Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 51-34 (2019) 8–14

Uncertainty-based Human Motion Tracking with Stable Gaussian Process State Space Models

Lukas Pöhler Jonas Umlauft * **Sandra Hirche** *

* *Chair of Information-oriented Control, Department of Electrical and Computer Engineering, Technical University of Munich, D-80333 Munich, Germany (e-mail: [lukas.poehler,jonas.umlauft,hirche]@tum.de)*

Abstract: Data-driven approaches are well suited to represent human motion because arbitrary complex trajectories can be captured. Gaussian process state space models allow to encode human motion while quantifying uncertainty due to missing data. Such human motion models are relevant for many application domains such as learning by demonstration and motion prediction in human-robot collaboration. For goal-directed tasks it is essential to impose stability constraints on the model representing the human motion. Motivated by learning by demonstration applications, this paper proposes an uncertainty-based control Lyapunov function approach for goal-directed path tracking. We exploit the model fidelity which is related to the location of the training and test data: Our approach actively strives into regions with more demonstration data and thus higher model certainty. This achieves accurate reproduction of the human motion independent of the initial condition and we show that generated trajectories are uniformly globally asymptotically stable. The approach is validated in a nonlinear learning by demonstration task where human-demonstrated motions are reproduced by the learned dynamical system, and higher precision than competitive state of the art methods is achieved.

Keywords: Path tracking, Human centered automation, Modeling of human performance, Nonlinear system identification, Lyapunov methods, Control under uncertainty

1. INTRODUCTION

Control engineering is increasingly applied at the intersection of human and technical systems. Prominent examples are learning by demonstration scenarios and human-robot collaboration. A key aspect to successfully incorporate human behavior is the representation and prediction of humans. However, volunteer human motion is difficult to model based on first principles and is characterized by a high degree of nonlinearity. Thus, conventional physics-based parametric models are not suitable to represent human movement. Recent technological advances in sensor technology and data processing allow data-driven modeling with high precision and data-efficiency. Therefore, data-driven approaches are a promising path to model and flexibly reproduce motions through the symbiosis of powerful algorithms from machine learning and well-founded mathematical tools from control theory. Non-parametric (data-driven) models offer the advantage over parametric models that no bias is implied towards a particular class of system and the set of candidate model parameters is not restricted. With only minimal prior knowledge their complexity adapts to the amount of training data and possibly increases the number of parameters infinitely. This makes non-parametric models well-suited for human motion representation, as in Lang et al. (2017).

Movement tasks in human centered applications often rely on goal-directed expert demonstrations. For their replication, dynamical systems generating the motion are beneficial due to smoothness of resulting trajectories, robustness to perturbation, time independence and real-time capability. The objective is to learn a dynamical system, which generates stable trajectories similar to the training data independently of the initial condition. However, recording training data is time consuming and therefore the state space is only covered sparsely. Thus, data-efficient methods are needed in learning by demonstration applications to achieve a high similarity of the reproduction.

When data-driven methods are applied to model goal-directed motions, guaranteed convergence to a set point of the resulting dynamical system is not guaranteed. A first approach for stable movement generation based on demonstrations are dynamic movement primitives (DMPs) introduced by Ijspeert et al. (2002). It is based on a stable linear second-order system which is augmented by a nonlinear component learned from demonstrations. The stable estimator of dynamical system (SEDS) approach presented by Khansari-Zadeh and Billard (2011) builds on parametric Gaussian mixture models (GMMs). Learning the parameters based on demonstrations is restricted to stable GMMs using a Lyapunov condition. Blocher et al. (2017) stabilize GMMs using contraction theory. However, these approaches rely on manually set parameters and do not consider uncertainty of the learned system from sparse data. Recently, the encoding of desired trajectories as autonomous dynamical system using the non-

2405-8963
10.1016/j.ifacol.2019.01.002

parametric Gaussian process (GP) has emerged due to the following favorable properties: implicit bias/variance trade-off, incorporation of prior knowledge about the data due to Bayesian nature, and the quantifiable model uncertainty from missing data (Rasmussen and Williams, 2006). First steps on enforcing stability for GPs as data-driven probabilistic models are discussed by Khansari-Zadeh and Billard (2014), Umlauft et al. (2018) and Umlauft et al. (2017b). Most applications of GPs do not take advantage of the inferred model fidelity since it is ignored, considered as process noise in the system (Beckers and Hirche, 2016) or used for other control tasks, e.g. gain tuning as in Beckers et al. (2017). However, regions of the state space where the model is uncertain bare a risk of undesired behavior and should be avoided.

This paper considers a learning by demonstration scenario with the aim to precisely model human goal-directed trajectories using dynamical systems. We propose a framework to generate guaranteed converging uncertainty-avoiding trajectories from a stabilized Gaussian process state space model (GPSSM). We propose an uncertainty-based control Lyapunov approach which takes advantage of the GPs quantified model fidelity. By applying a gradient descent on this Lyapunov function, asymptotically stable trajectories are generated along regions in the state space with low uncertainty, thus approach the demonstrations and avoid uncertain regions. We verify the path tracking in simulations and show that it outperforms state of the art methods on a real-world human demonstration dataset with respect to tracking precision. After defining the problem setting in Section 2, Section 3 reviews learning of stabilized GPSSMs from demonstration data. Section 4 introduces the uncertainty-based control Lyapunov function, proofs its properties and presents our approach for path tracking. Simulation results are shown in Section 5.

Notation: Lower/upper case bold symbols denote vectors/matrices, $\mathbb{R}_{+,0}/\mathbb{R}_+$ all real positive numbers with/ without zero and $\mathbb{E}[\cdot]/\mathrm{var}[\cdot]$ expected value/variance of a random variable, respectively. $\cdot \succ 0$ denotes the positive definiteness of matrices/functions, $\boldsymbol{I}_n$ the $n \times n$ identity matrix, $\|\cdot\|$ the Euclidean norm if not specified otherwise and $\Delta_{\boldsymbol{x}} f$ the gradient of a function f with respect to $\boldsymbol{x}$. The Lie derivative of a scalar function $V : \mathbb{R}^n \to \mathbb{R}$ along a vector field $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ is denoted by $\mathcal{L}_{\boldsymbol{f}} V(\boldsymbol{x})$.

2. PROBLEM FORMULATION

We assume a training data set $\mathcal{D}$ of human trajectories of a goal-directed task being given. $\mathcal{D}$ contains the demonstrations and consists of N data pairs representing the current location as continuous-valued state $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ and a noisy version of the velocity

$$\mathcal{D} = \left\{ \left(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)} \right) \right\}_{i=1}^{N}, \qquad \boldsymbol{y}^{(i)} = \dot{\boldsymbol{x}}^{(i)} + \boldsymbol{\omega}^{(i)}, \quad (1)$$

where $\boldsymbol{\omega}^{(i)} \sim \mathcal{N}(0, \sigma_{\text{on}}^2 \boldsymbol{I}_n)$ are i.i.d. samples with $\sigma_{\text{on}}^2 \in \mathbb{R}_+$.

We assume the data originates from a dynamical system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \quad (2)$$

where the function $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^n$ is unknown but the following assumptions are made.

Assumption 1. The function $\boldsymbol{f}(\boldsymbol{x})$ is Lipschitz continuous.

Assumption 2. The dynamical system (2) is uniformly globally asymptotically stable at $\boldsymbol{x} = \boldsymbol{0}$.

Assumption 1 is a mild assumption and can be concluded from the physical nature that a human demonstrator can only expend limited force onto his body leading to finite acceleration. The goal-directed demonstrations lead to Assumption 2 for the underlying system. Without loss of generality, the goal position is here set to the origin.

Our goal is to find a model

$$\dot{\boldsymbol{x}} = \hat{\boldsymbol{f}}(\boldsymbol{x}), \quad (3)$$

for which the average reproduction error between the generated and demonstrated path is minimized and the uniform global asymptotic stability of system (3) is guaranteed. Thus, we aim for precise path tracking by reducing the distance of the states $\boldsymbol{x}(t, \boldsymbol{x}_0)$ on the path starting at $\boldsymbol{x}_0 = \boldsymbol{x}(0, \boldsymbol{x}_0)$ to the training points. [1]

To solve the problem of precise tracking of human motion, we generate trajectories from the model $\hat{\boldsymbol{f}}(\boldsymbol{x})$ in (3). $\hat{\boldsymbol{f}}(\boldsymbol{x})$ is defined as the gradient descent on the proposed control Lyapunov function (CLF). This CLF is obtained by simulating mean trajectories from an asymptotically stable dynamic model $\bar{\boldsymbol{f}}(\boldsymbol{x})$, which is an GPSSM trained from the data set $\mathcal{D}$, and integrating along them over the model uncertainty. In the following, we start with a review on Gaussian processes as they build the core of the identification of the stable dynamic model $\bar{\boldsymbol{f}}(\boldsymbol{x})$.

3. LEARNING STABLE GAUSSIAN PROCESS STATE SPACE MODELS

3.1 Gaussian Processes Regression

A GP is defined as a collection of random variables 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 (Rasmussen and Williams, 2006). It is often considered as a distribution over functions with interference between them and denoted by

$$f_{\boldsymbol{\psi}}(\boldsymbol{x}) \sim \mathcal{GP}(m_{\boldsymbol{\psi}}(\boldsymbol{x}), k_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{x}')). \quad (4)$$

It is specified by a mean function $m_{\boldsymbol{\psi}}(\boldsymbol{x}) \colon \mathcal{X} \to \mathbb{R}$ and a covariance function $k_{\boldsymbol{\psi}}(\boldsymbol{x}, \boldsymbol{x}') \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The subscript $\boldsymbol{\psi}$ denotes the dependency on hyperparameters which characterize the set of functions over which the GP is a distribution. A widely used covariance function is the squared exponential (SE) kernel

$$k_{\boldsymbol{\psi}}^{\text{SE}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\left(\sum_{j=1}^{n} \frac{(x_j - x_j')^2}{-2 l_j^2} \right), \quad (5)$$

which results in a distribution over bounded, infinitely differentiable functions. The hyperparameters of the SE kernel are the lengthscales $l_j \in \mathbb{R}_+$, $j = 1, \ldots, n$. To inject the prior knowledge, that the system (2) has a non-zero

[1] We do not consider *tracking* in the classical sense, since we focus on geometric paths rather than time-parametrized trajectories. Accordingly, by *precise tracking*, we indicate a high spacial similarity between the generated and the demonstrated path, which is for most tasks more important than moving at the proper speed.

velocity everywhere outside the origin (otherwise it would not be globally asymptotically stable), we set a linear prior

$$m(\boldsymbol{x}) = \boldsymbol{\alpha}^\intercal \boldsymbol{x} \tag{6}$$

with $\boldsymbol{\alpha} \in \mathbb{R}^n$, resulting in a hyperparameter vector $\boldsymbol{\psi} = \begin{bmatrix}\alpha_1 & \dots & \alpha_n & l_1 & \cdots & l_n & \sigma_f^2\end{bmatrix}^\intercal$.

Since (4) represents only functions with scalar outputs, n independent GPs are employed to model the dynamical system (2), denoted by

$$\boldsymbol{f}_{\boldsymbol{\Psi}}(\boldsymbol{x}) = \begin{cases} f_{\psi_1}(\boldsymbol{x}) \sim \mathcal{GP}\left(m_1(\boldsymbol{x}), k_1(\boldsymbol{x}, \boldsymbol{x}')\right) \\ \vdots \qquad\qquad \vdots \\ f_{\psi_n}(\boldsymbol{x}) \sim \mathcal{GP}\left(m_n(\boldsymbol{x}), k_n(\boldsymbol{x}, \boldsymbol{x}')\right), \end{cases} \tag{7}$$

where $\boldsymbol{\Psi}$ is the set of all $\boldsymbol{\psi}_j$ with $j = 1, \dots, n$, which are hyperparameters corresponding to mean and kernel functions $k_j(\cdot,\cdot)$ and $m_j(\cdot)$, respectively.

Considering a dataset $\mathcal{D}$ as in (1), the Gaussian process is often employed for regression. Given a test input $\boldsymbol{x}^*$, the j-th component of the inferred output $\boldsymbol{y}^*$ is jointly Gaussian distributed with the training data

$$\begin{bmatrix} y_j^* \\ \boldsymbol{y}_j \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m_j(\boldsymbol{x}^*) \\ \boldsymbol{m}_j \end{bmatrix}, \begin{bmatrix} k_j^* & \boldsymbol{k}_j^\intercal \\ \boldsymbol{k}_j & \boldsymbol{K}_j + \sigma_{\text{on}}^2 \end{bmatrix}\right), \tag{8}$$

where $k_j^* = k_j(\boldsymbol{x}^*, \boldsymbol{x}^*) \in \mathbb{R}$, $\boldsymbol{y}_j = \begin{bmatrix} y_j^{(1)} \cdots y_j^{(N)} \end{bmatrix}^\intercal \in \mathbb{R}^N$,

$$\boldsymbol{k}_j = \begin{bmatrix} k_j\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^*\right) & \cdots & k_j\left(\boldsymbol{x}^{(N)}, \boldsymbol{x}^*\right) \end{bmatrix}^\intercal \in \mathbb{R}^N,$$

$$\boldsymbol{m}_j = \begin{bmatrix} m_j\left(\boldsymbol{x}^{(1)}\right) & \cdots & m_j\left(\boldsymbol{x}^{(N)}\right) \end{bmatrix}^\intercal \in \mathbb{R}^N$$

and

$$\boldsymbol{K}_j = \begin{bmatrix} k_j\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(1)}\right) & \cdots & k_j\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(N)}\right) \\ \vdots & \ddots & \vdots \\ k_j\left(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(1)}\right) & \cdots & k_j\left(\boldsymbol{x}^{(N)}, \boldsymbol{x}^{(N)}\right) \end{bmatrix} \in \mathbb{R}^{N \times N}.$$

Conditioning on test input $\boldsymbol{x}^*$ and the data $\mathcal{D}$ yields

$$\mathbb{E}[y_j^* | \mathcal{D}, \boldsymbol{x}^*] = m_j(\boldsymbol{x}^*) + \boldsymbol{k}_j^\intercal (\boldsymbol{K}_j + \sigma_{\text{on}}^2 \boldsymbol{I}_N)^{-1} (\boldsymbol{y}_j - \boldsymbol{m}_j) \tag{9}$$

$$\text{var}[y_j^* | \mathcal{D}, \boldsymbol{x}^*] = k_j^* - \boldsymbol{k}_j^\intercal (\boldsymbol{K}_j + \sigma_{\text{on}}^2 \boldsymbol{I}_N)^{-1} \boldsymbol{k}_j. \tag{10}$$

To obtain the optimal hyperparameters $\boldsymbol{\psi}_j$, a likelihood maximization according to Bayesian principles is performed for every $j = 1, \dots, n$ as

$$\max_{\boldsymbol{\psi}_j} \log p\left(\boldsymbol{y}_j | \boldsymbol{x}^{(1:N)}, \boldsymbol{\psi}_j\right) = \tag{11}$$

$$\max_{\boldsymbol{\psi}_j} \left(-\frac{1}{2} \boldsymbol{y}_j^T \boldsymbol{K}_j^{-1} \boldsymbol{y}_j - \frac{1}{2} \log\det \boldsymbol{K}_j - \frac{N}{2} \log(2\pi)\right),$$

where $\boldsymbol{x}^{(1:N)}$ denotes the set of all $\boldsymbol{x}^{(i)}$, $i = 1, \dots, N$. Even though this optimization problem is non-convex it is usually solved with gradient-based methods because local minima correspond to a different interpretation of the data and still deliver a conclusive prediction.

3.2 Enforcing an Equilibrium Point

The Gaussian process as presented has been used to model dynamical systems, see e.g. (Kocijan, 2016). However, we impose a minor modification: As the equilibrium point at $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$ is known from Assumption 2, we extend (8) as follows

$$\begin{bmatrix} y_j^* \\ \boldsymbol{y}_j \\ 0 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \boldsymbol{0} \\ 0 \end{bmatrix}, \begin{bmatrix} k_j^* & \boldsymbol{k}_j^\intercal & k_j(\boldsymbol{x}^*, \boldsymbol{0}) \\ \boldsymbol{k}_j & \boldsymbol{K}_j + \sigma_{\text{on}}^2 & \boldsymbol{k}_j^{\boldsymbol{0}} \\ k_j(\boldsymbol{0}, \boldsymbol{x}^*) & \boldsymbol{k}_j^{\boldsymbol{0}\intercal} & k_j(\boldsymbol{0}, \boldsymbol{0}) \end{bmatrix}\right),$$

where

$$\boldsymbol{k}_j^{\boldsymbol{0}} = \begin{bmatrix} k_j\left(\boldsymbol{x}^{(1)}, \boldsymbol{0}\right) & \cdots & k_j\left(\boldsymbol{x}^{(N)}, \boldsymbol{0}\right) \end{bmatrix}^\intercal \in \mathbb{R}^N.$$

Note that for the training point $(\boldsymbol{0}, 0)$, there is no observation noise, since it is known a priori without uncertainty and thus not measured. As a result, every function described by the GP passes through the origin. For the mean and variance predictions, we define

$$\mu_j(\boldsymbol{x}^*) \coloneqq \mathbb{E}[y_j^* | \mathcal{D}, \boldsymbol{x}^*, (\boldsymbol{0}, 0)], \quad j = 1, \dots, n \tag{12}$$

$$\sigma_j^2(\boldsymbol{x}^*) \coloneqq \text{var}[y_j^* | \mathcal{D}, \boldsymbol{x}^*, (\boldsymbol{0}, 0)], \quad j = 1, \dots, n \tag{13}$$

equivalently to (9) and (10). For their concatenation

$$\boldsymbol{\mu}(\boldsymbol{x}) = \begin{bmatrix} \mu_1(\boldsymbol{x}) & \cdots & \mu_n(\boldsymbol{x}) \end{bmatrix}^\intercal, \tag{14}$$

$$\boldsymbol{\sigma}^2(\boldsymbol{x}) = \begin{bmatrix} \sigma_1^2(\boldsymbol{x}) & \cdots & \sigma_n^2(\boldsymbol{x}) \end{bmatrix}^\intercal, \tag{15}$$

the following is concluded:

Lemma 1. The posterior variance function $\boldsymbol{\sigma}^2 \colon \mathcal{X} \to \mathbb{R}_{+,0}^n$ from (15) from a GP with SE kernel (5) is bounded, infinitely differentiable and componentwise positive definite.

Proof. The variance function $\sigma^2(\boldsymbol{x})$ (index j omitted for notational simplicity) of a GP with SE kernel (5) is bounded by $0 \leq \sigma^2(\boldsymbol{x}) \leq \sigma_f^2$ and infinitely differentiable because it inherits these properties from the kernel. According to Umlauft et al. (2017a), if $\sigma_N^2(\boldsymbol{x})$ and $\sigma_{N+1}^2(\boldsymbol{x})$ denote variance functions for N and $N+1$ training data points respectively, then

$$\sigma_N^2(\boldsymbol{x}) \geq \sigma_{N+1}^2(\boldsymbol{x}), \qquad \forall \boldsymbol{x} \in \mathcal{X}. \tag{16}$$

Considering only the training data point $(\boldsymbol{0}, 0)$, $\sigma^2(\boldsymbol{0}) = 0$ holds, as can be directly seen from (13). With N training data points it results from non-negativity and (16) that $\sigma_N^2(\boldsymbol{0}) = 0$ holds. For all $\boldsymbol{x} \in \mathcal{X} \setminus \{\boldsymbol{0}\}$, we can conclude that $\sigma^2(\boldsymbol{x}) > 0$ because of the measurement noise $\sigma_{\text{on}}^2 > 0$ and finite training data points. ■

Fig. 1 illustrates the properties of the posterior variance function of a GP as defined in (13). The GP variance generally grows with distance from training data. Therefore, it can be interpreted as a suitable proximity measure.

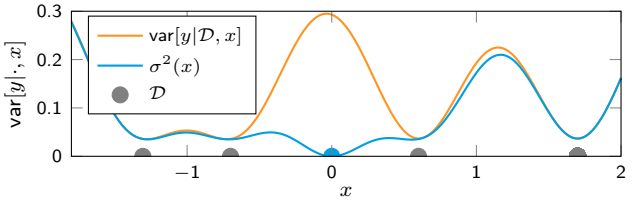

Fig. 1. The orange line visualizes the GP variance $\text{var}[y|\mathcal{D}, x]$ from (10) for $N = 4$ training points. The blue line visualizes $\sigma^2(x)$ from (13), with the enforced equilibrium point at $x = 0$ as $(0, 0)$. The variance reduces globally according to (16), thus $\sigma^2(x) \leq \text{var}[y|\mathcal{D}, x]$, $\forall x \in \mathcal{X}$. The positive definiteness of $\sigma^2(x)$ from Lemma 1 is also observable.

3.3 Stabilization of the System

For the GPSSM we have ensured an equilibrium point at the origin, $\boldsymbol{\mu}(\boldsymbol{0}) = \boldsymbol{0}$, but it might be only locally attractive or even unstable. Therefore, we follow the approach proposed by Umlauft et al. (2017b) and Khansari-Zadeh

and Billard (2014) to stabilize the GPSSM by applying a corrective signal $\boldsymbol{u} \in \mathbb{R}^n$, given by

$$\dot{\boldsymbol{x}} = \bar{\boldsymbol{f}}(\boldsymbol{x}) \coloneqq \boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{u}(\boldsymbol{x}). \tag{17}$$

Note, that $\bar{\boldsymbol{f}}(\boldsymbol{x})$ is not the final proposed model $\hat{\boldsymbol{f}}(\boldsymbol{x})$ but only an intermediate step. We will refer to it as the *stabilized GPSSM* and the following is assumed.

Assumption 3. There exists $\boldsymbol{u}(\boldsymbol{x})$ such that, the system (17) is uniformly globally asymptotically stable and $\bar{\boldsymbol{f}} : \mathcal{X} \to \mathbb{R}^n$ is locally Lipschitz $\forall \boldsymbol{x} \in \mathcal{X}$.

The existence of a corrective signal is shown by Khansari-Zadeh and Billard (2014) and chosen to only minimally change $\boldsymbol{\mu}(\boldsymbol{x})$. It is derived based on a stabilizing Lyapunov function, which is adapted for the data $\mathcal{D}$. Umlauft et al. (2017b) propose the use of sum of squares (SOS) as stabilizing Lyapunov function $V_{\boldsymbol{Q}} : \mathcal{X} \to \mathbb{R}_{+,0}$, defined as

$$V_{\boldsymbol{Q}} = \boldsymbol{m}(\boldsymbol{x})^\intercal \boldsymbol{Q} \boldsymbol{m}(\boldsymbol{x}), \tag{18}$$

where $\boldsymbol{Q} \succ 0$ and $\boldsymbol{m}(\boldsymbol{x})$ are monomials of a user defined degree, see also Papachristodoulou and Prajna (2005) for details on the SOS technique. The entries of $\boldsymbol{Q}$ are chosen such that the original GPSSM $\boldsymbol{\mu}(\boldsymbol{x})$ is only minimally adapted, thus $\|\boldsymbol{u}(\boldsymbol{x})\|$ is minimal. This is achieved by minimizing the violations of $\frac{\partial V_{\boldsymbol{Q}}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial t} < 0$ for $\mathcal{D}$. Since the measurements are noisy and only a subset of all positive definite functions can be described by the selected parametric form, this might not be fulfilled for all data points. Nevertheless, it is suitable for the stabilization.

For the computation of the corrective signal $\boldsymbol{u} \in \mathbb{R}^n$, we impose a continuous positive definite function $\rho(\|\boldsymbol{x}\|)$ as lowest acceptable rate of decrease for the Lyapunov function $V_{\boldsymbol{Q}}$. The minimal corrective signal is

$$\boldsymbol{u}^* = \underset{\boldsymbol{u}}{\operatorname{argmin}} \frac{1}{2} \boldsymbol{u}^\intercal \boldsymbol{u}, \tag{19a}$$

$$\text{s.t. } \Delta_{\boldsymbol{x}} V_{\boldsymbol{Q}}(\boldsymbol{x}) \left(\boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{u}\right) \leq -\rho(\|\boldsymbol{x}\|) \quad \text{if } \boldsymbol{x} \neq \boldsymbol{0} \qquad \boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{u} = 0 \quad \text{if } \boldsymbol{x} = \boldsymbol{0}. \tag{19b}$$

The solution to this problem for the Lyapunov function is

$$\boldsymbol{u}^*(\boldsymbol{x}) = \begin{cases} \boldsymbol{0} & \text{if } \Delta_{\boldsymbol{x}} V_{\boldsymbol{Q}}(\boldsymbol{x}) \boldsymbol{\mu}(\boldsymbol{x}) + \rho(\|\boldsymbol{x}\|) \leq 0 \\ \frac{\Delta_{\boldsymbol{x}} V_{\boldsymbol{Q}}(\boldsymbol{x}) \boldsymbol{\mu}(\boldsymbol{x}) + \rho(\|\boldsymbol{x}\|)}{-\|\Delta_{\boldsymbol{x}} V_{\boldsymbol{Q}}(\boldsymbol{x})\|^2} \Delta_{\boldsymbol{x}}^\intercal V_{\boldsymbol{Q}}(\boldsymbol{x}) & \text{otherwise.} \end{cases}$$

Remark 1. Compared to previous work (Kocijan, 2016; Umlauft et al., 2017b; Khansari-Zadeh and Billard, 2014), we introduced two modifications of GPSSMs, a linear prior and an enforced equilibrium point to ensure positive definiteness of the variance function. This is necessary to achieve a stable system as shown in the next section.

4. PATH TRACKING

In Section 3, we outlined how GPSSMs are applied to learn a stable model for the true dynamics (2) which fulfill Assumptions 1 and 2. However, the stabilized GPSSM $\bar{\boldsymbol{f}}(\boldsymbol{x})$ (17) does not take advantage of the GP uncertainty and therefore trajectories $\bar{\boldsymbol{x}}$ generated from it do not necessarily approach the demonstration data. In this section, we present how to utilize knowledge about model fidelity to introduce the uncertainty-aware system $\hat{\boldsymbol{f}}(\boldsymbol{x})$.

4.1 Uncertainty-based Control Lyapunov Function

Given that the stabilized GPSSM $\bar{\boldsymbol{f}}(\boldsymbol{x})$ is Lipschitz continuous by Assumption 3, there exists for each $\boldsymbol{x}_0 \in \mathcal{X}$ a unique solution to the differential equation (17) denoted by $\bar{\boldsymbol{x}}(t, \boldsymbol{x}_0) : [0,\ \infty] \times \mathcal{X} \to \mathcal{X}$ according to Khalil and Grizzle (1996). Together with its uniform global asymptotic stability holds

$$\bar{\boldsymbol{x}}(0, \boldsymbol{x}_0) = \boldsymbol{x}_0, \quad \lim_{t \to \infty} \bar{\boldsymbol{x}}(t, \boldsymbol{x}_0) = \boldsymbol{0}, \quad \forall \boldsymbol{x}_0 \in \mathcal{X}. \tag{20}$$

As already illustrated in Fig. 1, the posterior variance function of the GP represents model fidelity based on the location of the training points. Because it can be interpreted as a measure for proximity to the demonstrations, it seems reasonable to reduce the norm of $\boldsymbol{\sigma}^2(\boldsymbol{x})$ along the path. We therefore consider the proximity of a path to the demonstrations as accumulated model uncertainty along the trajectory of the stabilized GPSSM, written as path integral

$$V_{\boldsymbol{\sigma}}(\boldsymbol{x}_0) = \int_0^\infty \left\|\boldsymbol{\sigma}^2\left(\bar{\boldsymbol{x}}(t, \boldsymbol{x}_0)\right)\right\| \left\|\dot{\bar{\boldsymbol{x}}}(t, \boldsymbol{x}_0)\right\| dt. \tag{21}$$

Besides the interpretation as a cost-to-go term of the path $\bar{\boldsymbol{x}}(t, \boldsymbol{x}_0)$, it can also be seen as Lyapunov candidate.

Theorem 1. Consider the solutions $\bar{\boldsymbol{x}}(t, \boldsymbol{x}_0)$ for the system (17) under Assumption 3 and the GP posterior variance function $\boldsymbol{\sigma}^2(\boldsymbol{x})$ in (15) with SE kernel (5). Then, the function $V_{\boldsymbol{\sigma}}(\boldsymbol{x}_0) : \mathcal{X} \to \mathbb{R}$, defined in (21) is a Lyapunov candidate, thus positive definite and differentiable. Additionally, it is radially unbounded.

Proof. Since trajectories starting in the origin never leave it due to $\bar{\boldsymbol{f}}(\boldsymbol{0}) = \boldsymbol{0}$ (Assumption 3), we can conclude $V_{\boldsymbol{\sigma}}(\boldsymbol{0}) = 0$. Since σ_j^2 is positive definite by Lemma 1, any trajectory starting outside of the origin leads to a positive integral, thus $V_{\boldsymbol{\sigma}}(\boldsymbol{x}_0) > 0$ for all $\boldsymbol{x}_0 \in \mathcal{X} \setminus \{\boldsymbol{0}\}$, which concludes the positive definiteness. To show the differentiability of $V_{\boldsymbol{\sigma}}(\boldsymbol{x})$, consider the Leibniz integral rule ($t, \boldsymbol{x}_0$ omitted for brevity)

$$\frac{\partial}{\partial \boldsymbol{x}} \int_0^\infty \left\|\boldsymbol{\sigma}^2\left(\bar{\boldsymbol{x}}\right)\right\| \left\|\dot{\bar{\boldsymbol{x}}}\right\| dt = \int_0^\infty \frac{\partial}{\partial \boldsymbol{x}} \left\|\boldsymbol{\sigma}^2\left(\bar{\boldsymbol{x}}\right)\right\| \left\|\dot{\bar{\boldsymbol{x}}}\right\| dt.$$

The condition that the integrand is differentiable in t is fulfilled in our case for the following reasoning: The solution of the differential equation (17) given by $\bar{\boldsymbol{x}}(t, \boldsymbol{x}_0)$ is Lipschitz according to the continuity of solutions of differential equations with their initial conditions (Khalil and Grizzle, 1996, Theorem 3.4). The composite of the Lipschitz function $\bar{\boldsymbol{x}}(t, \boldsymbol{x}_0)$ and the differentiable function $\sigma_j(\cdot)$, $j = 1, \ldots, n$ is differentiable and therefore, the integrand is differentiable in t. As a consequence of the Leibniz integral rule, the function $V_{\boldsymbol{\sigma}}(\boldsymbol{x})$ is differentiable. Due to the observation noise for all training points outside the origin, $\exists r > 0$ such that $\forall \|\boldsymbol{x}\| > r$ holds $\sigma_j^2(\boldsymbol{x}) \geq \sigma_{\text{on}}^2$, $\forall j = 1, \ldots, n$. Thus, for $\|\boldsymbol{x}_0\| \to \infty$ the path integral is unbounded $V_{\boldsymbol{\sigma}}(\boldsymbol{x}_0) \to \infty$ because the integrand is lower bounded by a constant for $\|\boldsymbol{x}_0\| > r$ and the length of the path over which is integrated is unbounded. Thus, $V_{\boldsymbol{\sigma}}$ is radially unbounded. ■

Remark 2. Note, that the Lyapunov function $V_{\boldsymbol{Q}}$ is not related to the proposed uncertainty Lyapunov function $V_{\boldsymbol{\sigma}}$. It is only required to obtain the stabilizing control $\boldsymbol{u}$ which was, in the theoretical part, given by Assumption 3.

4.2 Stable Path Tracking System

The function $V_{\boldsymbol{\sigma}}(\boldsymbol{x})$ is not just a Lyapunov candidate but also encodes the accumulated model uncertainty along

the trajectories of the stabilized GPSSM. We therefore propose the following gradient descent on $V_{\boldsymbol{\sigma}}(\boldsymbol{x})$

$$\dot{\boldsymbol{x}} = \hat{\boldsymbol{f}}(\boldsymbol{x}) \coloneqq -k_c \Delta_{\boldsymbol{x}}^{\intercal} V_{\boldsymbol{\sigma}}(\boldsymbol{x}), \tag{22}$$

where $k_c > 0$ scales the speed of system. By descending $V_{\boldsymbol{\sigma}}(\boldsymbol{x})$, we expect, that the generated trajectories are attracted by regions of the state space which are close to the demonstrations and converge to the origin along the trajectories of system (17). The following is concluded

Theorem 2. Consider the uncertainty Lyapunov function $V_{\boldsymbol{\sigma}}(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}_+$ defined in (21), where the solutions $\bar{\boldsymbol{x}}(t, \boldsymbol{x}_0)$ are obtained from system (17) under Assumption 3 and $\boldsymbol{\sigma}^2(\boldsymbol{x})$ is the GP posterior variance function in (15) with SE kernel (5). Then, the dynamical system obtained from its gradient descent in (22) is uniformly globally asymptotically stable for any $k_c > 0$.

Proof. The Lyapunov candidate $V_{\boldsymbol{\sigma}}(\boldsymbol{x})$ fulfills the required properties by Theorem 1. Its time derivative

$$\begin{aligned}\dot{V}_{\boldsymbol{\sigma}}(\boldsymbol{x}) &= \Delta_{\boldsymbol{x}} V_{\boldsymbol{\sigma}}(\boldsymbol{x}) \dot{\boldsymbol{x}} = -k_c \Delta_{\boldsymbol{x}} V_{\boldsymbol{\sigma}}(\boldsymbol{x}) \Delta_{\boldsymbol{x}}^{\intercal} V_{\boldsymbol{\sigma}}(\boldsymbol{x}) \\ &= -k_c \left\| \Delta_{\boldsymbol{x}} V_{\boldsymbol{\sigma}}(\boldsymbol{x}) \right\|^2\end{aligned}$$

is negative definite, shown as follows: $\|\Delta_{\boldsymbol{x}} V_{\boldsymbol{\sigma}}(\boldsymbol{0})\| = 0$ holds due to the differentiability and the positive definiteness of $V_{\boldsymbol{\sigma}}$. To show $\|\Delta_{\boldsymbol{x}} V_{\boldsymbol{\sigma}}(\boldsymbol{x})\| > 0$ holds $\forall \boldsymbol{x} \in \mathcal{X} \setminus \{\boldsymbol{0}\}$, we consider the Lie derivative $\mathcal{L}_{\hat{\boldsymbol{f}}} V_{\boldsymbol{\sigma}}(\boldsymbol{x})$. It is strictly negative $\forall \boldsymbol{x} \in \mathcal{X} \setminus \{\boldsymbol{0}\}$ because the length of the integrated path decreases in this direction and the integrand is strictly positive (Lemma 1) outside the origin. A non-zero Lie derivative leads to a non-zero norm of the gradient, which shows the negative definiteness of $\dot{V}_{\boldsymbol{\sigma}}(\boldsymbol{x})$. Since $V_{\boldsymbol{\sigma}}(\boldsymbol{x})$ is radially unbounded by Theorem 1, the uniform asymptotic stability holds globally. ■

Remark 3. To some extent our approach is related to an optimal control problem. The Lyapunov function (21) accumulates the uncertainty along the GPSSM path. It can therefore be interpreted as a cost-to-go function. However, we do not compute the value function, for which (22) would solve the optimal control problem (minimizing the accumulated variance) for three reasons: First, it would require to solve the Hamilton-Jacobi-Bellman equation for which in a continuous state space only approximate solutions exist in the general case. Second, the resulting value function is generally not differentiable, a required property for the further design steps. Third, finding an (approximatively) optimal solution is computationally very expensive, especially in high-dimensional state spaces. Our solution requires a forward simulation to obtain $\bar{\boldsymbol{x}}(t, \boldsymbol{x}_0)$, but other than (for the optimal solution) exploring all possible directions, we only search along the GPSSM mean direction (a good approximation if the demonstrations converge). This keeps our approach computationally tractable.

5. EVALUATION

Before demonstrating the approach numerically in a human path tracking setting, we address a few points regarding the implementation including the stabilization of the GPSSM, the calculation of the uncertainty-based Lyapunov function and the performance measure. For convenience, we label our approach with UCLD for **U**ncertainty-based **C**ontrol **L**yapunov function **D**escent.

5.1 Implementation

Stabilization of GPSSMs The trajectories $\bar{\boldsymbol{x}}$ are obtained from a forward simulation of the stabilized estimate $\bar{\boldsymbol{f}}$ from (17) as shown in Section 3.3. As stabilizing Lyapunov function we use SOS as defined in (18) with a minimal acceptable rate of decrease ρ of the form

$$\rho(\|\boldsymbol{x}\|) = \min(\sin(\rho_0), \|\boldsymbol{x}\|^2).$$

This allows the interpretation of $\rho_0 \in (0, \pi]$ as the angle between the direction of the stabilized system and the tangent of the Lyapunov level line. We chose $\rho_0 = 5°$.

Computation of the uncertainty-based Lyapunov function The most critical part of the implementation is the computation of the uncertainty-based Lyapunov function $V_{\boldsymbol{\sigma}}$ because the path over which it integrates, $\bar{\boldsymbol{x}}(t, \boldsymbol{x}_0)$, is not known analytically. However, through simulation, we find arbitrary close approximations as shown in the following: We obtain a finite number of consecutive states $\bar{\boldsymbol{x}}_0, \dots, \bar{\boldsymbol{x}}_M$ through forward simulation of (17), which approximate the solution $\bar{\boldsymbol{x}}(t, \boldsymbol{x}_0)$. The space between the discretization is approximated by straight lines and we define the line integral between two points $\bar{\boldsymbol{x}}_m$ and $\bar{\boldsymbol{x}}_{m+1}$ over the variance $\sigma_j(\bar{\boldsymbol{x}})$ for all dimensions $j = 1, \dots, n$ as $L_j(\bar{\boldsymbol{x}}_m, \bar{\boldsymbol{x}}_{m+1}) \in \mathbb{R}^+$. The vector $\boldsymbol{L}(\bar{\boldsymbol{x}}_m, \bar{\boldsymbol{x}}_{m+1}) = [L_1(\bar{\boldsymbol{x}}_m, \bar{\boldsymbol{x}}_{m+1}) \cdots L_n(\bar{\boldsymbol{x}}_m, \bar{\boldsymbol{x}}_{m+1})]^{\intercal}$ concatenates the line integrals for all dimensions and we can estimate the integral (21) by taking the norm of $\boldsymbol{L}(\boldsymbol{x}_m, \boldsymbol{x}_{m+1})$ for all points $\boldsymbol{x}_0, \dots, \boldsymbol{x}_M$ and sum them together as given by

$$\bar{V}_{\boldsymbol{\sigma}}(\boldsymbol{x}) = \sum_{m=0}^{M} \|\boldsymbol{L}(\boldsymbol{x}_m, \boldsymbol{x}_{m+1})\| \approx V_{\boldsymbol{\sigma}}(\boldsymbol{x}). \tag{23}$$

The line integrals L_j can be computed analytically for the squared exponential kernel as shown in Appendix A.

Although $\bar{V}_{\boldsymbol{\sigma}}$ is only an approximation of $V_{\boldsymbol{\sigma}}$, it still fulfills the positive definiteness and radial unboundedness properties. In order to apply $\bar{V}_{\boldsymbol{\sigma}}$ as the control Lyapunov function, its gradient is evaluated by first order finite differences (with a finite difference $\epsilon_{fd} = 10^{-4}$). The forward simulation is performed with $\Delta t = 0.1s$.

Measuring tracking precision We consider the area error to evaluate tracking precision. It is defined for N_{demo} generated trajectories $\boldsymbol{x}_p^{(k)}$, with $k = 1, \dots, N_{\text{sim}}$ and $p = 1, \dots, N_{\text{demo}}$ as

$$\Delta_{\text{rep}} \coloneqq \frac{1}{N_{\text{demo}}} \sum_{p=1}^{N_{\text{demo}}} \sum_{k=1}^{N_{\text{sim}}} A(\boldsymbol{x}_p^{(k)}, \boldsymbol{x}_p^{(k+1)}, \boldsymbol{x}_{p'}^{(i)}, \boldsymbol{x}_{p'}^{(i+1)}), \tag{24}$$

with $\boldsymbol{x}_{p'}^{(i)}$ as the closest training point regarding Euclidean distance to $\boldsymbol{x}_p^{(k)}$ and $\mathrm{A}(\xi^1, \xi^2, \xi^3, \xi^4)$ as area of a tetragon of the points ξ^1 to ξ^4.

5.2 Setup

For the numerical evaluation, we demonstrate the effectiveness of UCLD approach in a path tracking task of demonstrated human goal-directed movement, a robot is supposed to reproduce. We compare our UCLD [2] with two frameworks that do not consider model uncertainties:

[2] Available at *https://gitlab.lrz.de/ga68car/cphs2018ucld.git*

- The stable estimator of dynamical systems (SEDS) from Khansari-Zadeh and Billard (2011), constrains the learning of a Gaussian Mixture Model to ensure stability of its dynamics. We denote this system in the following as $\dot{\boldsymbol{x}}_{\text{SEDS}} = \boldsymbol{f}_{\text{SEDS}}(\boldsymbol{x}_{\text{SEDS}})$.
- Taking directly the stabilized GPSSM as already introduced in Section 3.3 as $\dot{\bar{\boldsymbol{x}}} = \bar{\boldsymbol{f}}(\bar{\boldsymbol{x}})$, which was proposed in Umlauft et al. (2017b).

We demonstrate that the proposed method UCLD, precisely the dynamical system $\dot{\boldsymbol{x}} = \hat{\boldsymbol{f}}(\boldsymbol{x})$ defined in (22), results in precise reproduction of human motions while providing the necessary convergence guarantees. For validation, we employ the LASA handwriting dataset [3] consisting of 24 movements, demonstrated by a human in two dimensions with varying number of repetitions and 150 or 250 data points each and sampled with $\Delta t = 0.1s$. The goal is to reproduce the motion as precise as possible and we apply the area error from (24) as a precision measure. For the reproduction, one initial state is randomly sampled in a ball with radius $r = 10$ around each starting point of the training trajectories. The simulation is stopped when the terminal condition $\|\boldsymbol{x}\| < 10$ is reached and $k_c = 1$.

5.3 Results

For the *Sharp-C* of the LASA dataset, the training data $\mathcal{D}$, the stabilized GPSSM $\bar{\boldsymbol{f}}$ and the model uncertainty $\|\boldsymbol{\sigma}^2(\bar{\boldsymbol{x}})\|$ are shown in Fig. 2. By integrating shown trajectories over $\|\boldsymbol{\sigma}^2(\bar{\boldsymbol{x}})\|$, $\bar{V}_{\boldsymbol{\sigma}}(\boldsymbol{x})$ (23) is calculated. The resulting uncertainty-based control Lyapunov function is depicted in Fig. 3. It also shows the trajectories generated by SEDS $\boldsymbol{x}_{\text{SEDS}}(t)$, the stabilized GPSSM without uncertainty awareness $\bar{\boldsymbol{x}}(t)$ and our proposed approach UCLD $\boldsymbol{x}(t)$. Table 1 shows the quantitative comparison of the tracking precision defined in (24) for all these approaches for the *Sharp-C* movement and the full LASA dataset of 24 motions. The uncertainty-based control Lyapunov function outperforms the stabilized GPSSM and SEDS methods numerically in terms of tracking precision. However, the additional calculation steps result in longer average training and reproduction times.

Table 1. Reproduction error Δ_{rep} and average times for training and reproduction in seconds.

	UCLD $\hat{\boldsymbol{f}}$	$\bar{\boldsymbol{f}}$	SEDS $\boldsymbol{f}_{\text{SEDS}}$
Δ_{rep} Sharp-C	725.8	1553	5006
Δ_{rep} all 24 motions	1976	3787	3533
$\bar{t}_{\text{train}}$ all 24 motions	208s	116s	24.0s
$\bar{t}_{\text{rep}}$ all 24 motions	2.12s	0.81s	0.07s

5.4 Discussion

With the numerical example, we demonstrate that the UCLD approach is well suited for precise tracking of human motion in a learning by demonstration setting. Fig. 3 makes clear that previous approaches (like SEDS), which do not consider model uncertainty, suffer from imprecise generalization: Once the path has left the proximity of the training data, there is no incentive to return to a region where the desired behavior is known. Thus, the generalization is not able to recover once it has left the trained region or if starting from a point not in the training set. In contrast, the proposed method UCLD has incorporated a mechanism to recover in these cases because the gradient descent on the uncertainty-based control Lyapunov function leads to regions containing trajectories with low uncertainty and more demonstrations towards the goal. Our evaluation on real-world data suggests that this advantage improves the tracking precision in the reproduction of demonstrated movements. Compared to SEDS, the stabilized GPSSM and UCLD approaches employ with GPs a non-parametric model for the dynamics which allow to represent arbitrary complex motions. The comparison of stabilized GPSSM and UCLD demonstrates the advantage of uncertainty-based path tracking towards higher tracking precision. Also important to note is that the approach works parameter free (except for k_c, which is here only a time scaling factor), thus no manual tuning of a risk-averseness or similar is necessary. However, the required

[3] Available at *https://bitbucket.org/khansari/seds*

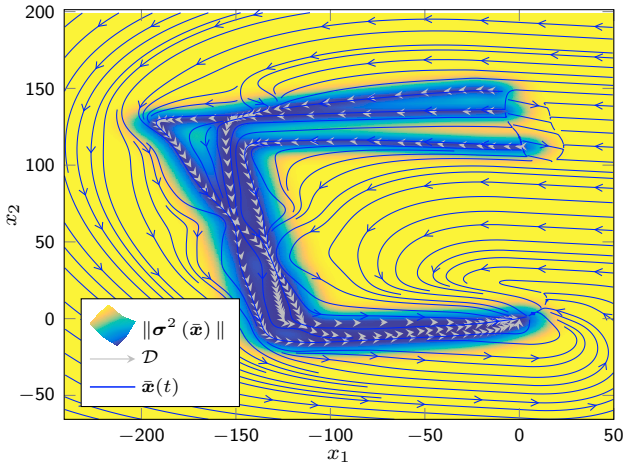

Fig. 2. The training data (gray arrows), the model uncertainty $\left\|\boldsymbol{\sigma}^2(\bar{\boldsymbol{x}})\right\|$ of the GPs (colormap) and the stable trajectories $\bar{\boldsymbol{x}}(t)$ (blue streamlines), generated by $\bar{\boldsymbol{f}}$, for the LASA *Sharp-C* movement are shown.

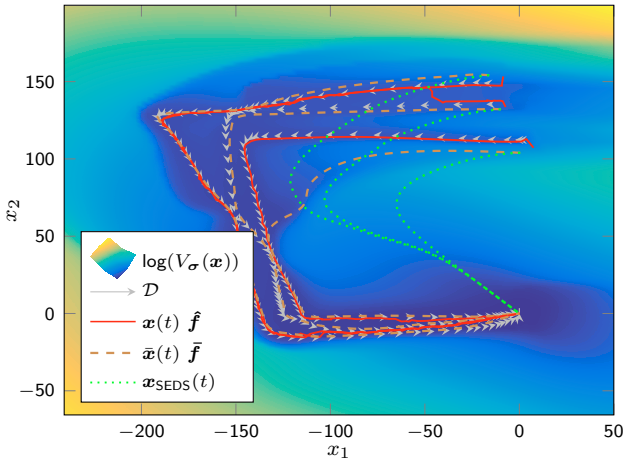

Fig. 3. The logarithm of the uncertainty-based control Lyapunov function $V_{\boldsymbol{\sigma}}(\boldsymbol{x}_k)$ (colormap) defined in (21) with the training data (gray arrows) from the LASA *Sharp-C* movement are shown. The generated paths from 3 initial states for UCLD $\boldsymbol{x}(t)$, generated by $\hat{\boldsymbol{f}}$, stabilized GPSSM $\bar{\boldsymbol{x}}(t)$, generated by $\bar{\boldsymbol{f}}$, and SEDS $\boldsymbol{x}_{\text{SEDS}}(t)$, generated by $\boldsymbol{f}_{\text{SEDS}}$, are given by red, dashed brown and dotted green lines, respectively.

forward simulation depends on Δt. Higher values of Δt make the approach faster but lead to numerical difficulties. In our implementation, we a priori compute the Lyapunov function on a discrete grid and interpolate it for the gradient descent. The approach aims to reduce the distance to training data (which can also be formulated in an optimal control problem) being computational tractable for high dimensions of the state space. However, our approach does not provide any optimality guarantees regarding a particular cost function as pointed out in Remark 3.

6. CONCLUSION

The main contribution of this paper is an uncertainty-aware control Lyapunov function approach for human motion tracking based on Gaussian process state space models. By integrating over the variance function of a Gaussian process along the stabilized mean trajectories, it takes the model uncertainty into account. It therefore allows to track demonstrated motions with high precision. It generates trajectories towards regions of the state space, where training data is available and thus model fidelity is high. We proof uniform global asymptotic stability of the generated trajectories. In an evaluation with real human data, we show the advantages of the proposed approach in a reproduction task of human goal-directed motions.

ACKNOWLEDGEMENTS

The ERC Grant "Control based on Human Models" supported this work under grant agreement no. 337654.

REFERENCES

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

Beckers, T., Umlauft, J., Kulic, D., and Hirche, S. (2017). Stable Gaussian process based tracking control of Lagrangian systems. In *Conference on Decision and Control (CDC)*, 5180–5185. IEEE. doi: 10.1109/CDC.2017.8264427.

Blocher, C., Saveriano, M., and Lee, D. (2017). Learning stable dynamical systems using contraction theory. In *International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 124–129. IEEE.

Ijspeert, A.J., Nakanishi, J., and Schaal, S. (2002). Movement imitation with nonlinear dynamical systems in humanoid robots. In *International Conference on Robotics and Automation (ICRA)*. IEEE.

Khalil, H.K. and Grizzle, J. (1996). *Nonlinear systems*, volume 3. Prentice hall New Jersey.

Khansari-Zadeh, S.M. and Billard, A. (2011). Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Trans. on Robotics*, 27(5), 943–957.

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

Kocijan, J. (2016). *Modelling and Control of Dynamic Systems Using Gaussian Process Models*. Springer.

Lang, M., Endo, S., Dunkley, O., and Hirche, S. (2017). Object Handover Prediction using Gaussian Processes clustered with Trajectory Classification. *arXiv:1707.02745*.

Papachristodoulou, A. and Prajna, S. (2005). A tutorial on sum of squares techniques for systems analysis. In *American Control Conference (ACC)*, 2686–2700. IEEE.

Rasmussen, C.E. and Williams, C.K. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA.

Umlauft, J., Beckers, T., Kimmel, M., and Hirche, S. (2017a). Feedback linearization using Gaussian processes. In *Conference on Decision and Control (CDC)*, 5249–5255. IEEE.

Umlauft, J., Pöhler, L., and Hirche, S. (2018). An uncertainty-based control Lyapunov approach for control-affine systems modeled by Gaussian process. *IEEE Control Systems Letters*, 2(3), 483–488.

Umlauft, J., Lederer, A., and Hirche, S. (2017b). Learning stable Gaussian process state space models. In *American Control Conference (ACC)*, 1499–1504. IEEE.

Appendix A. COMPUTATION OF THE PATH INTEGRAL

We aim to compute the integral over the GP variance $\sigma_j^2(\boldsymbol{x})$ as defined in (13) along the line from $\boldsymbol{x}_m$ to $\boldsymbol{x}_{m+1}$, described as $\boldsymbol{\gamma}(s) = \bar{\boldsymbol{x}} + s\tilde{\boldsymbol{x}}$, $s \in [0,\ 1]$. We define $\underline{\boldsymbol{x}} = \boldsymbol{x}_m$, $\bar{\boldsymbol{x}} = \boldsymbol{x}_{m+1}$, $\tilde{\boldsymbol{x}} = \bar{\boldsymbol{x}} - \underline{\boldsymbol{x}}$, $k_i = k(\boldsymbol{x}, \boldsymbol{x}^{(i)})$ and B_{ij} is defined as the i,j-element of $\left(\boldsymbol{K} + \sigma_{\text{on}}^2 \boldsymbol{I}_N\right)^{-1}$. We drop the index j of the variance function for notational simplicity. The integral is formulated as

$$L(\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}) = \int_0^1 \sigma^2(\boldsymbol{\gamma}(s)) \left\| \frac{d\boldsymbol{\gamma}(s)}{ds} \right\| ds,$$

where $\boldsymbol{\gamma}(0) = \underline{\boldsymbol{x}}$ and $\boldsymbol{\gamma}(1) = \bar{\boldsymbol{x}}$. We rewrite the length of the path $\left\| \frac{d\boldsymbol{\gamma}(s)}{ds} \right\| = \|\tilde{\boldsymbol{x}}\|$ and substitute $k(\boldsymbol{x}, \boldsymbol{x}) = \sigma_f^2$, then

$$L(\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}) = \|\tilde{\boldsymbol{x}}\| \int_0^1 \sigma_f^2 - \sum_{i,j=1}^{N} B_{ij} k\left(\boldsymbol{x}^{(i)}, \boldsymbol{\gamma}(s)\right) k\left(\boldsymbol{x}^{(j)}, \boldsymbol{\gamma}(s)\right) ds$$

To compute the integral over a SE kernel product, we write

$$k\left(\boldsymbol{x}^{(i)}, \boldsymbol{\gamma}(s)\right) k\left(\boldsymbol{x}^{(j)}, \boldsymbol{\gamma}(s)\right) = \sigma_f^4 \exp\left(-as^2 - b_{ij}s - c_{ij}\right)$$

$$\text{where } a = \sum_{k=1}^{n} \frac{\tilde{x}_k^2}{l_k^2}, \quad b_{ij} = \sum_{k=1}^{n} \frac{\tilde{x}_k \left(2\underline{x}_k - x_k^{(i)} - x_k^{(j)}\right)}{l_k^2},$$

$$c_{ij} = \sum_{k=1}^{n} \frac{\frac{1}{2}{x_k^{(i)}}^2 + \frac{1}{2}{x_k^{(j)}}^2 + \underline{x}_k^2 - \left(x_k^{(i)} + x_k^{(j)}\right)\underline{x}_k}{l_k^2}.$$

Thus, we introduce

$$M_{ij} := \int_0^1 k\left(\boldsymbol{x}^{(i)}, \boldsymbol{\gamma}(s)\right) k\left(\boldsymbol{x}^{(j)}, \boldsymbol{\gamma}(s)\right) ds$$

$$= \sigma_f^4 \frac{\sqrt{\pi} \exp\left(\frac{b_{ij}^2}{4a} - c_{ij}\right)}{2\sqrt{a}} \left(\operatorname{erf}\left(\frac{2a + b_{ij}}{2\sqrt{a}}\right) - \operatorname{erf}\left(\frac{b_{ij}}{2\sqrt{a}}\right)\right),$$

where $\operatorname{erf}(\cdot)$ is the Gauss error function. Thus, we conclude

$$L(\underline{\boldsymbol{x}}, \bar{\boldsymbol{x}}) = \|\tilde{\boldsymbol{x}}\| \left(\sigma_f^2 - \sum_{i,j=1}^{N} B_{ij} M_{ij}\right).$$