Stability of Controllers for Gaussian Process Forward Models

Julia Vinogradska[1,2] JULIA.VINOGRADSKA@DE.BOSCH.COM
Bastian Bischoff[1] BASTIAN.BISCHOFF@DE.BOSCH.COM
Duy Nguyen-Tuong[1] DUY.NGUYEN-TUONG@DE.BOSCH.COM
Henner Schmidt[1] SCHMIDT.HENNER@GMX.DE
Anne Romer[1] FR.ANNEROMER@GOOGLEMAIL.COM
Jan Peters[2,3] MAIL@JAN-PETERS.NET

[1]Corporate Research, Robert Bosch GmbH; [2]IAS Lab, TU Darmstadt; [3]MPI for Intelligent Systems

Abstract

Learning control has become an appealing alternative to the derivation of control laws based on classic control theory. However, a major shortcoming of learning control is the lack of performance guarantees which prevents its application in many real-world scenarios. As a step in this direction, we provide a stability analysis tool for controllers acting on dynamics represented by Gaussian processes (GPs). We consider arbitrary Markovian control policies and system dynamics given as (i) the mean of a GP, and (ii) the full GP distribution. For the first case, our tool finds a state space region, where the closed-loop system is provably stable. In the second case, it is well known that infinite horizon stability guarantees cannot exist. Instead, our tool analyzes finite time stability. Empirical evaluations on simulated benchmark problems support our theoretical results.

1. Introduction

Learning control based on Gaussian process (GP) forward models has become an established approach in the machine learning and control theory communities. Many successful applications impressively demonstrate the efficiency of this approach (Deisenroth et al., 2015; Engel et al., 2006; Pan & Theodorou, 2014; Kocijan et al., 2004; Klenske et al., 2013; Maciejowski & Yang, 2013; Nguyen-Tuong & Peters, 2011). In contrast to classic control theory methods, learning control does not presuppose a detailed understanding of the underlying dynamics but tries to infer the required information from data. Thus, only very little expert knowledge about the system dynamics is required and fewer assumptions, such as a parametric form and parameter estimates, must be made. Employing GPs as forward models for learning control is particularly appealing since GPs incorporate uncertainty about the system dynamics.

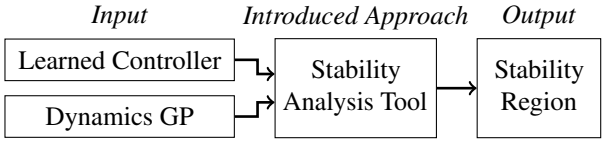


Figure 1. We present a tool to analyze stability for a given controller and Gaussian process (GP) dynamics model. Our tool analytically derives a stability region where convergence to the goal is guaranteed. Thus, we can provide stability guarantees for many of the existing GP based learning control approaches.

Unfortunately, performance guarantees do rarely exist for system dynamics and policies learned from data. An important type of performance guarantee is *(asymptotic) stability*. A *stability region* in the state space ensures, that all trajectories starting in this region converge to the target. Classic control theory offers a rich variety of stability analysis, e.g., for linear, nonlinear, and stochastic systems (Khalil, 2014; Skogestad & Postlethwaite, 2005; Kushner, 1967; Khasminskii & Milstein, 2011). However, to our best knowledge, no stability guarantees exist for GP based control. Thus, in contrast to classic control theory, there is no means to assess the effect of a controller on closed-loop structures with GP forward models. This major shortcoming prevents application of learned controllers in many real-world scenarios.

As a contribution to overcome this drawback, we present a tool to check the stability of learned policies for GP closed-loop control systems. Two types of dynamics are considered in this paper: dynamics given as (i) the mean of a GP, and (ii) the full GP distribution. While the first case results in a deterministic system, uncertainty is present in the second case. The notions of stability from deterministic analysis do not apply here. Especially when uncertainty is unbounded,

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

e.g., Gaussian state distribution, analysis over an infinite time horizon will typically result that any point is unstable (Steinhardt & Tedrake, 2012). To obtain meaningful results in case (ii), we follow the path to analyze stability for a finite time horizon (Steinhardt & Tedrake, 2012; Kushner, 1966). Unfortunately, even for finite time horizons, propagating state distributions through a GP is analytically intractable. Therefore, we present a novel approach based on numerical quadrature to approximate GP predictions when the input is a distribution. For stability guarantees, we rely on quadrature error analysis. In contrast to commonly employed approaches, our method handles complex non-Gaussian distributions with multiple modes. Based on this method, our tool finds a starting state region, s. t. after a given, finite time the goal is reached with a desired probability.

We believe that this paper is one of the first steps towards theoretical stability analysis for control of probabilistic models learned from data. The presented tool can provide stability guarantees for many existing learning control approaches based on GPs. The paper will be organized as follows: first, we briefly review related work and introduce the problem to be addressed. Section 2 provides necessary background for the stability analysis. In sections 3 and 4, a tool to check stability of closed-loop control with a GP system model is introduced. Section 5 provides empirical evaluations on benchmark control tasks. A conclusion is given in Section 6.

1.1. Related Work

To date, only very few special cases of system dynamics and policies learned from data have been analyzed with respect to stability. For example, (Perkins & Barto, 2003; Nakanishi et al., 2002) analyze the scenario of an agent switching between given safe controllers. In (Kim & Ng, 2005), stability is monitored while learning control for the special case of a linear controller and linear dynamics. To our best knowledge, stability analysis for probabilistic models, such as GPs, and arbitrary policies has not been addressed so far.

In classic control theory, the first formal analysis of closed-loop dynamics dates back to the 19th century (Routh, 1877; Hurwitz, 1895; Lyapunov, 1992). Lyapunov's approach allows to analyze stability for nonlinear systems $\dot{x} = f(x)$. It studies the system behavior around *equilibrium points*, i.e., points, where all derivatives $\dot{x}$ of the state vanish and the system comes to a hold. An equilibrium point x_e is *stable* , if for every $\varepsilon > 0$ there exists $\delta > 0$ such that $\|x(t) - x_e\| < \epsilon$ for every solution $x(t)$ of the differential equation with $\|x(t_0) - x_e\| < \delta$. The equilibrium point is *asymptotically stable*, if it is stable and δ can be chosen such that $\|x(t) - x_e\| \to 0$ as $t \to \infty$, cf. (Khalil, 2014). Stability of x_e follows directly from the existence of a *Lyapunov function*. While this approach applies even to nonlinear dynamics, it is not constructive and no general method exists.

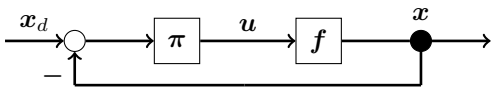


Figure 2. A closed-loop control structure with controller $\boldsymbol{\pi}$ and the system dynamics $\boldsymbol{f}$. We study stability for two types of dynamics: (i) the mean of the GP and (ii) the full GP predictive distribution.

For consideration of uncertainty in the dynamics, stochastic differential equations (SDEs) have been introduced (Adomian, 1983). SDEs are differential equations where some terms are stochastic processes. In the presence of uncertainty, x_e is *stable in probability*, if for every $\varepsilon > 0$ and $p > 0$, there exists $\delta > 0$ such that $\mathrm{P}\{\|x(t) - x_e\| > \varepsilon\} < p$ for $t > t_0$ and solutions $x(t)$ with $\|x(t_0) - x_e\| < \delta$. Furthermore, x_e is *asymptotically stable in probability*, if it is stable in probability and $\mathrm{P}\{\|x(t) - x_e\| > \varepsilon\} \to 0$ for a suitable δ, cf. (Khasminskii & Milstein, 2011). Stability follows from the existence of a *supermartingale*, a stochastic Lyapunov function analogue. However, supermartingales exist only under noise constraints. Relaxing the supermartingale criterion allows for stability statements for a finite time horizon, cf. (Kushner, 1966; Steinhardt & Tedrake, 2012).

1.2. Problem Statement

The goal of this paper is to provide a tool to analyze stability of a closed-loop structure when the dynamics model is given as a GP, see Figure 1. As inputs, our tool expects a control policy and a GP dynamics model. It checks the stability of the corresponding closed-loop structure and returns a stability region. Trajectories starting in this region are guaranteed to converge to the target (in probability). Subsequently, we discuss the different components of the tool in more detail.

We consider controllers, which depend only on the current state and are differentiable with respect to the state. The dynamics model is given as a GP with squared exponential covariance function. Our analysis generalizes to many other kernel choices, but this will not be discussed in this paper. We consider a discrete-time system $\boldsymbol{x}^{(t+1)} = \boldsymbol{f}(\boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)})$ with $\boldsymbol{x}^{(t)} \in \mathbb{R}^D$, $\boldsymbol{u}^{(t)} \in \mathbb{R}^F$, $t = 1, 2, \ldots$ and a controller $\boldsymbol{\pi} \colon \mathbb{R}^D \to \mathbb{R}^F$, whose objective is to move the system to a desired state $\boldsymbol{x}^d$, see Figure 2. In this paper, we study two possible cases for the dynamics $\boldsymbol{f}$: (i) the mean of a GP and (ii) the full GP predictive distribution. Note, that in the second case, distributions have to be propagated through the GP resulting in non-Gaussian state distributions.

2. Preliminaries

In this section, we introduce basic concepts for the proposed stability analysis. First, we briefly discuss Gaussian process regression, as GPs are employed to describe the considered dynamics. Second, numerical quadrature is introduced, as it is crucial for the proposed uncertainty propagation method.

2.1. Gaussian Process Regression

Given noisy observations $\mathcal{D} = \{(\boldsymbol{z}^i, y^i = f(\boldsymbol{z}^i) + \varepsilon) \mid 1 \leq i \leq N\}$, where $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$, the prior on f is $\mathcal{N}(0, K(Z,Z) + \sigma_n^2 I)$. The covariance matrix $K(Z,Z)$ is defined by the squared exponential kernel

$$k(\boldsymbol{z}, \boldsymbol{w}) = \sigma_f^2 \exp\left(-1/2(\boldsymbol{z} - \boldsymbol{w})^\intercal \Lambda^{-1} (\boldsymbol{z} - \boldsymbol{w})\right),$$

with signal variance σ_f^2 and squared lengthscales $\Lambda = \operatorname{diag}(l_1^2, \ldots, l_{D+F}^2)$ for all input dimensions. Given a query point $\boldsymbol{z}_*$, the conditional probability of $f(\boldsymbol{z}_*)$ is

$$f(\boldsymbol{z}_*) \mid \mathcal{D} \sim \mathcal{N}\big(\boldsymbol{k}(\boldsymbol{z}_*, Z)\boldsymbol{\beta},\ k(\boldsymbol{z}_*, \boldsymbol{z}_*) - \boldsymbol{k}(\boldsymbol{z}_*, Z)(K(Z,Z) + \sigma_n^2 I)^{-1} \boldsymbol{k}(Z, \boldsymbol{z}_*)\big) \quad (1)$$

with $\boldsymbol{\beta} = (K(Z,Z) + \sigma_n^2 I)^{-1} \boldsymbol{y}$. The hyperparameters $\sigma_n^2, \sigma_f^2, \Lambda$ are estimated by maximizing the log marginal likelihood of the data (Rasmussen & Williams, 2005).

In this paper, $\boldsymbol{f}$ models system dynamics. It takes state-action pairs $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u})^\intercal$ and outputs successor states $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$. As these outputs are multivariate, we train conditionally independent GPs for each output dimension. We write $\sigma_{n,m}^2$, $\sigma_{f,m}^2, \Lambda_m$ for the GP hyperparameters in output dimension m and k_m for the corresponding covariance function.

2.2. Numerical Quadrature

Numerical quadrature approximates the value of an integral

$$\int_a^b f(\boldsymbol{x}) d\boldsymbol{x} \approx \sum_{i=1}^{p} w_i f(\boldsymbol{\xi}_i)$$

given a finite number p of function evaluations. A widely used class of quadrature rules are interpolatory quadrature rules, which integrate all polynomials up to a certain degree exactly. In this paper, we employ Gaussian quadrature rules, where the evaluation points $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_p$ are chosen to be the roots of certain polynomials from orthogonal polynomial families. They achieve the highest accuracy possible for univariate interpolatory formulæ (Süli & Mayers, 2003). For multivariate integrals, the quadrature problem is significantly harder. While many formulæ for the univariate case can straightforwardly be generalized to multivariate integrals, they often suffer from the curse of dimensionality. However, quadrature methods that scale better and are feasible for up to 20 dimensions have been developed. See (Skrainka & Judd, 2011) for an overview.

3. Stability of GP Mean Dynamics

In this section, we provide a tool to check the stability of a closed-loop control structure with a given differentiable Markovian control policy π and GP mean dynamics. We attempt to find a region of starting points in the state space, such that trajectories starting in this region are guaranteed to converge to the desired point $\boldsymbol{x}^d$. To obtain this result, we analyze the sensitivity of the GP predictive mean to variations in the input space and derive upper bounds for the distance of the predictions at two different points. This analysis can be used to find a region around $\boldsymbol{x}^d$, where the next state is closer to $\boldsymbol{x}^d$ than the point before. It follows straightforwardly, that the full metric ball of maximal radius which lies completely inside this region is a stability region.

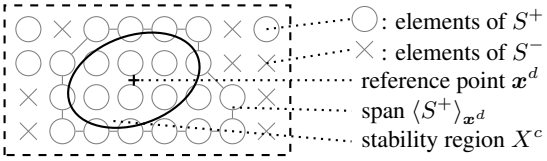


Figure 3. Basic idea of Algorithm 1 to construct a stability region: for a finite set of states S, employ upper bound from Lemma 1 to check whether the successor state is closer to $\boldsymbol{x}^d$, obtaining sets S^+ and S^-. Return metric ball of maximal radius that fits into the span $\langle S^+ \rangle_{\boldsymbol{x}^d}$ as stability region X^c. For $\boldsymbol{x}^{(0)} \in X^c$, the controlled system never leaves X^c and converges to $\boldsymbol{x}^d$ as $t \to \infty$.

In the following, we will briefly review the notion of stability employed for the analysis, derive an algorithm to find a stability region and, subsequently, prove its correctness.

3.1. Stability Notion

If the system dynamics $\boldsymbol{f}$ is given by the mean of a GP, the resulting closed-loop structure is deterministic. To assess the quality of a controller $\boldsymbol{\pi}$, which aims to stabilize the system at the reference point $\boldsymbol{x}^d$, we propose an algorithm to find a stability region of the closed-loop system.

DEFINITION 1. The reference point $\boldsymbol{x}^d$ is *stable*, if for every $\varepsilon > 0$ there exists $\delta > 0$, such that $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| < \varepsilon$ for $t = 1, 2, \ldots$ and $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^d\| < \delta$. If $\boldsymbol{x}^d$ is stable and there exists a $\delta > 0$ such that $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| \to 0$ for $t \to \infty$, $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^d\| < \delta$, $\boldsymbol{x}^d$ is *asymptotically stable*. A subset X^c of the state space is a *stability region*, if $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| \to 0$ as $t \to \infty$ for all $\boldsymbol{x}^{(0)} \in X^c$.

This definition matches Lyapunov's stability notion, cf. (Khalil, 2014). The algorithm below checks asymptotical stability of $\boldsymbol{x}^d$ by constructing a stability region X^c.

3.2. Algorithm Sketch

We derive an algorithm, that can find a stability region of a closed-loop control structure with GP mean dynamics. To find a stability region, we analyze how the distance between the current state and the target evolves. The basic idea of the algorithm involves *positively invariant sets*, i.e., sets that once entered, the system will never leave again (Blanchini, 1999). More precisely, we aim to find all $\boldsymbol{x}$, such that

$$\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\| < \gamma \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| \quad (2)$$

Algorithm 1 Stability region X^c for GP mean dynamics

Construct grid S, $S^+ \leftarrow \emptyset$, $X^c \leftarrow \emptyset$
for $\boldsymbol{x}^{(0)} \in S$ **do**
  Compute upper bound $C \geq \|\boldsymbol{x}^{(1)} - \boldsymbol{x}^d\|$
  if $C < \gamma \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^d\|$ **then** $S^+ \leftarrow S^+ \cup \{\boldsymbol{x}\}$ **fi**
od
if $\langle S^+ \rangle_{\boldsymbol{x}^d}$ exists **then**
  Fit metric ball $\mathcal{B}(\boldsymbol{x}^d) \subseteq \langle S^+ \rangle_{\boldsymbol{x}^d}$
  $X^c \leftarrow \mathcal{B}(\boldsymbol{x}^d)$ **fi**
return X^c

for $\boldsymbol{x}^{(t)} = \boldsymbol{x}$ and a fixed $\gamma < 1$. Assume there is a region containing $\boldsymbol{x}^d$, where Eq. (2) holds. For all states in this region, the distance to the target point decreases in one time step. If it is possible to fit a full metric ball centered at the target point $\boldsymbol{x}^d$ in this region, then all trajectories starting in the ball will never leave it. In addition, the distance of the current state to the reference point will decrease in every step by at least factor γ. Thus, it follows from the existence of such a ball, that the target point is asymptotically stable. However, finding a region in closed form, where Eq. (2) holds, is usually intractable. Instead, we follow an algorithmic approach to construct a stability region.

The proposed algorithm finds a region, where Eq. (2) holds, if one exists. It employs an upper bound for $\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\|$, that depends only on $\boldsymbol{x}^{(t)}$. With this upper bound we can check, whether Eq. (2) holds for a (finite) set of points. However, to find a state space region where Eq. (2) holds, this upper bound must allow for generalization from discrete points. More precisely, we need an upper bound of $\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\|$ that depends on $\boldsymbol{x}^{(t)}$ smoothly, thus having bounded gradients, and can be handled conveniently. We employ an upper bound derived from sensitivity analysis of the GP mean to variations in the input space. This bound depends on $\boldsymbol{x}^{(t)}$ in a way that can be exploited to compute a finite set of points S, e.g., a grid, that is sufficient to consider as follows: For any grid point, we check if Eq. (2) holds, obtaining the set S^+ with the grid points, that fulfill Eq. (2) and the rest in S^-. Let $\langle S^+ \rangle_{\boldsymbol{x}^d}$ be the polygon spanned by the connected component of S^+ containing $\boldsymbol{x}^d$. We can choose S such that Eq. (2) holds for all points in $\langle S^+ \rangle_{\boldsymbol{x}^d}$, see Figure 3. Algorithm 1 gives an overview, Sec. 3.3 provides technical details and proves correctness of the approach.

3.3. Correctness of the Algorithm

In this section, we elaborate on the computation steps of Algorithm 1 and prove that the returned state space region X^c is a stability region of the closed-loop system in the sense of Definition 1. Fundamental to Algorithm 1 is the upper bound for $\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\|$, which can be obtained as follows. We denote $\hat{\boldsymbol{x}} := (\boldsymbol{x}, \boldsymbol{\pi}(\boldsymbol{x}))^\intercal$ and recall that $\boldsymbol{f}(\hat{\boldsymbol{x}})$ is the GP predictive mean at $\hat{\boldsymbol{x}}$, see Eq. 1.

LEMMA 1. *Let $\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^d \in \mathbb{R}^{D+F}$ and $B \in \mathbb{R}^{D \times D}$ be a positive definite matrix. The distance of GP predictive means $\boldsymbol{f}(\hat{\boldsymbol{x}})$ and $\boldsymbol{f}(\hat{\boldsymbol{x}}^d)$ in the metrics induced by B is bounded by*

$$\|\boldsymbol{f}(\hat{\boldsymbol{x}}) - \boldsymbol{f}(\hat{\boldsymbol{x}}^d)\|_B^2 \leq (\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}^d)^\intercal M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})(\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}^d)$$

with a symmetric matrix M. This matrix can be constructed explicitly and depends on $\boldsymbol{x}, \boldsymbol{x}^d$ and the policy $\boldsymbol{\pi}$.

Proof. This statement is obviously true for $\boldsymbol{x} = \boldsymbol{x}^d$, so let $\boldsymbol{x} \neq \boldsymbol{x}^d$. Evaluating $\|\boldsymbol{f}(\hat{\boldsymbol{x}}) - \boldsymbol{f}(\hat{\boldsymbol{x}}^d)\|_B^2$ as

$$\sum_{m,m'=1}^{D} \sum_{i,k=1}^{N} b_{mm'}(k_m(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^i) - k_m(\hat{\boldsymbol{x}}^d, \hat{\boldsymbol{x}}^i)) \\ (k_{m'}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^k) - k_{m'}(\hat{\boldsymbol{x}}^d, \hat{\boldsymbol{x}}^k))\beta_{m,i}\beta_{m',k} \quad (3)$$

we realize the need for an upper bound of $k_m(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^i) - k_m(\hat{\boldsymbol{x}}^d, \hat{\boldsymbol{x}}^i)$. We integrate the gradient field of k_m along a curve $\boldsymbol{\tau}$ from $\hat{\boldsymbol{x}}^d$ to $\hat{\boldsymbol{x}}$. As this path integral does not depend on the particular curve $\boldsymbol{\tau}$, we may choose $\boldsymbol{\tau} = \boldsymbol{\tau}^{D+F} \dots \boldsymbol{\tau}^1$ as the curve along the edges of the hypercube defined by $\hat{\boldsymbol{x}}^d$ and $\hat{\boldsymbol{x}}$, i.e., $\boldsymbol{\tau}_p^j(t) = \hat{x}_p$ if $p \leq j-1$, $\boldsymbol{\tau}_p^j(r) = \hat{x}_p^d + r(\hat{x}_p - \hat{x}_p^d)$ with $r \in [0;1]$ if $p = j$, and $\boldsymbol{\tau}_p^j(t) = \hat{x}_p^d$ otherwise. This definition yields

$$k_m(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^i) - k_m(\hat{\boldsymbol{x}}^d, \hat{\boldsymbol{x}}^i) = \sum_{j=1}^{D+F} \int_{\hat{\boldsymbol{x}}_j^d}^{\hat{\boldsymbol{x}}_j} \frac{\partial k_m(\boldsymbol{\chi}, \hat{\boldsymbol{x}}^i)}{\partial \chi_j} \Bigg|_{\boldsymbol{\chi} = \boldsymbol{\tau}^j} d\chi_j \quad (4)$$

and we compute the partial derivatives as

$$\partial k_m(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^d)/\partial \hat{x}_j = -l_j^{-2}(\hat{x}_j - \hat{x}_j^d)k_m(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^d) \quad (5)$$

in all state-action space dimensions $1 \leq j \leq D+F$.

We rewrite the sum in Eq. (3) by substituting equations (4) and (5). To find an upper bound for this sum, we estimate the occurring integrals by the length of the integration interval times an upper or lower mean value (obtained, e.g., via Riemannian upper and lower sum) according to the sign of the respective summand. Sorting the summands by products of integration interval lengths $(\hat{x}_j - \hat{x}_j^d)(\hat{x}_p - \hat{x}_j^d)$, this sum can be rewritten as quadratic form. The entries of M can be chosen to form a symmetric matrix by making $M_{pj} = M_{jp}$ half of the coefficient of $(\hat{x}_j - \hat{x}_j^d)(\hat{x}_p - \hat{x}_p^d)$. □

For any point in the state action space, we can find an upper bound for the distance of its prediction and the target point. Note that this distance estimated by Lemma 1 depends heavily on the eigenvalues of $M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})$. This fact is exploited to compute a grid S, which constitutes the first step of Algorithm 1. As M is constructed as a symmetric matrix, the eigenvalue problem is well conditioned, i.e., when M is perturbed, the change in the eigenvalues of M is at most as large as the perturbation.

LEMMA 2. *Let $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^D$ and $M(\boldsymbol{x}, \boldsymbol{x}^d, \pi)$, B be as defined in Lemma 1. If $\|\boldsymbol{f}(\hat{\boldsymbol{x}}) - \boldsymbol{f}(\hat{\boldsymbol{x}}^d)\|_B < c\|\boldsymbol{x} - \boldsymbol{x}^d\|$ for $c < 1$, there exist Δ_j such that $\|\boldsymbol{f}(\hat{\boldsymbol{z}}) - \boldsymbol{f}(\hat{\boldsymbol{x}}^d)\|_B < c\|\boldsymbol{z} - \boldsymbol{x}^d\|$, for all $\boldsymbol{z}$ with $|\hat{z}_j - \hat{x}_j| < \Delta_j$, $1 \leq j \leq D + F$.*

Proof. Solving for the eigenvalues of $M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})$, the set $Q_{\boldsymbol{x}} := \{\boldsymbol{v} \in \mathbb{R}^{D+F} \,|\, (\boldsymbol{v} - \hat{\boldsymbol{x}}^d)^\intercal M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})(\boldsymbol{v} - \hat{\boldsymbol{x}}^d) < a\}$ can be determined. It is symmetric to the axes defined by the eigenvectors of $M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})$ and meets them at $\pm a^{-1}\sqrt{\lambda}$. For $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{\Delta}$ we want to ensure $\boldsymbol{z} \in Q_{\boldsymbol{z}}$, if $\boldsymbol{x} \in Q_{\boldsymbol{x}}$. Thus, we estimate how much the eigenvalues of $M(\boldsymbol{z}, \boldsymbol{x}^d, \boldsymbol{\pi})$ differ from those of $M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})$. As $M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})$ is symmetric, the eigenvalue problem has condition $\kappa(\lambda, M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})) = 1$ for any eigenvalue λ. Thus, $|\partial\lambda / \partial\hat{x}_j| \leq \|\partial M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi}) / \partial\hat{x}_j\|$. Computing $\|\partial M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi}) / \partial\hat{x}_j\|$ or upper bounds for this expression allows solving for all Δ_j. □

We are now able to compute a grid S, such that it is sufficient to check Eq. (2) for all grid points to retrieve a (continuous) stability region. While the grid width may become small, there is a lower bound for it. As the GP falls back to zero far away from training data, the entries of M are bounded and, being continuous, Lipschitz. Thus, a lower bound exists.

THEOREM 1. *The region X^c returned by Algorithm 1 is a stability region. All trajectories starting in X^c move closer to the desired point $\boldsymbol{x}^d$ in each step. Convergence to $\boldsymbol{x}^d$ is guaranteed for all points in X^c as $t \to \infty$.*

Proof. We exploit Lemma 2 to compute a grid S as the first step in Algorithm 1. Employing Lemma 1, the algorithm checks for all points in S, whether Eq. (2) holds, obtaining the sets S^+ and S^-, respectively. Lemma 2 ensures that Eq. (2) also holds for all points inside the polygon $\langle S^+ \rangle_{\boldsymbol{x}^d}$. For every point in $\langle S^+ \rangle_{\boldsymbol{x}^d}$, the next state is closer to the target point $\boldsymbol{x}^d$. Fitting a full metric ball $\mathcal{B}(\boldsymbol{x}^d)$ in $\langle S^+ \rangle_{\boldsymbol{x}^d}$ we recover an invariant set X^c. More precisely, the trajectories starting in X^c move closer to $\boldsymbol{x}^d$ with every time step. □

4. Stochastic Stability of GP Dynamics

In this section, we study closed-loop systems with dynamics given as a full GP distribution. For any query point $\boldsymbol{x}^{(t)}$ a GP predicts the next state $\boldsymbol{x}^{(t+1)}$ to be normally distributed. If, however, $\boldsymbol{x}^{(t)}$ is not a point, but a distribution, the integral

$$p(\boldsymbol{x}^{(t+1)}) = \int_{\mathbb{R}^D} p(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)}) p(\boldsymbol{x}^{(t)}) d\boldsymbol{x}^{(t)} \quad (6)$$

determines the next state distribution. Note that $p(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)})$ is Gaussian with respect to $\boldsymbol{x}^{(t+1)}$. The next state distribution $p(\boldsymbol{x}^{(t+1)})$, however, is not Gaussian, even if $p(\boldsymbol{x}^{(t)})$ is. Generally, $p(\boldsymbol{x}^{(t+1)})$ is analytically intractable and only approximations, e.g., via moment matching (Quiñonero-Candela et al., 2003) or linearization (Ko & Fox, 2008), can be computed. These methods suffer from severe inaccuracies in many cases and, thus, are unsuitable for stability analysis. In this paper, we propose numerical quadrature to approximate $p(\boldsymbol{x}^{(t+1)})$, instead. This technique yields significantly better results, e.g., it can handle distributions with multiple modes. In addition, error analysis is readily available (Wasowicz, 2006; Masjed-Jamei, 2014) and can be employed to derive stability guarantees.

In the following, we will discuss *finite time stochastic stability*, introduce an algorithm to find a stability region based on numerical quadrature, and prove its correctness.

4.1. Stability Notion

Consider a deterministic system, that is locally, but not globally, asymptotically stable. Adding noise to the system may render the target point unstable. Especially when noise is unbounded (e.g., Gaussian), all trajectories will eventually leave any ball around the target point with probability 1. For this reason, in the study of SDEs, other stability notions than Lyapunov's are common. In particular, bounding the probability to drift away from the target over a finite time T is desirable, as in the following definition (Kushner, 1966).

DEFINITION 2. Let Q_1, Q_2 be subsets of the state space X, with Q_2 open and $Q_1 \subset Q_2 \subset X$. The system is *finite time stable* with respect to $Q_1, Q_2, 1 - \lambda, T$, if $\boldsymbol{x}^{(0)} \in Q_1$ implies $P\{\boldsymbol{x}^{(t)} \in Q_2\} \geq 1 - \lambda$ for all $t \leq T$.

However, we are interested in finding a set Q_s of initial conditions, such that the goal Q is reached within time T with a desired probability, cf. (Steinhardt & Tedrake, 2012).

DEFINITION 3. The set Q_s is a *stability region* with respect to the target region Q, time horizon T, success probability $1 - \lambda$, if $P\{\boldsymbol{x}^{(T)} \in Q\} \geq 1 - \lambda$ holds for all $\boldsymbol{x}^{(0)} \in Q_s$ with $\lambda > 0$ and the target region $Q \subset X$.

This definition focuses on reaching a target Q after time T, whereas Definition 2 bounds the exit probability from a region Q_2 within time $0 \leq t \leq T$. The methods proposed in this paper can be employed to analyze stability in the sense of both definitions. In the following, we will present an algorithm to find a stability region according to Definition 3.

4.2. Algorithm Sketch

We will now discuss how to find a stability region as in Definition 3 for a closed-loop system with GP dynamics. To analyze system behavior, the capability to compute next state distributions is crucial. As discussed previously, Eq. (6) is analytically intractable and approximation methods must be employed. We propose numerical quadrature to approximately propagate distributions . We will show that numerical quadrature approximates state distributions as Gaussian

Algorithm 2 Stability region for GP dynamics

Input: T, target region Q, approximation err. tolerance e_{tol}
Output: stability region X^c

Construct grid S; $S^+ \leftarrow \emptyset$
Compute quadrature $\boldsymbol{w}, \boldsymbol{\xi}$, such that $\|\epsilon_T\|_{\mathcal{C}(X)} < e_{tol}$
for $\boldsymbol{x} \in S$ **do**
  $p(\boldsymbol{x}^{(1)}) \leftarrow \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\pi}(\boldsymbol{x}))$; $\boldsymbol{x}^{(T)} \leftarrow \boldsymbol{\alpha}^{(T)}\boldsymbol{\phi}$
  $P_{min}\{\boldsymbol{x}^{(T)} \in Q\} \leftarrow \boldsymbol{\alpha}^{(T)}\boldsymbol{m} - \mathrm{vol}(Q)\|\epsilon_T\|_{\mathcal{C}(X)}$
  if $P_{min}\{\boldsymbol{x}^{(T)} \in Q\} > 1-\lambda$ **then** $S^+ \leftarrow S^+ \sqcup \{\boldsymbol{x}\}$ **fi**
od
return $X^c \leftarrow \langle S^+ \rangle$

mixture models. Fortunately, computation of multi-step-ahead predictions becomes convenient and fast even for long time horizons. However, relying on approximations of the state distribution is not sufficient for stability guarantees. The error introduced by approximation and error propagation must be bounded to recover reliable statements on the probability for $\boldsymbol{x}^{(T)}$ to be in a set Q. This error bound $\epsilon_t(\boldsymbol{x})$ at time t can be obtained following one of the available quadrature error analyses, e.g., (Masjed-Jamei, 2014).

With numerical quadrature and quadrature error analysis, we can compute a lower bound for the *success probability*, i.e., the probability for $\boldsymbol{x}^{(T)}$ to be in the target set Q. A priori, this enables checking success probabilities for a finite set of points. Fortunately, as in the case of GP mean dynamics, the statement can be generalized to continuous regions. Our algorithm constructs a grid in the state space and computes success probabilities for all grid points. We prove that a stability region can be inferred from these grid point results.

In the following, we elaborate on numerical quadrature as approximate inference method and, studying quadrature error propagation, prove the correctness of the proposed approach. The presented tool is outlined in Algorithm 2.

4.2.1. Numerical Quadrature

In most applications, especially when the states have physical interpretations, the state space is bounded. For this reason, we assume $X = [a_1, b_1] \times \cdots \times [a_D, b_D]$ and solve

$$F[p(\boldsymbol{x}^{(t)})] := \int_X p(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)}) p(\boldsymbol{x}^{(t)}) d\boldsymbol{x}^{(t)}. \quad (7)$$

We propose Gaussian product integration, which extends univariate Gaussian quadrature to a multivariate rule using a product grid $\mathbb{X}$ of evaluation points and positive weights w_n for all nodes $\boldsymbol{\xi}^n \in \mathbb{X}$. Integral (7) is then approximated by

$$F[p(\boldsymbol{x}^{(t)})] \approx \sum_{\boldsymbol{\xi}^n \in \mathbb{X}} w_n p(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)} = \boldsymbol{\xi}^n) p(\boldsymbol{x}^{(t)} = \boldsymbol{\xi}^n), \quad (8)$$

resulting in a weighted sum of Gaussian distributions. The approximate state distribution at time $t+1$ can be given by

$$p(\boldsymbol{x}^{(t+1)}) \approx \boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(t+1)} \quad (9)$$

with $\alpha_n^{(t+1)} := w_n p(\boldsymbol{x}^{(t)} = \boldsymbol{\xi}^n)$, $\phi_n(\boldsymbol{x}) := p(\boldsymbol{x}^{(t+1)} = \boldsymbol{x} \mid \boldsymbol{x}^{(t)} = \boldsymbol{\xi}^n)$. Note that the Gaussian basis functions $\phi_n(\boldsymbol{x})$ do not change over time, so the state distribution at time t is represented by the weight vector $\boldsymbol{\alpha}^{(t)}$. To propagate any distribution multiple steps through the GP, the basis functions ϕ_n must be calculated only once and the task reduces to sequential updates of the weight vector $\boldsymbol{\alpha}$. As $p(\boldsymbol{x}^{(t)}) \approx \boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(t)}$, the weight vector $\boldsymbol{\alpha}^{(t+1)}$ is given by

$$\boldsymbol{\alpha}^{(t+1)} = \mathrm{diag}(\boldsymbol{w})\Phi\boldsymbol{\alpha}^{(t)} = (\mathrm{diag}(\boldsymbol{w})\Phi)^t \boldsymbol{\alpha}^{(1)}$$

with the matrix Φ, $\Phi_{ij} = \phi_j(\boldsymbol{\xi}^i)$ with $1 \le i, j \le n$, which contains the basis function values at all grid points.

Our algorithm aims to find a stability region Q_s, i.e., a region where the success probability is at least $1-\lambda$ for a given time horizon T, target region Q, and $\lambda > 0$. Computing $\boldsymbol{m} = \int_Q \boldsymbol{\phi}(\boldsymbol{x}) d\boldsymbol{x}$, the probability for $\boldsymbol{x}^{(T)}$ to be in Q is approximately $\boldsymbol{m}^\intercal \boldsymbol{\alpha}^{(T)}$.

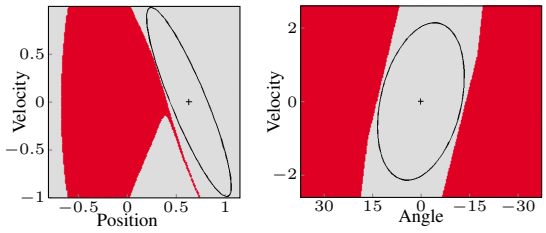
4.3. Correctness of the Algorithm

We will now show that the region returned by Algorithm 2 is a stability region as in Definition 3. In Section 4.2.1 we proposed numerical quadrature to approximate GP predictions when the input is a distribution. However, to obtain stability guarantees it is not sufficient to consider approximate solutions. For a lower bound on the success probability, additional knowledge about the approximation error and how it is propagated through the dynamics GP is essential.

When approximate inference steps are cascaded for multi-step-ahead predictions, errors are introduced and propagated through the dynamics. Typically, error propagation fortifies initial errors with every time step. The difference of approximation and true distribution after a finite time is bounded. Fortunately, numerical quadrature allows to control this error bound by refining the used quadrature rule.

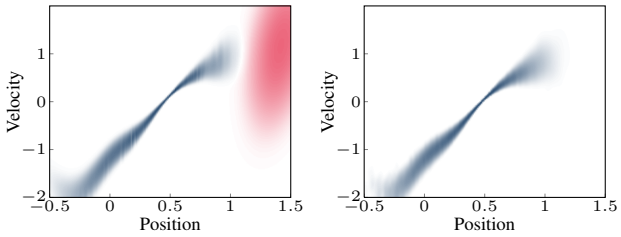
Lemma 3. *Let $\boldsymbol{x}^{(0)} \in X$ and ϵ_T be the pointwise approximation error $\epsilon_T(\boldsymbol{x}) = p(\boldsymbol{x}^{(T)} = \boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{x})^\intercal \boldsymbol{\alpha}^{(T)}$ at time T. There exists an upper bound for $\|\epsilon_T\|_{\mathcal{C}(X)}$ that can be controlled by the choice of quadrature rule.*

Proof. The statement obviously holds for $T{=}0$ and for $T{=}1$, as $p(\boldsymbol{x}^{(1)})$ is Gaussian and computed exactly. If $T \ge 2$, an error is introduced as $p(\boldsymbol{x}^{(t)})$ cannot be computed analytically for $t \ge 2$. Employing numerical quadrature to compute $p(\boldsymbol{x}^{(2)})$, we obtain $p(\boldsymbol{x}^{(2)}) = \boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(2)} + \epsilon_2$ with our approximation $\boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(2)}$ and the unknown initial quadrature error $\epsilon_2(\boldsymbol{x})$. However, error analysis as in (Masjed-Jamei, 2014) gives us a bound for $\|\epsilon_2\|_{\mathcal{C}(X)} := \max_{\boldsymbol{x} \in X} |\epsilon_2(\boldsymbol{x})|$. When propagating further, an unknown, bounded error term and an approximation term must be handled. For the approximation term, we get $F[\boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(t)}] = \boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(t+1)} + \boldsymbol{\epsilon}^\intercal \boldsymbol{\alpha}^{(t)}$



(a) Mountain Car (b) Inverted Pendulum

Figure 4. Stability regions for GP mean dynamics on the two benchmark tasks. This figure shows empirically obtained stability regions in gray, points that did not converge in red. The ellipsoid indicates the stability region returned by our algorithm.



(a) Our NQ (gray) and MM (red) (b) MC Sampling

Figure 5. Multi-step-ahead prediction when the input is a distribution. This figure shows the results obtained with our numerical quadrature (NQ) approach in gray and moment matching (MM) in red (a) and the reference Monte Carlo (MC) sampling result (b).

with quadrature error bounds $\boldsymbol{\epsilon} = (\epsilon_i(\boldsymbol{x}))_i$ for the integrals (7), when setting $p(\boldsymbol{x}^{(t)}) = \phi_i$. The error term $\epsilon_t(\boldsymbol{x})$ is propagated to $F[\epsilon_t]$ with $\max_{\boldsymbol{x}\in X}|F[\epsilon_t]| \leq C\|\epsilon_t\|_{\mathcal{C}(X)}$ and $C = \max_{\boldsymbol{x}\in X}\int p(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)})d\boldsymbol{x}^{(t)}$. Finally, the error at time T is bounded by $\|\epsilon_T\|_{\mathcal{C}(X)} \leq \|\boldsymbol{\epsilon}\|^{\intercal}\boldsymbol{\alpha}^{(T-1)} + \sum_{t=2}^{T-2} C^t\|\boldsymbol{\epsilon}\|^{\intercal}\boldsymbol{\alpha}^{(t)} + C^{t-1}\|\epsilon_2\|$, where $\boldsymbol{\epsilon}$ and ϵ_2 can be made arbitrarily small by refining the quadrature rule. □

Lemma 3 theoretically enables stability guarantees for finite time horizons T. However, as with any approximation of the distribution (7), this error bound grows exponentially with T, while quadrature error decreases polynomially with function evaluations. This drawback limits real-word application of Lemma 3 to small T, but we found that typically, approximation behaves far better than the worst-case bound.

For any starting point, we can now compute the approximate state distribution at time T and a lower bound on the success probability, i.e., the probability for $\boldsymbol{x}^{(T)}$ to be in the target region Q. It remains to show that it is sufficient to compute the success probability for a discrete set of starting states and generalize to the underlying continuous state space region.

LEMMA 4. *Let $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^D$ be starting states and $T \in \mathbb{N}$. If $P\{\boldsymbol{x}^{(T)} \in Q\} \geq 1-\lambda$, there exist Δ_j such that $P\{\boldsymbol{z}^{(T)} \in Q\} \geq 1-\lambda$ for all $\boldsymbol{z}$ with $|z_j - x_j| < \Delta_j$, $1 \leq j \leq D$.*

Proof. Firstly, we note that the absolute error bound does not depend on the starting state. The state distribution $p(\boldsymbol{x}^{(1)})$ is the GP prediction at the query point $\boldsymbol{x}^{(0)}$. Thus, for $T = 1$ the claim follows from Lemma 2. When $t \geq 2$, only an approximation of the state distribution is known. The mixture model weights at time $t = 2$ depend only on the values of $p(\boldsymbol{x}^{(1)})$ at the set of evaluation points $\mathbb{X}$. Thus, the lemma follows for $T = 2$. If $t > 2$, the difference of the approximate distributions for the starting points $\boldsymbol{x}$ and $\boldsymbol{z}$ is

$$\boldsymbol{\phi}^{\intercal}(\boldsymbol{\alpha}_{\boldsymbol{x}}^{(t)} - \boldsymbol{\alpha}_{\boldsymbol{z}}^{(t)}) = \boldsymbol{\phi}^{\intercal}(\mathrm{diag}(\boldsymbol{w})\Phi)^{t-2}(\boldsymbol{\alpha}_{\boldsymbol{x}}^{(2)} - \boldsymbol{\alpha}_{\boldsymbol{z}}^{(2)}).$$

Thus, the difference in approximate probability mass is linear in $(\boldsymbol{\alpha}_{\boldsymbol{x}}^{(t)} - \boldsymbol{\alpha}_{\boldsymbol{z}}^{(t)})$. This fact concludes the proof. □

THEOREM 2. *The region X^c returned by Algorithm 2 is a stability region in the sense of Definition 3. For all starting points $\boldsymbol{x}^{(0)} \in X^c$ the probability mass in the target region Q at time T is at least $1-\lambda$.*

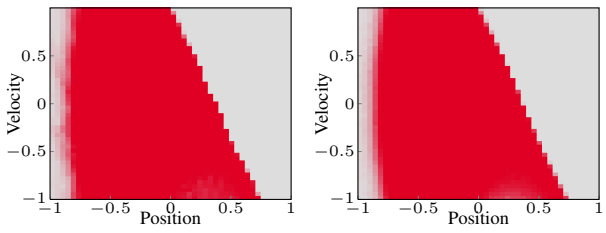
Proof. For the first step in Algorithm 2 we exploit Lemma 4 to construct a grid S. For S, stability follows for the region around the grid points S^+ with success probability $\geq 1-\lambda$. Secondly, employing Lemma 3 we determine a quadrature rule that ensures the pointwise error at time T to be at most e_{tol}. Finally, we compute approximate success probabilities for all grid points, as in Sec. 4.2.1. Subtracting the maximum error mass $e_{tol}\,\mathrm{vol}(Q)$ in target region Q, we obtain all grid points S^+, which have success probability $\geq 1-\lambda$. □

5. Empirical Evaluation

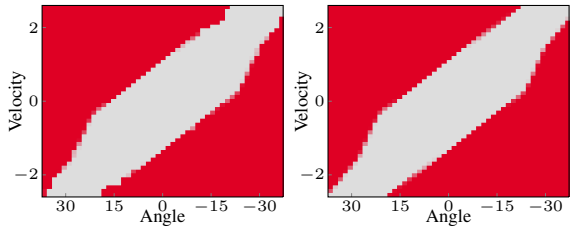
In this section, we evaluate the previously obtained theoretical stability results on two benchmark tasks: mountain car and inverted pendulum. Moreover, the performance of the proposed uncertainty propagation is compared to the state-of-the-art moment matching approach and Monte Carlo sampling. First, we briefly introduce the two test-beds.

Mountain Car A car with limited engine power has to reach a desired point in the mountainscape (Sutton & Barto, 1998). The state space has two dimensions: position and velocity of the car. We analyze stability of a PD-controller $\pi((x, \dot{x})^{\intercal}) = K_p x + K_d \dot{x}$. The gains are chosen as $K_p = 25$ and $K_d = 1$ and the control signal is limited to $u_{max} = 4$. The GP dynamics model was trained on 250 data points from trajectories with random starting points and control gains.

Inverted Pendulum The goal is to bring the pendulum to an upright position with limited torque (see (Doya, 2000)) and balance it there. The system state has two dimensions: pendulum angle and velocity. We evaluate stability of a PD-controller with $K_p = 6$, $K_d = 3$ and control limit $u_{max} = 1.2$. The dynamics GP was trained on 200 points from rollouts with random starting points and control gains.



(a) Mountain Car, Our NQ (b) Mountain Car, MC Sampling



(c) Pendulum, Our NQ (d) Pendulum, MC Sampling

Figure 6. Stability regions for dynamics given as GP full distribution and time horizon of 100s. The mountain car ((a) and (b)) and inverted pendulum ((c) and (d)) results were obtained with the proposed numerical quadrature (NQ) and Monte Carlo (MC).

5.1. Stability of GP Predictive Mean Dynamics

To evaluate the presented theory on stability of the closed-loop system with GP mean dynamics, a stability region is determined as described in Section 3. We compare this region to the true stability region, empirically obtained as follows. A grid on the state space is defined and rollouts from every grid point are computed. After a sufficiently long time (1000s), we check whether the state has converged to the target point. Thus, we empirically determine a region of starting points, where the system converges to the desired state. Figure 4 shows the obtained regions for the mountain car and pendulum system. In both cases the theoretically obtained stability region is a subset of the empirically determined region. This effect is due to our analysis yielding a full metric ball centered around the target point, although the full stability region is not necessarily convex. Also, trajectories which first move away from the target point, but finally converge to it, are not considered in the presented theory. Instead, all trajectories starting in the theoretically obtained stability region move towards the target point contractively.

5.2. Numerical Quadrature Uncertainty Propagation

The key to stability analysis for closed-loop systems with dynamics given as full GP distribution is the prediction at uncertain inputs. We compare the performance of the presented approximate inference to the state-of-the-art moment matching (MM) method and Monte Carlo (MC). Consider the following scenario: in the mountain car domain, we position the car on the right slope with some positive velocity. Furthermore, we introduce small Gaussian uncertainty about the starting state. We employ a constant control signal, that is too small to bring the car up directly. We compute rollouts, propagating state distributions through the GP with (i) the presented numerical quadrature (NQ), (ii) MM as in (Deisenroth, 2010), and (iii) MC. The resulting distributions for a time horizon of 1.2s are shown in Figure 5. The MM approximation differs significantly from MC and NQ results, concentrating most of the probability mass, where the MC approximation has very low probability density. The NQ result closely matches the distribution obtained with MC.

5.3. Stability of Gaussian Process Dynamics

Employing numerical quadrature, we determine stability regions for the two test-beds and a time horizon of 100s.As Figure 6 shows, the obtained stability regions match the empirical MC results. The error bound from Lemma 3 demands for extremely fine quadrature rules. For long time horizons, as in our experiments, this requirement is computationally infeasible. However, we found that the real-world results are substantially better than this worst-case bound. We also experienced computation time ($\approx$ 120s) for NQ to be a fraction of the time required for long time MC predictions. Of course, this will not hold for systems with many state dimensions and our particular setup of product quadrature rules, as these rules suffer from the curse of dimensionality. However, there are various approaches to overcome this drawback of NQ (Heiss & Winschel, 2008; Novak & Ritter, 1996; Xiao & Gimbutas, 2010; Ryu & Boyd, 2015) and our analysis holds for arbitrary quadrature rules.

6. Conclusion

In this paper, we have analyzed stability of controllers, that act on Gaussian process forward models. We have derived a tool to find stability regions for two possible types of system dynamics: (i) the mean and (ii) the full GP predictive distribution. In the first case, we have constructed a stability region, i.e., trajectories starting inside this region are guaranteed to converge to the target point and stay there for all times. The theoretical result has been compared to the empirically obtained stability region on two benchmark problems. In the second case, a novel approach based on numerical quadrature has been introduced to approximately propagate uncertainties through a GP. In contrast to other state-of-the-art methods, our approach can model complex distributions with multiple modes. Evaluation results closely match the true distribution approximated by extensive sampling. The introduced approximate inference method has been used to derive finite-time stability guarantees based on quadrature error analysis. Empirical Monte Carlo results confirm our theoretical results on the two benchmark problems. Overall, the proposed methods can provide stability guarantees for many existing learning control approaches based on GPs.

References

Adomian, George. *Stochastic systems*. Mathematics in Science and Engineering. Elsevier Science, 1983.

Blanchini, F. Set invariance in control. *Automatica*, 35(11): 1747 – 1767, 1999.

Deisenroth, Marc Peter, Fox, Dieter, and Rasmussen, Carl Edward. Gaussian processes for data-efficient learning in robotics and control. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(2):408–423, 2015.

Deisenroth, M.P. *Efficient Reinforcement Learning Using Gaussian Processes*. Karlsruhe series on intelligent sensor actuator systems. KIT Scientific Publ., 2010.

Doya, Kenji. Reinforcement learning in continuous time and space. *Neural Computation*, 12:219–245, 2000.

Engel, Yaakov, Szabo, Peter, and Volkinshtein, Dmitry. Learning to control an octopus arm with gaussian process temporal difference methods. In Weiss, Y., Schölkopf, B., and Platt, J.C. (eds.), *Advances in Neural Information Processing Systems 18*, pp. 347–354. MIT Press, 2006.

Heiss, Florian and Winschel, Viktor. Likelihood approximation by numerical integration on sparse grids. *Journal of Econometrics*, 144(1):62 – 80, 2008.

Hurwitz, A. Ueber die Bedingungen, unter welchen eine Gleichung nur Wurzeln mit negativen reellen Theilen besitzt. *Mathematische Annalen*, 46(2):273–284, 1895.

Khalil, Hassan K. *Nonlinear control*. Prentice Hall, 2014.

Khasminskii, R. and Milstein, G.N. *Stochastic Stability of Differential Equations*. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg, 2011.

Kim, H. J. and Ng, Andrew Y. Stable adaptive control with online learning. In Saul, L.K., Weiss, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems 17*, pp. 977–984. MIT Press, 2005.

Klenske, Edgar D, Zeilinger, Melanie Nicole, Scholkopf, Bernhard, and Hennig, Philipp. Nonparametric dynamics estimation for time periodic systems. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pp. 486–493. IEEE, 2013.

Ko, J. and Fox, D. GP-BayesFilters: Bayesian filtering using gaussian process prediction and observation models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008.*, 2008.

Kocijan, Juš, Murray-Smith, Roderick, Rasmussen, Carl Edward, and Girard, Agathe. Gaussian process model based predictive control. In *American Control Conference, 2004. Proceedings of the 2004*, volume 3, pp. 2214–2219. IEEE, 2004.

Kushner, H.J. Finite time stochastic stability and the analysis of tracking systems. *Automatic Control, IEEE Transactions on*, 11(2):219–227, 1966.

Kushner, H.J. *Stochastic Stability and Control*. Mathematics in science and engineering. Academic Press, 1967.

Lyapunov, A.M. *General Problem of the Stability Of Motion*. Control Theory and Applications Series. Taylor & Francis, 1992.

Maciejowski, Jan M and Yang, Xiaoke. Fault tolerant control using gaussian processes and model predictive control. In *Control and Fault-Tolerant Systems (SysTol), 2013 Conference on*, pp. 1–12. IEEE, 2013.

Masjed-Jamei, Mohammad. New error bounds for gauss-legendre quadrature rules. *Filomat*, 28(6):1281–1293, 2014.

Nakanishi, J., Farrell, J.A., and Schaal, S. A locally weighted learning composite adaptive controller with structure adaptation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.*, pp. 882–889 vol.1, 2002.

Nguyen-Tuong, D. and Peters, J. Model learning in robotics: a survey. *Cognitive Processing*, (4), 2011.

Novak, Erich and Ritter, Klaus. High dimensional integration of smooth functions over cubes. *Numerische Mathematik*, 75(1):79–97, 1996.

Pan, Yunpeng and Theodorou, Evangelos. Probabilistic differential dynamic programming. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., and Weinberger, K.Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 1907–1915. Curran Associates, Inc., 2014.

Perkins, Theodore J. and Barto, Andrew G. Lyapunov design for safe reinforcement learning. *J. Mach. Learn. Res.*, 3:803–832, March 2003.

Quiñonero-Candela, J., Girard, A., Larsen, J., and Rasmussen, C. E. Propagation of uncertainty in bayesian kernel models - application to multiple-step ahead forecasting. In *International Conference on Acoustics, Speech and Signal Processing*, pp. 701–704, vol. 2, 2003.

Rasmussen, Carl Edward and Williams, Christopher K. I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

Routh, Edward John. *A Treatise on the Stability of a Given State of Motion: Particularly Steady Motion*. Macmillan and Company, 1877.

Ryu, Ernest K. and Boyd, Stephen P. Extensions of gauss quadrature via linear programming. *Foundations of Computational Mathematics*, 15(4):953–971, 2015.

Skogestad, Sigurd and Postlethwaite, Ian. *Multivariable Feedback Control: Analysis and Design*. John Wiley & Sons, 2005.

Skrainka, Benjamin S and Judd, Kenneth L. High performance quadrature rules: How numerical integration affects a popular model of product differentiation. *Available at SSRN 1870703*, 2011.

Steinhardt, Jacob and Tedrake, Russ. Finite-time regional verification of stochastic nonlinear systems. In Durrant-Whyte, H.F., Roy, N., and Abbeel, P. (eds.), *Robotics: Science and Systems VII*, pp. 321–328. MIT Press, 2012.

Süli, E. and Mayers, D.F. *An Introduction to Numerical Analysis*. Cambridge University Press, 2003.

Sutton, R.S. and Barto, A.G. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Wasowicz, Szymon. On error bounds for gauss-legendre and lobatto quadrature rules. *JIPAM. Journal of Inequalities in Pure & Applied Mathematics*, 7(3):Paper No. 84, 7 p., 2006.

Xiao, Hong and Gimbutas, Zydrunas. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Computers & Mathematics with Applications*, 59(2):663 – 676, 2010.