbol{x}_{n+1} | \boldsymbol{x}_n, \boldsymbol{\theta}^{\boldsymbol{\psi}}\right), \tag{12}$$

is solved, where $\boldsymbol{\psi} \in \Psi^*$ is equivalent to constraint (8) with $P \succ 0$ and $\alpha > 0$.

The optimization (12) is a general constrained nonlinear program in a rather high dimensional space (depending on number of parameters of the regression method in step 2). However, independent of the optimality, the model $\boldsymbol{f}_{\boldsymbol{\psi}^*}$ of the form (3) is exponentially stable, thus any sample path of the system converges. For computational simplicity, we focus on two types of distribution which naturally fulfill the constraints as explained in the next sections.

4.1. Stability with Beta Distribution

For certain choices of distributions, constraint (8) is fulfilled for all possible parameter $\boldsymbol{\theta}$, thus $\Theta^* = \Theta$, which makes the optimization unconstrained. One example of such a distribution is the Beta distribution as given in the following corollary.

Corollary 1. *The scalar system $x_{k+1} = A(x_k)x_k$ where $A(x_k) = \kappa(\tilde{A}(x_k) - \eta)$ with Beta distributed $\tilde{A}(x_k) \sim \mathcal{B}(a(x_k), b(x_k))$ and $\kappa = 2$, $\eta = 0.5$ with state dependent parameters $[a(x_k)\ b(x_k)]^\intercal = \boldsymbol{\theta}_{\mathcal{B}}^{\boldsymbol{\psi}}(x_k)$, with any $\boldsymbol{\theta}_{\mathcal{B}}^{\boldsymbol{\psi}} : \mathcal{X} \to \Theta_{\mathcal{B}} = \mathbb{R}_+^2$ is exponentially stable.*

Proof. Applying the affine transformation to mean and variance leads to[4]

$$\mathbb{E}\left[A(x_k)\right] = \kappa\left(\mathbb{E}\left[\tilde{A}(x_k)\right] - \eta\right) = \kappa\left(\frac{a}{a+b} - \eta\right),$$
$$\mathbb{V}\left[A(x_k)\right] = \kappa^2\,\mathbb{V}\left[\tilde{A}(x_k)\right] = \kappa^2 \frac{ab}{(a+b)^2(a+b+1)}.$$

Condition (11) is rewritten to

$$\left(\kappa(\mathbb{E}\left[A(x_k)\right] - \eta)\right)^2 + \kappa^2\,\mathbb{V}\left[A(x_k)\right] \leq 1 - \alpha, \quad \forall x_k \in \mathcal{X},$$

where the best possible choice for η minimizes $(\mathbb{E}\left[A(x_k)\right] - \eta)^2$, because it leaves the largest possible range for κ. As $\mathbb{E}\left[A\right]$ is in the interval $]0, 1[$ the minimization is achieved with the choice $\eta = \frac{1}{2}$. Then, condition (11), divided by κ^2 on both sides, evaluates to

$$\frac{a^2}{(a+b)^2} - \frac{a}{a+b} + \frac{1}{4} + \frac{ab}{(a+b)^2(a+b+1)} =$$
$$-\underbrace{\frac{ab}{(a+b)(a+b+1)}}_{0\ldots\frac{1}{4}} + \frac{1}{4} \leq \frac{1-\alpha}{\kappa^2}.$$

As $\alpha > 0$ can be chosen arbitrarily small this condition holds for every $|\kappa| \leq 2$. Hence, according to Theorem 1 the system $x_{k+1} = A(x_k)x_k$ is exponentially stable. □

To ensure maximal flexibility of the model, $\kappa = 2$ is set for further considerations. This leads to the conclusion that $\Theta_{\mathcal{B}}^* = \Theta_{\mathcal{B}} = \mathbb{R}_+^2$. Therefore, in the optimization, no constraints on $\boldsymbol{\psi}$ must be considered, thus $\Psi = \Psi^*$.

4.2. Stability with Dirichlet Distribution

Constructing A from a Dirichlet distribution also allows for unconstrained optimization as it also leads to stable behavior as shown in the following corollary.

Corollary 2. *The d-dimensional system $\boldsymbol{x}_{k+1} = A(\boldsymbol{x}_k)\boldsymbol{x}_k$ where each row of $A(\boldsymbol{x}_k)$ consists of the first d elements of a $d+1$ dimensional Dirichlet distributed vector, thus,*

$$A_{(i,:)} = \boldsymbol{a}_{(1:d)}^{(i)}, \text{ with } \boldsymbol{a}^{(i)} \sim \mathcal{D}\left(\boldsymbol{\theta}_{\mathcal{D}}^{\boldsymbol{\psi}_i}(\boldsymbol{x}_k)\right), \quad \forall i = 1 \ldots d,$$

with any $\boldsymbol{\theta}_{\mathcal{D}}^{\boldsymbol{\psi}_i} : \mathcal{X} \to \Theta_{\mathcal{D}}, \forall i$, is asymptotically stable w.p.1.

[3] Even though A, Q, P are scalars here, we keep them capitalized for notational consistency.

[4] The state dependency of a, b is dropped for notational convenience.

Proof. By construction the sample space of A, Ω_A contains only elements for which

$$A_{(i,j)} > 0, \quad A_{(i,j)} < 1, \qquad \forall i,j = 1 \dots d$$
$$\text{and} \sum_{j=1}^{d} A_{(i,j)} < 1, \qquad \forall i = 1 \dots d. \tag{13}$$

Consider now a realization of $A(\boldsymbol{x}_k)$ denoted by $\bar{A}$ and since the following statements hold for any realization in the sample space, we omit writing $\forall \bar{A} \in \Omega_A$. It follows

$$\sum_{j=1}^{d} \bar{A}_{(i,j)} < 1 \, \forall i \quad \Rightarrow \quad \max_{i=1:d} \sum_{j=1}^{d} \bar{A}_{(i,j)} < 1$$
$$\Rightarrow \quad \|\bar{A}\|_\infty = \max_{i=1:d} \sum_{j=1}^{d} |\bar{A}_{(i,j)}| < 1,$$

where the last inequality holds because all elements of $\bar{A}$ are strictly positive and $\|\bar{A}\|_\infty$ denotes the Maximum Absolute Row Sum Norm. Consider now M consecutive realizations $\bar{A}^{(i)}$ with $i = 1, \dots, M$. For the maximum norm of state in the M-th step holds

$$\|\boldsymbol{x}_{k+M}\|_\infty = \left\| \prod_{m=1}^{M} \bar{A}^{(m)} \boldsymbol{x}_k \right\|_\infty \leq \prod_{m=1}^{M} \left\| \bar{A}^{(m)} \right\|_\infty \|\boldsymbol{x}_k\|_\infty$$
$$\leq \left(\max_m \left\| \bar{A}^{(m)} \right\|_\infty \right)^M \|\boldsymbol{x}_k\|_\infty \xrightarrow{M \to \infty} 0,$$

where the submultiplicativity property of induced matrices (Horn & Johnson, 2013) is used. As convergence towards the origin holds for each element in the sample space, the system is stable with probability one. Therefore, the parameter space is unrestricted $\Theta_{\mathcal{D}}^* = \Theta_{\mathcal{D}} = \mathbb{R}_+^{d+1}$. □

Remark 2. *Note that this approach only allows to represent the special class of positive systems. Nevertheless, positive systems play an important role in control engineering for modeling the evolution of strictly positive quantities as shown in (Farina & Rinaldi, 2011).*

Remark 3. *An affine transformation, as shown for Beta distribution is not possible here because absolute values are taken in the row sum. Therefore, from* $\sum_{j=1}^{d} |\bar{A}_{(i,j)}| < 1$ *one cannot conclude* $\sum_{j=1}^{d} |\kappa(\bar{A}_{(i,j)} - 0.5)| < 1$ *for any* $\kappa > 1$.

5. Simulations

5.1. Setup

We validate our approach, labeled *LeSSS* (for **Le**arning **S**table **S**tochastic **S**ystems), using synthetic and human motion data and the simulation of a chemical reactor. For the Beta distribution, Gaussian Mixture Regression (GMR) is used for the mapping from the state to the parameters $\boldsymbol{\theta}_{\mathcal{B}} : \mathcal{X} \to \Theta_{\mathcal{B}}$. Thus, the parameter vector $\boldsymbol{\psi}$, is the concatenation of the prior π_l, the means $\boldsymbol{\mu}_l$ and the covariances Σ_l for $l = 1, \dots L$. The code (based on Calinon (2009)) includes k-means clustering initialization and a transformation of Σ_l and π_l to make it an unconstrained optimization. To evaluate the likelihood function for each training point $\{\bar{x}_{n+1}, \bar{x}_n\}$, the Beta distribution parameters are computed $[a_n \; b_n]^\intercal = \boldsymbol{\theta}_{\mathcal{B}}^{\boldsymbol{\psi}}(\bar{x}_n)$ using GMR. Then, the log likelihood of $\bar{A}_n = \bar{x}_{n+1}/\bar{x}_n$ given the parameters $[a_n \; b_n]$ is evaluated using the density function of the Beta distribution. As all possible parameter $\boldsymbol{\theta}_{\mathcal{B}} = [a \; b]^\intercal \in \mathbb{R}_+^2$ lead to stability, finding $\boldsymbol{\psi}^*$ is an unconstrained optimization problem.

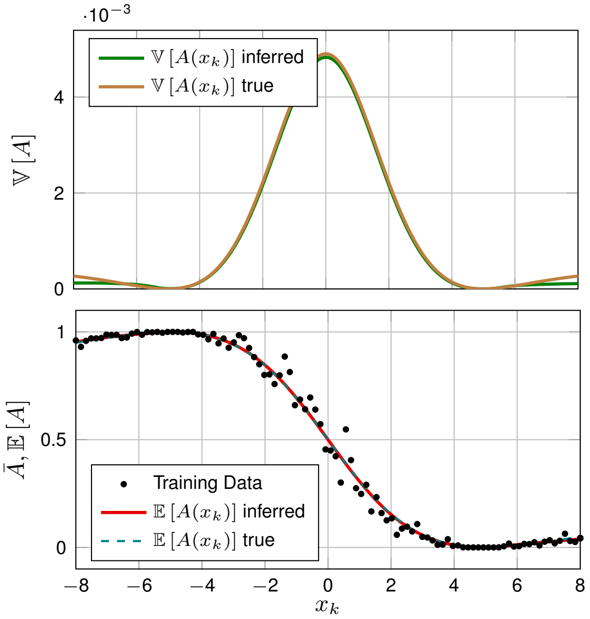

Figure 2. Comparison of true and inferred x_k dependent variance (top) and mean (bottom) function.

For the Dirichlet distribution, the mapping from the state to the parameters $\boldsymbol{\theta}_{\mathcal{D}}^{\boldsymbol{\psi}} : \mathcal{X} \to \Theta_{\mathcal{D}}$ uses a nearest neighbor approach for computational simplicity. The $2d = 4$ closest data points are considered for fitting the training parameters of the Dirichlet distribution locally. Then a training point is placed at the center of these four points. At reproduction, the closest such training point and its Dirichlet parameters are taken for regression. This does not necessarily maximize the likelihood, but shows accurate results for reproduction. We compare the following models from literature regarding reproduction precision and convergence properties:

- The approach introduced by Boots et al. (2008) learns stable linear dynamical system (stable LDS) from data. It constraints the search of the deterministic dynamic matrix A to ensure the stability of $\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k$.

- Gaussian Process Dynamical Models (GPDM) (Wang et al., 2005) represent dynamical system in the general form $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k)$, with Gaussian Process $\boldsymbol{f} \sim \mathcal{GP}(\boldsymbol{0}, k(\boldsymbol{x}_k, \boldsymbol{x}'_k))$. We employ a zero prior mean function and a squared exponential kernel. The hyperparameters of the kernel are optimized using the likelihood as described by Rasmussen & Williams (2006). In reproduction, this method can either be used in deterministic setting by only taking the posterior mean prediction $\boldsymbol{\mu}_{\text{GP}}(\boldsymbol{x}_k)$ thus $\boldsymbol{x}_{k+1} = \boldsymbol{\mu}_{\text{GP}}(\boldsymbol{x}_k)$ or the stochastic setting $\boldsymbol{x}_{k+1} \sim \mathcal{N}(\boldsymbol{\mu}_{\text{GP}}(\boldsymbol{x}_k), \Sigma_{\text{GP}}(\boldsymbol{x}_k))$, where $\Sigma_{\text{GP}}(\boldsymbol{x}_k)$ is the posterior variance. GPDMs are bounded (Beckers & Hirche, 2016a;b) but not stable.
- The Stable Estimator of Dynamical Systems (SEDS) as introduced by Khansari-Zadeh & Billard (2011) constraints the likelihood optimization of GMR parameters to a class of mean stable dynamical time-continuous systems. The GMR maps from current state $\boldsymbol{x}$ to the time derivative $\dot{\boldsymbol{x}}$. It focuses on deterministic systems by only considering stability criteria for the mean prediction of the GMR, $\boldsymbol{\mu}_{\text{GMM}}(\boldsymbol{x})$, while ignoring the stochastic nature of GMMs, (its variance prediction Σ_{GMM}). We also run this method in a stochastic setting, where $\dot{\boldsymbol{x}} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\text{GMM}}(\boldsymbol{x}), \Sigma_{\text{GMM}}(\boldsymbol{x})\right)$. For our simulations, five mixtures are employed.

Before starting the comparison to existing approaches, *LeSSS* is demonstrated on a synthetic dataset.

5.2. Simulation 1: Synthetic Data

For the first simulation, the task is to identify the stable nonlinear stochastic system given by

$$x_{k+1} = A(x_k)x_k, \tag{14}$$
$$\text{where } A(x_k) \sim \mathcal{B}\left((x_k - 5)^2, (x_k + 5)^2\right).$$

The learning algorithm is given 100 training points $\{\bar{x}_n, \bar{x}_{n+1}\}_{n=1}^{100}$ equally spaced in the state space interval $[-8, 8]$ which are drawn from the state dependent Beta distribution (14). Here $L = 3$ was chosen for the number of mixtures in the GMR for the mapping $\boldsymbol{\theta}_{\mathcal{B}}^{\psi} : \mathcal{X} \to \Theta_{\mathcal{B}}$.

Figure 2 compares the mean and variance of the original system (14) to the one inferred by our model. It clearly shows that the model offers sufficient flexibility to reconstruct the original system. Note: It is also possible to verify the parameter functions $a(x_k), b(x_k)$ as given in (14), but we directly look at the mean and variance functions as there exists a unique mapping and it is more intuitive for interpretation. It must be omitted, that the data was generated from the same model which the algorithm is learning. This explains the good fitting, but is of course not often the case in practical application. Therefore, we continue with a real world dataset in the following.

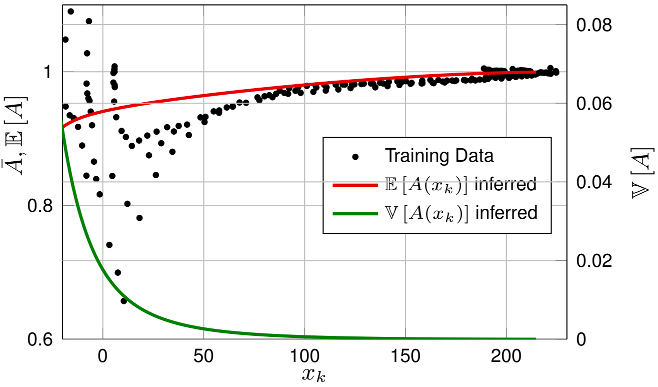

Figure 3. The inferred mean and variance function for the training set of the projected Z-shape movement are shown along with the training data (the realizations of the random variable $\bar{A}$).

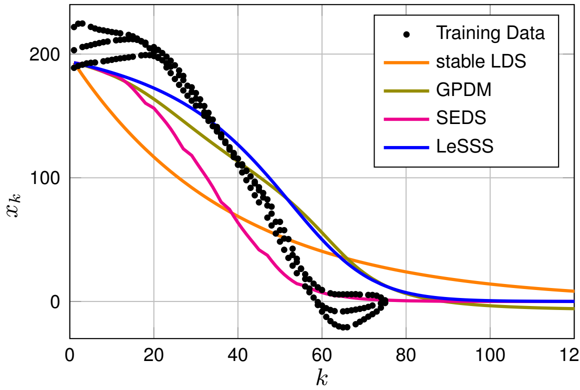

Figure 4. Comparison of simulations for $x_0 = 193$ for *stable LDS* and the mean predictions from the models *GPDM*, *SEDS* and *LeSSS*.

5.3. Simulation 2: Human Motion Data

For the next simulation, we use the data set for letter-shaped motions provided by Khansari-Zadeh & Billard (2011). The 225 trainings points of 3 trajectories of the two dimensional Z-shaped motion are projected on the y-axis. The GMR for $\boldsymbol{\theta}_{\mathcal{B}}^{\psi} : \mathcal{X} \to \Theta_{\mathcal{B}}$ is trained with two mixtures.

Figure 3 shows the training data along with the fit of the mean and variance functions. The mean function shows a smoothed estimate of the training data. The model identifies properly that the training data has higher variability (around $x_k = 0$) and captures this in its variance function.

Figure 4 compares the reproduction of the models *stable LDS*, *GPDM*, *SEDS* and *LeSSS* if taking the deterministic (mean) output of each model (all starting from the same

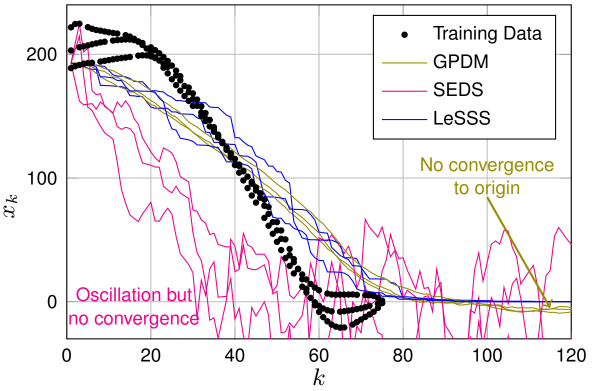

Figure 5. Comparison of simulations for $x_0 = 193$ with the stochastic models of *GPDM*, *SEDS* and *LeSSS*.

initial point). The *stable LDS* approach leads to a converging trajectory, but fails to capture the complexity of the dynamic (as the true dynamic is nonlinear). The *GPDM* converges to a spurious attractor at $x \approx -9.3$ which is undesired but not surprising. *SEDS* and *LeSSS* both lead to asymptotic stable reproductions of the movement. Since the data does not contain the full state (due to the projection on the y-axis), it is not possible to reproduce the movement precisely with a dynamical system model.

Figure 5 compares the reproduction of the three stochastic dynamical models *GPDM*, *SEDS* and *LeSSS* based on three sample paths drawn from each model. The *GPDM* again converges to the spurious attractor. *SEDS* clearly shows that convergence of the mean is not sufficient for converging trajectories of a stochastic system, as the drawn sample paths are strongly oscillating around the origin without tendency to converge. In the stochastic case only *LeSSS* generates converging trajectories.

Figure 6 shows an example for the human motion imitation in the 2D case on a different training data set. It shows the deterministic trajectory and 5 sample path realizations, where all of them show high reproduction precision and convergence to the orign.

5.4. Simulation 3: Chemical Reactor Simulation

For the last validation, we utilize simulated data from a simplified chemical reactor (Einarsson, 1998). The closed-loop reactor is modeled by a piecewise affine system with two states: the fluid level x_1 and the temperature x_2. Both states are physically positive quantities, therefore the approach in Section 4.2 is suitable. The switching between different dynamic matrices is state dependent and occurs at $x_1 = 3$ and $x_2 = 50$, which corresponds to a discrete change of the control inputs. The training data consists of 8 trajectories of 15 steps each, which are pairwise initialized at the 4 different regions of the dynamics and perturbed with white noise with $\sigma = 0.01$ for both states.

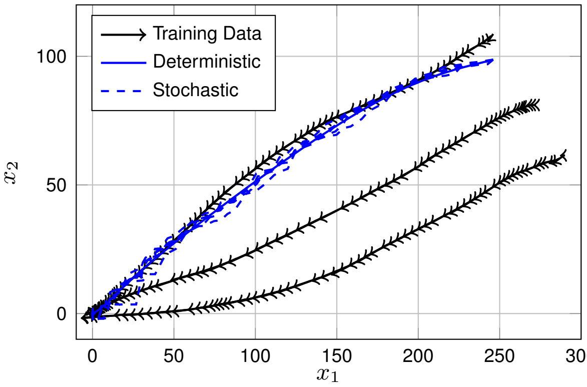

Figure 6. 2D simulation for human motion data set with *LeSSS*.

Error/stable?	*stable LDS*	*GPDM*	*SEDS*	*LeSSS*
deterministic	322/yes	173/no	332/yes	162/yes
stochastic	n/a	177/no	364/no	165/yes

Table 1. Comparison of the reproduction error in terms of area between each demonstration and its corresponding reproduction for the stochastic and deterministic case for the chemical reactor simulation. The area is computed for each of the 8 initial points separately and cumulated for each of the approaches. It also indicates which models are stable.

Figure 7 shows the training data along with the reproduction using *stable LDS*, *GPDM*, *SEDS* and *LeSSS* in the deterministic setting. The initial points in the test case were set close to the one in the training data. The *stable LDS* is not capable to capture the varying behavior in the different regions of the piecewise affine system and therefore fails in accuracy of the reproduction. *GPDM* leads again to convergence outside the origin, which is undesirable. *SEDS* and *LeSSS* are both converging as it is enforced by design. Figure 8 shows that similar to the 1D case *GPDM* and *SEDS* fail to converge in the stochastic case while *LeSSS* is stable in all sample paths. Table 1 compares the methods with regard to the reproduction precision quantitatively. It shows that *LeSSS* outperforms other methods in this measure while providing the necessary guarantees regarding convergence.

5.5. Discussion

The simulations show that *LeSSS* is powerful enough to represent various nonlinear dynamics, while capturing the probabilistic nature of the process. The incorporation of the prior knowledge on goal convergence ensures that the learned model is stable in probability.

The computational complexity for learning the parameters of the model using interior-point methods, is mainly determined by the employed mapping in the first layer $\boldsymbol{\theta}^{\psi}$. The

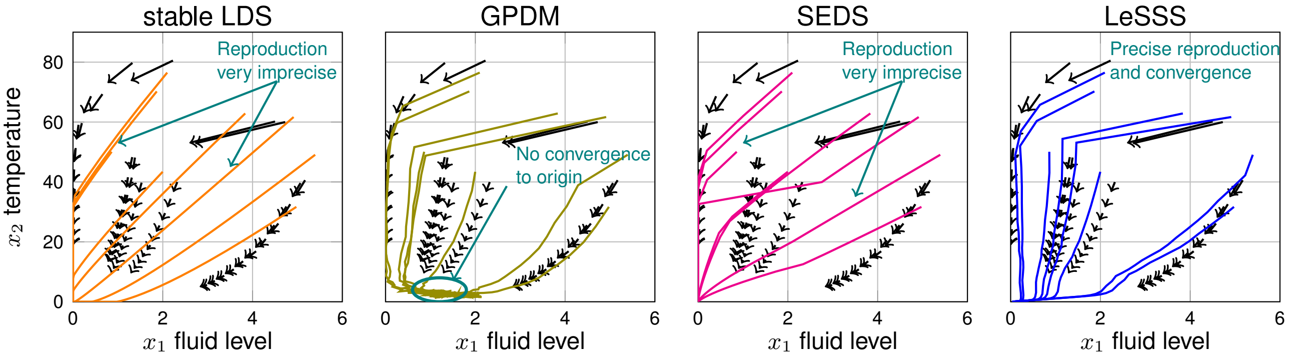

Figure 7. Training data (black) of chemical reactor and the deterministic simulation for *stable LDS*, *GPDM*, *SEDS* and *LeSSS* approach.

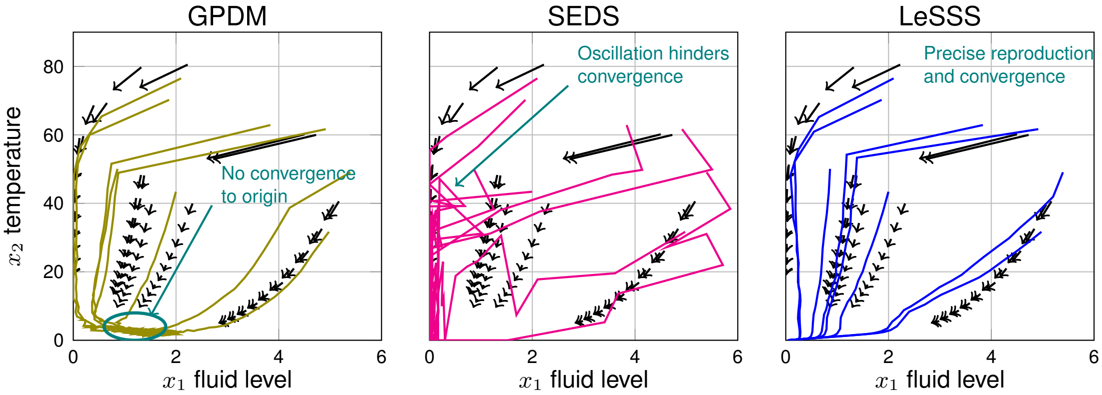

Figure 8. Training data (black) of chemical reactor and the stochastic simulation for *GPDM*, *SEDS* and *LeSSS* approach.

Simulation 2	LDS	GPDM	SEDS	LeSSS
Time	0.016s	8.53s	2.47s	9.88s
Simulation 3	LDS	GPDM	SEDS	LeSSS
Time	4.54s	11.2s	22.3s	18.2s

Table 2. Computation times for model learning and simulation for all compared approaches.

computation times on a i5 CPU 2.30GHz, 2 Cores and 8GB RAM are given for Simulation 2 and 3 in Table 2. Since the *GPDM*, *SEDS* and *LeSSS* all solve non-convex optimization problems, their commutation times are in the same order of magnitude. The linear model has advantage here. Regarding the scalability with more training points, the parameter fitting performs similarly to other approaches requiring likelihood computation since this is the major factor. However, the scalability strongly depends on the employed distribution and the mapping in the first layer $\boldsymbol{\theta}^{\psi}$.

This work only deals with system with a single equilibrium point, but could be extended to system with more complex attractor dynamics. However, further knowledge is required because - in addition to the position of all equilibrium points - their regions of attraction must be known.

6. Conclusion

This work proposes a framework for learning nonlinear stable stochastic dynamical systems from data. We introduce a flexible model, which builds on the state-dependent coefficient form and derive exponential stability conditions based on stochastic Lyapunov methods. The criteria is applicable to various probability distributions, while we focus to investigate the application to Beta and Dirichlet distributions. Simulation results verify sufficient flexibility of the model and the correct identification of the system's uncertainty. In comparison to existing approaches it showed advantages in reproduction precision and convergence properties on human motion data and simulated data from a real system.

Acknowledgment

The research leading to these results has received funding from the European Research Council under the European Union Seventh Framework Program (FP7/2007-2013) ERC Starting Grant ”Control based on Human Models (conhumo)” agreement number 337654. We also would like to thank the reviewers for very constructive feedback.

References

Beckers, Thomas and Hirche, Sandra. Equilibrium distributions and stability analysis of Gaussian process state space models. In *Conference on Decision and Control (CDC)*, pp. 6355–6361, 2016a.

Beckers, Thomas and Hirche, Sandra. Stability of Gaussian process state space models. In *European Control Conference*, pp. 2275–2281, 2016b.

Boots, Byron, Gordon, Geoffrey J, and Siddiqi, Sajid M. A constraint generation approach to learning stable linear dynamical systems. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1329–1336. Curran Associates, Inc., 2008.

Calinon, Sylvain. *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009.

Chiuso, Alessandro and Pillonetto, Gianluigi. Learning sparse dynamic linear systems using stable spline kernels and exponential hyperpriors. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 397–405. Curran Associates, Inc., 2010.

Cimen, Tayfun. State-dependent Riccati equation (SDRE) control: A survey. *IFAC Proceedings Volumes*, 41(2): 3761–3775, 2008.

Einarsson, Valur. *On Verification of Switched Systems using Abstractions*. PhD thesis, Linkoping University, Automatic Control, The Institute of Technology, 1998.

Farina, Lorenzo and Rinaldi, Sergio. *Positive linear systems: Theory and applications*. John Wiley&Sons, 2011.

Fox, Emily, Sudderth, Erik B., Jordan, Michael I., and Willsky, Alan S. Nonparametric Bayesian learning of switching linear dynamical systems. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 457–464. Curran Associates, Inc., 2009.

Ghahramani, Zoubin and Roweis, Sam T. Learning nonlinear dynamical systems using an EM algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 431–437. MIT Press, 1999.

Horn, Roger A. and Johnson, Charles R. *Matrix analysis*. Cambridge Univ. Press, Cambridge, 2013.

Ijspeert, Auke Jan, Nakanishi, Jun, and Schaal, Stefan. Movement imitation with nonlinear dynamical systems in humanoid robots. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2002.

Khansari-Zadeh, Seyed Mohammad and Billard, Aude. Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Transactions on Robotics*, 27 (5):943–957, 2011.

Khansari-Zadeh, Seyed Mohammad and Billard, Aude. Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6):752–765, 2014.

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

Kushner, Harold Joseph. *Introduction to stochastic control*. Holt, Rinehart and Winston New York, 1971.

Ljung, Lennart. *System Identification*. Prentice Hall PTR, NJ, USA, 1998.

Paraschos, Alexandros, Daniel, Christian, Peters, Jan, and Neumann, Gerhard. Probabilistic movement primitives. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2616–2624, 2013.

Rasmussen, Carl Edward and Williams, Christopher KI. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA, January 2006.

Wang, Jack M., Fleet, David J., and Hertzmann, Aaron. Gaussian process dynamical models. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1441–1448, 2005.