

# Probabilistic robust linear quadratic regulators with Gaussian processes

Alexander von Rohr<sup>1,3,4</sup>

VONROHR@DSME.RWTH-AACHEN.DE

Matthias Neumann-Brosig<sup>2,3</sup>

MATTHIAS.NEUMANN-BROSIG@IAV.DE

Sebastian Trimpe<sup>1,4</sup>

TRIMPE@DSME.RWTH-AACHEN.DE

*1 Institut for Data Science in Mechanical Engineering, RWTH Aachen University, Germany*

*2 Institut Analysis und Algebra, TU Braunschweig, Germany*

*3 Ingenieurgesellschaft Auto und Verkehr (IAV)*

*4 Max Planck Institute for Intelligent Systems, Tübingen/Stuttgart, Germany*

**Editors:** A. Jadbabaie, J. Lygeros, G. J. Pappas, P. A. Parrilo, B. Recht, C. J. Tomlin, M. N. Zeilinger

## Abstract

Probabilistic models such as Gaussian processes (GPs) are powerful tools to learn unknown dynamical systems from data for subsequent use in control design. While learning-based control has the potential to yield superior performance in demanding applications, robustness to uncertainty remains an important challenge. Since Bayesian methods quantify uncertainty of the learning results, it is natural to incorporate these uncertainties into a robust design. In contrast to most state-of-the-art approaches that consider worst-case estimates, we leverage the learning method's posterior distribution in the controller synthesis. The result is a more informed and, thus, more efficient trade-off between performance and robustness. We present a novel controller synthesis for linearized GP dynamics that yields robust controllers with respect to a probabilistic stability margin. The formulation is based on a recently proposed algorithm for linear quadratic control synthesis, which we extend by giving probabilistic robustness guarantees in the form of credibility bounds for the system's stability. Comparisons to existing methods based on worst-case and certainty-equivalence designs reveal superior performance and robustness properties of the proposed method.

**Keywords:** learning-based control, Gaussian processes, probabilistic robust control

## 1. Introduction

The trade-off between performance and robustness is at the heart of any practical control design (Boulet and Duan, 2007): do we seek an optimal controller for a particular system *or* one that works well over a larger set of systems and conditions? It is possible to quantify this trade-off in the various methods of robust control (Zhou et al., 1996). Robust control, however, is typically based on a worst-case treatment of uncertainty sets and does not consider a distribution over the set of possible model parameters. Control design based on probabilistic models learned from data is less well established. Yet, for learned models, the performance-robustness trade-off is especially relevant since uncertainty is inherent in any learning result.

When controllers are deployed on physical machines, practical robustness guarantees are essential to ensure the plant's safe operation. Dealing with large uncertainties, worst-case controllers are often highly conservative and require additional assumptions to retain their guarantees. Given the learned probabilistic model, it is sometimes impossible to give deterministic stability guarantees if

the set of possible model parameters is too large or even infinite. By relaxing deterministic guarantees and allowing controllers to fail (with low probability), we can achieve a principled robustness-performance trade-off. This relaxation and the information given by the distribution allow us to improve the expected control performance while retaining practically useful stability guarantees.

Gaussian processes (GPs) are a state-of-the-art method to learn probabilistic dynamics models that represent uncertainty as a distribution over functions. Even though the use of GPs to learn dynamics is gaining popularity in recent years, subsequent robust controller synthesis based on probabilistic models is less established. In their pioneering work in this direction, [Berkenkamp and Schoellig \(2015\)](#) propose a robust controller synthesis for a linearized GP dynamics model. Therein a worst-case robust synthesis method is presented where the uncertainty sets are derived from the GP posterior variance.

This paper extends prior work by designing controllers that use a linearized GP model’s probabilistic uncertainties. Instead of using the credible region as the bounds for the set of possible system parameters, we leverage the GP’s distribution to design efficient controllers with, in effect, the same robustness guarantees but superior performance. The proposed formulation results in a Bayesian optimal controller design that uses the GP’s probabilistic information. In particular, the desired properties are (i) probabilistic robustness (i.e., stable with high and predefined probability) and (ii) optimal in expectation with respect to (w.r.t.) the posterior distribution. In summary, the contributions of this paper are:

- a relaxation of the deterministic robust synthesis for GP error bounds proposed by [Berkenkamp and Schoellig \(2015\)](#) with reduced conservatism and improved control performance;
- a synthesis yielding a probabilistically robust LQR for a linearized GP dynamics model;
- probabilistic robustness guarantees for the resulting controller, where the desired confidence on the closed-loop stability can be chosen a-priori and is guaranteed by the algorithm.

The method describes an approximately optimal Bayesian robust stochastic control synthesis. In numerical experiments, we demonstrate improved control performance compared to the worst-case consideration proposed by [Berkenkamp and Schoellig \(2015\)](#).

### 1.1. Related work

While GPs are increasingly popular for learning dynamical systems ([Nguyen-Tuong and Peters, 2011](#); [Frigola et al., 2014](#); [Svensson et al., 2016](#); [Geist and Trimpe, 2020](#); [Buisson-Fenet et al., 2020](#)) and have recently been used in many branches of reinforcement learning ([Deisenroth and Rasmussen, 2011](#); [Doerr et al., 2017](#); [Vinogradska et al., 2018](#)) and model predictive control ([Ostafew et al., 2016](#); [Hewing et al., 2019](#); [Jain et al., 2018](#); [Koller et al., 2018](#)), only a few approaches exist for offline controller synthesis with formal stability guarantees. The approach in [Berkenkamp and Schoellig \(2015\)](#), as well as other works such as [Umlauft et al. \(2017\)](#), use probabilistic uncertainty to define error bounds, which are then treated in a worst-case fashion, while the underlying distribution is ignored. A key difference between these prior works and ours is the formulation of the objective: We demonstrate herein that leveraging the distribution yields better expected performance while retaining Bayesian stability guarantees similar to the deterministic robust synthesis. In practice, we gain a significant increase in performance at the cost of a small additional but pre-defined probability of being unstable.

[Umlauft et al. \(2018\)](#) propose to use a GP dynamics model to sample scenarios for a differential dynamic program. Their method is applicable to non-linear systems, using an iterative LQR for-

mulation, and includes probabilistic guarantees on the performance. In contrast to this approach, our method does not require solving the scenario program online but is limited to a single operating point in its current form. We derive explicit robustness properties for the linear, probabilistic setting.

Our synthesis method is primarily based on an LMI formulation for the upper bound on the LQR problem in the presence of probabilistic uncertainties proposed by [Umenberger and Schön \(2018\)](#). This formulation alleviates some of the conservativeness inherent in alternative worst-case formulations and thereby provides a synthesis for the LQR problem in the presence of probabilistic uncertainty. While the authors show empirically that the controller is robust w.r.t. the specified uncertainty, they do not give theoretical stability guarantees for the presented approach. We extend this work by a theoretical robustness analysis based on results from scenario optimization.

## 2. Problem formulation

Consider a stochastic, discrete-time and time-invariant dynamical system

$$x_{k+1} = f(x_k, u_k) + \omega_k, \quad (1)$$

and a locally valid linear approximation around an a-priori known operating point  $x^* = f(x^*, u^*)$

$$x_{k+1} \approx \bar{A}x_k + \bar{B}u_k + \omega_k, \quad (2)$$

where  $x_k \in \mathbb{R}^{d_x}$  is the state,  $u_k \in \mathbb{R}^{d_u}$  is the input and  $\omega_k \sim \mathcal{N}(0, \Sigma^\omega)$  is the process noise of the system. For notational convenience, we define the state-action tuple  $q_k \doteq (x_k, u_k)$  as well as the parameters of the linearized system as  $\bar{S} = [\bar{A} \ \bar{B}]$ . We assume access to the state  $x_k$  and the process noise is i.i.d. with  $\Sigma_\omega = \text{diag}(\sigma_{\omega,1}^2, \dots, \sigma_{\omega,d_x}^2)$ .

Designing controllers for a-priori known operating points that have a locally valid linear approximation of their dynamics is a problem often considered in practice, as well as in [Berkenkamp and Schoellig \(2015\)](#) and many industrial control processes.

### 2.1. Gaussian process dynamics model

The dynamics  $f$  and the parameters of the linear approximations  $\bar{A}$  and  $\bar{B}$  are unknown, but we can formulate a prior  $\hat{f}$  on (1) in the form of a GP. After observing a set  $D_N = \{q_k, x_{k+1}\}_{k=0}^N$  of one step transitions (usually as trajectories) of system (1), we compute the posterior for the dynamics  $f$  as well as the linearized dynamics  $\bar{A}$  and  $\bar{B}$ . We get a Bayesian estimate for the system matrices in (2) by taking the partial derivatives of the GP  $\hat{f}$ . The linearization of the GP dynamics model requires that the covariance function is at least twice differentiable at the operating point. Since GPs are closed under linear transformations, the resulting Jacobian at the point  $q^*$  is distributed according to a matrix normal distribution (cf. [Rasmussen and Williams \(2006, Chapter 9.4\)](#))  $\mathcal{MN}(M_S, U, V)$  where  $M_S$  is the mean and  $U$  and  $V$  are the covariances between rows and columns.<sup>1</sup> We denote the posterior probability distribution over system parameters as  $P_{\bar{S}}$  and its support as  $\Pi$ . For the synthesis problem, we define a truncated matrix normal distribution  $P_{\bar{S}}^c$  with the same mean and covariances, and its support  $\Pi_c$  with

$$\Pi_c = \{S \mid (\text{vec}(S) - \text{vec}(M_S))^T (U \otimes V)^{-1} (\text{vec}(S) - \text{vec}(M_S)) \leq \mathcal{X}_{d_s, c}^2\},$$

1. The matrix normal distribution and multivariate normal distribution are related by  $\text{vec}(\hat{S}) \sim \mathcal{N}(\text{vec}(M_S), U \otimes V)$ , where  $\text{vec}(M)$  is the (column-major) vectorization of  $M$  and  $\otimes$  is the Kronecker product.

where  $\chi_{d_s, c}^2$  is the  $c$ -quantile of the Chi-squared distribution with  $d_s = d_x^2 + d_x d_u$  degrees of freedom.

**Remark 1** *Learning a non-linear GP model for (1) instead of directly estimating a linear model for (2) using, for example, Bayesian linear regression, can be advantageous since the data  $D_N$  is collected from the non-linear system. The linear model will be biased if the data is collected outside the local region where (2) is a good approximation. Modeling  $f$  directly can avoid this bias.*

## 2.2. Bayesian optimal linear quadratic regulator

We want to design a linear and static state-feedback controller  $u_k = Kx_k$  for (2) that is optimal, in expectation, w.r.t. the truncated posterior distribution given by the GP model

$$\min_K J(K) = \int_{\Pi_c} J(K, S) P_S^c(S) dS, \quad (3.1)$$

$$J(K, S) = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{k=0}^T E_\omega [x_k^T Q x_k + u_k^T R u_k] \quad (3.2)$$

$$\text{s.t. } x_{k+1} = (A + BK)x_k + \omega_k, \quad S = [A \ B],$$

where  $Q$  and  $R$  are user defined, positive semi-definite and positive definite weight matrices respectively and  $x_0 = x^*$ . This formulation is different from the standard LQR problem with known  $A$  and  $B$  matrices. The case where uncertainty about  $\bar{A}$  and  $\bar{B}$  is small and can be ignored, certainty equivalence (CE), is investigated by Mania et al. (2019). Due to optimization of the *expected cost* w.r.t. the posterior belief the Bayesian optimal formulation may yield a different controller, even if the CE controller can stabilize all systems in  $\Pi_c$  (see Sec. 4).

In (3.2) we consider the (usual) expectation over the states and time, given a fixed system. Additionally, we take the expectation over model uncertainty (3.1). If (3.1) exists for some  $K$ , this  $K$  will almost surely stabilize any system in  $\Pi_c$ , since any unstable systems will have infinite expected cost. Conversely, if (3) is infeasible, there exists no  $K$  that is robustly stable.

**Assumption 1** *There exists at least one  $K$  such that (3.1) is finite on the support of  $P_S^c$ .*

Assumption 1 excludes any distributions containing systems that cannot be stabilized with the same controller. Distributions with unbounded support, such as the non-truncated normal distribution, are excluded since these will always contain unstable systems for every  $K$ . As a practical way of dealing with the unbounded support of a GP, we approximate  $P_S^c$  by a truncated distribution  $P_S^c$  over a bounded subset given by the credible interval  $c$ .

Depending on the prior distribution, likelihood, and data, the support of  $P_S^c$  might contain uncontrollable/unstabilizable systems. Non-controllable pairs of  $A, B$  are sets of measure zero w.r.t. the Lebesgue measure (Sontag, 1998, Prop. 3.3.12), meaning a sample of the GP linearization is almost surely controllable. Assumption 1 also excludes distributions that concentrate around an uncontrollable system as all of these systems are *hard* to control, i.e., the controllability matrices have small singular values, which will lead to exploding costs for every  $K$ .

## 3. Synthesis for linearized Gaussian processes

In this section, we introduce the synthesis algorithm by which we approximately optimize (3.1), and we prove a probabilistic stability property for the resulting controller.

### 3.1. Controller synthesis

To find a controller that minimizes the expected quadratic cost, it would be necessary to solve the integral in (3.1), for which no analytic solution exists (Umenberger and Schön, 2018). Umenberger and Schön (2018) developed a synthesis algorithm that optimizes a tight and convex upper bound for a Monte Carlo approximation to (3) by improving an initial common Lyapunov solution using a majorize-minimize algorithm.

Since the chosen GP model yields a point estimate for the process noise  $\Sigma^\omega$  we can slightly modify the LMI formulation of Umenberger and Schön (2018),

$$\begin{aligned}
 & \min_{L, Z \in \mathbb{S}, Y \in \mathbb{S}_{++}} \text{tr } Z \\
 \text{s.t. } & \begin{bmatrix} Y & * & * & * \\ A_i Y + B_i L & Y & * & * \\ Q^{1/2} Y & 0 & I & * \\ L & 0 & 0 & R^{-1} \end{bmatrix} \succcurlyeq 0 \\
 & \begin{bmatrix} Z & \Sigma^{\omega 1/2} \\ \Sigma^{\omega 1/2} & Y \end{bmatrix} \succcurlyeq 0 \\
 & \forall S_i = [A_i \ B_i] \in \Pi_c^M,
 \end{aligned} \quad (4)
 \end{aligned}$$

$$\begin{aligned}
 & \min_{K, X_i \in \mathbb{S}_{++}} \text{tr } \frac{1}{M} \sum_i X_i \Sigma^\omega \\
 \text{s.t. } & \begin{bmatrix} X_i - Q & * & * \\ A_i + B_i K & T_{\bar{X}_i}(X_i) & * \\ K & 0 & R^{-1} \end{bmatrix} \succcurlyeq 0 \\
 & \forall S_i = [A_i \ B_i] \in \Pi_c^M,
 \end{aligned} \quad (5)$$

where  $\mathbb{S}$  and  $\mathbb{S}_{++}$  denote the set of symmetric or positive definite matrices. For the initial solution (4)  $X = Y^{-1}$ ,  $K = LX$ , and  $Z$  is a slack variable. For the improvement step (5)  $T_{\bar{X}_i}(X_i)$  denotes the linear approximation of  $X_i^{-1}$  around  $\bar{X}_i$ , the solution to (5) for the previous iteration.

Let  $K^*$  denote the solution of (5) after a stopping criterion has been fulfilled. We are interested in the probability that  $K^*$  will stabilize the true linear system  $x_k = (\bar{A} + \bar{B}K^*)x_k + w_k$ . As a first step we establish an *a-priori* bound on the probability that the controller result of the initialization step (4)  $K_{\text{init}} = LX$  will be stable w.r.t. to the set of probable systems  $\Pi_c$ . We use the probabilistic knowledge about the system parameters to express the system's stability as a Bernoulli distribution. We define a stability indicator variable for the true linear system (2) and a fixed  $K$  as

$$\phi_K = \begin{cases} 1, & \text{if } \rho(\bar{A} + \bar{B}K) < 1 \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Further, we define the probabilistic robustness property as  $V_K(\epsilon) = (\mathbb{E}[\phi_K] \geq 1 - \epsilon)$ . By this definition, we accept a (small) probability  $\epsilon$  that the feedback control destabilizes the system.

**Theorem 1** *Let Assumption 1 hold and let  $K_{\text{init}}$  be the solution to (4). Let  $\epsilon$  and  $\beta > 0$  be given and  $n_k = 2d_x^2 + d_x d_u$  the number of optimization variables in (4). Let  $P_S^c$  be the posterior probability over  $S = [\bar{A} \ \bar{B}]$ . If we draw  $M \geq \left\lceil \frac{2}{\epsilon} \ln\left(\frac{1}{\beta}\right) + n_k \right\rceil$  samples according to  $P_S^c$ , then it holds that*

$$P(V_{K_{\text{init}}}(\epsilon)) > 1 - \beta. \quad (7)$$

**Proof** The optimization problem (4) is a convex scenario program and therefore we can apply Campi et al. (2009, Theorem 1). ■

Theorem 1 states that, according to our current (truncated) belief over the system parameters, if we sample  $M$  scenarios from  $P_S^c$ , we can be *confident* that with probability of at least  $1 - \beta$  the

desired robustness property  $V_{K_{\text{init}}}$  is fulfilled. Part of the robustness property is a *risk* parameter that describes the upper bound for the probability that  $K_{\text{init}}$  will destabilize the system.

Since  $K_{\text{init}}$  is highly conservative due to the common Lyapunov approximation, we improve the controller by iteratively solving (5). Due to the averaging over all solutions, (5) is not a scenario program and Theorem 1 does not hold for the improvements. We therefore validate the final controller in an a-posteriori analysis which we describe in the next subsection.

### 3.2. Controller validation

After iteratively solving (5) for a particular set of samples, we validate the success of synthesis, i.e., that the controller will stabilize (2) with high probability, *before* deploying the controller to the system. The validation ensures that the robustness properties that provably hold for the initialization (4) will also hold after the iterative improvement. At the same time we can improve our initial guarantees regarding the probability of a successful synthesis  $P(V_{K^*}(\epsilon))$ .

The validation is done on the learned distribution but for samples not seen by the synthesis method. For the validation procedure, we approximate the expectation  $\mathbb{E}[\phi_{K^*}]$  by sampling systems from  $P_{\bar{S}}^c$ . The sample-based approach avoids conservativeness in our analysis at the cost of some approximation error. Since sampling and checking the spectral radius are computationally relatively cheap, we can use a lot of samples and make this approximation error small enough for most practical purposes. Sample-based robust analysis is a simple method that does not rely on a specific uncertainty structure (cf. Ray and Stengel (1993)).

**Theorem 2** *Let Assumption 1 hold. If we draw  $M_{\text{val}} \geq \frac{1}{2\epsilon_{\text{val}}} \log(\frac{1}{\alpha})$  samples from  $P_{\bar{S}}^c$ , Algorithm 1 will return a probabilistically stable controller  $K^*$  where*

$$P(\mathbb{E}[\phi_{K^*}] \geq 1 - \epsilon_{\text{pr}}) > 1 - \alpha, \quad (8)$$

with  $\epsilon_{\text{pr}} = c - (\epsilon + \epsilon_{\text{val}})$  and according to the posterior distribution  $P_{\bar{S}}$ .

**Proof** Algorithm 1 returns a controller only if the empirical expected value of  $\phi$ ,  $\hat{\phi}$  is at least  $1 - \epsilon$ . We can directly apply the one sided Hoeffding inequality  $P(\hat{\phi} - \mathbb{E}[\phi] \geq \epsilon_{\text{val}}) \leq \exp(-2M_{\text{val}}\epsilon_{\text{val}}^2)$ . By choosing  $M_{\text{val}} \geq \frac{1}{2\epsilon_{\text{val}}} \log(\frac{1}{\alpha})$  it follows that  $\exp(-2M_{\text{val}}\epsilon_{\text{val}}^2) \leq \alpha$  and  $\hat{\phi} \geq 1 - \epsilon$ , thus

$$P(1 - \epsilon - \mathbb{E}[\phi] \geq \epsilon_{\text{val}}) \leq \alpha \Rightarrow P(\mathbb{E}[\phi] \leq 1 - \epsilon - \epsilon_{\text{val}}) \leq \alpha. \quad (9)$$

Since we sample scenarios from the truncated distribution  $P_{\bar{S}}^c$  as opposed to  $P_{\bar{S}}$ , we do not account for systems in a set with probability mass  $1 - c$ . It follows that  $P(\mathbb{E}[\phi] \leq \epsilon_{\text{pr}}) \leq \alpha$ . ■

If Assumption 1 does not hold, Algorithm 1 will return no controller with high probability, giving a certificate of infeasibility for the problem approximated by the scenario program. In this case, we can either improve the prior distribution by adding additional knowledge or collect more data. The choice of  $\beta$  in Algorithm 1 does not influence the a-posteriori stability, but it determines how often we need to run the synthesis on a new sample set before we find a controller that satisfies  $\hat{\phi} \geq 1 - \epsilon$ . Consequently choosing a high  $\beta$  will increase the expected runtime of Algorithm 1.

When compared to the synthesis by Berkenkamp and Schoellig (2015) the problem is relaxed since we allow a (small) probability  $\alpha$  that the controller does not robustly stabilize the credible region of the model. Otherwise, we can achieve the same stability guarantees by setting  $\epsilon_{\text{pr}} = 95\%$ , the credible region chosen therein. Keeping the similar robustness properties in mind, we now compare both approaches in the next section.

---

**Algorithm 1** Probabilistic robust LQR synthesis
 

---

- 1: **Input:** Posterior distribution  $P_{\hat{S}}^c$  of  $\hat{S}$ , risk parameter  $\epsilon$  and  $\epsilon_{\text{val}}$ , confidence parameter  $\beta$  and  $\alpha$ , cost matrices  $Q$  and  $R$ .
  - 2: **do**
  - 3:     Generate at least  $M = \left\lceil \frac{2}{\epsilon} \ln\left(\frac{1}{\beta}\right) + n_k \right\rceil$  samples from  $P_{\hat{S}}^c$  (Theorem 1).
  - 4:     Find  $K^*$  by solving first (4) and then (5) iteratively.
  - 5:     **if** (4) is infeasible **return**
  - 6:     Draw at least  $M_{\text{val}} = \exp\left(\frac{1}{2\epsilon_{\text{val}}} \log\left(\frac{1}{\alpha}\right)\right)$  samples from  $P_{\hat{S}}^c$ . (Theorem 2)
  - 7:      $\hat{\phi}_{K^*} = \frac{1}{M_{\text{val}}} \sum_{i=0}^{M_{\text{val}}} \begin{cases} 1, & \text{if } \rho(A_i + B_i K^*) < 1 \\ 0, & \text{otherwise} \end{cases}$
  - 8:     **while**  $\hat{\phi}_{K^*} < 1 - \epsilon$
  - 9:     **return**  $K^*$ .
- 

## 4. Empirical results

In this section, we apply the results from Sec. 3 to a synthetic benchmark problem to illustrate and empirically verify the performance of the proposed synthesis and its robustness properties. We compare performance and robustness of the proposed *probabilistic robust* (PR) synthesis described in Algorithm 1 with the *robust* (R) method to solve the LQR problem based on linearized GP models in Berkenkamp and Schoellig (2015). Additionally, we compare against the *certainty equivalence* (CE) setting in which the mean of  $P_{\hat{S}}$  is used to solve the standard Riccati equations.

In the second subsection, we use GPs to learn a given system from data and synthesize a controller based on the distribution over linearized systems. We numerically compare the stability and performance properties of the discussed methods. We want to remark that the primary goal here is not to evaluate how well GPs learn the given dynamics. Instead, our focus is to compare how different approaches use the given uncertainty and how the resulting controller performs.

We collect the confidence and risk parameters for Algorithm 1 in the tuple  $\psi = (c, \epsilon, \beta, \epsilon_{\text{val}}, \alpha)$ . The source code for all experiments, including all parameters, is available online<sup>2</sup>.

### 4.1. Empirical results on a synthetic distribution

As a benchmark problem, we create a synthetic distribution  $P_{\hat{S}}$  as  $\mathcal{MN}([\bar{A} \ \bar{B}], U, V)$  that highlights the benefits of the proposed method. As the covariance matrices we choose a sample  $E = U \otimes V$  from a Wishart distribution with  $(d_x^2 + d_x d_u)$  degrees of freedom and a scale matrix  $\sigma^2(0.5 \cdot I_{18} + 0.5 \cdot 1_{18})$ , where  $1_{18}$  denotes a  $18 \times 18$  all-ones matrix. This allows for correlations in the uncertainty between system parameters. By changing  $\sigma^2$  we can compare the synthesis methods over multiple levels of uncertainty.

As the mean for the synthetic distribution we choose the LQR benchmark problem proposed by Dean et al. (2020)

$$\bar{A} = \begin{bmatrix} 1.01 & 0.01 & 0 \\ 0.01 & 1.01 & 0.01 \\ 0 & 0.01 & 1.01 \end{bmatrix} \quad \bar{B} = I_3 \quad Q = 10^{-3} I_3 \quad R = I_3 \quad \Sigma^\omega = 10^{-3} I_3, \quad (10)$$

---

2. <https://github.com/Data-Science-in-Mechanical-Engineering/prlqr>

where  $I_3$  denotes the 3-dimensional identity matrix. The credible region for the robust synthesis is chosen as 95% and the parameters for the proposed method are set correspondingly to  $\psi = (0.98, 0.02, 0.20, 0.01, 0.001)$ . While the distribution of risk over the parameters  $\epsilon_{\text{pr}} = c - (\epsilon + \epsilon_{\text{val}})$  is arbitrary, it does influence the numbers of samples for the scenario program as well as the validation. In practice, we set the parameters via trial and error, such that the run-time of the algorithm is reasonable. Stability and performance are evaluated on a sample of 10.000 systems drawn from the *non-truncated* matrix normal distribution. We calculate the expected cost for each sampled system over a finite horizon of 200 (cf. Schlüter et al. (2020, Eq. 20)). We report the cost for all sampled and stable closed-loop systems as well as the frequency of unstable systems in Fig. 1.

The empirical results clearly show a reduced level of conservativeness in the problem formulation when comparing the proposed PR synthesis to the R synthesis. The PR synthesis problem is feasible with higher levels of uncertainty in the model and, additionally, the *expected cost* of the resulting controller is lower for all levels of uncertainty. The trade-off is that the PR synthesis yielding an unstable controller for up to 0.3% of samples of the *non-truncated* distribution. As to be expected, having the different optimization goals in mind, the worst case costs for the controller of the PR synthesis is higher, but only for high levels of uncertainty and in rare cases. Similar empirical results for the PR synthesis have been shown previously by Umenberger and Schön (2018), albeit without sample requirements for probabilistic robustness guarantees. In contrast to the previous results we increased the sample size to  $M \geq 188$  according to Theorem 1 from  $M = 50$  used by Umenberger and Schön (2018).

Compared to the robust synthesis methods, the CE synthesis yields a controller that stabilizes at least 95% of the samples for very low uncertainties in the system parameters. It can, therefore, not be used in settings with significant uncertainties where we require probabilistic performance guarantees. Even at low model uncertainty, the cost’s variance of the CE controller remains high, which might be undesirable in practice. Starting from a  $\sigma^2$  of  $1e-4$ , the controller resulting from the PR synthesis is able to outperform the CE controller in expectation. This suggests that robustness properties do not need to come at the cost of performance in the considered setting with probabilistic uncertainties. To summarize the findings of Fig. 1: the proposed Algorithm 1 outperforms the robust controller in expectation, has higher worst-case costs only in very unlikely cases, and can find controllers with significantly higher model uncertainty.

## 4.2. Synthesis using GP models trained in simulation

In this subsection, we apply our framework to stabilize two non-linear systems, namely the *Furuta* or *rotary pendulum* and a *synthetic* system inspired by the problem proposed in Dean et al. (2020). As a comparison baseline, we show the controller based on the linearization of the true system (T). We add this comparison to show how the learning-based methods perform versus *perfect model knowledge*. The parameters  $\psi$  for the PR synthesis are identical to the previous subsection with  $\psi = (0.98, 0.02, 0.20, 0.01, 0.001)$  and the credible region for the robust method is again set to 95%. We use a standard GP with a squared exponential kernel and fixed hyperparameters for all our experiments. A dataset  $D_N$  for the GP model is generated by starting each system close to the operating point and sampling random control inputs from a normal distribution. The system is reset after  $n$  samples, and  $N_t$  rollouts are recorded, giving a total sample size of  $N = nN_t$ . We evaluate the *performance* by applying the controller to the true non-linear system, initialized in the operating point, and record the cost for 200 time steps. We repeat the performance measurement 200 times and report the mean of these runs as the cost for the given state feedback controller. Each

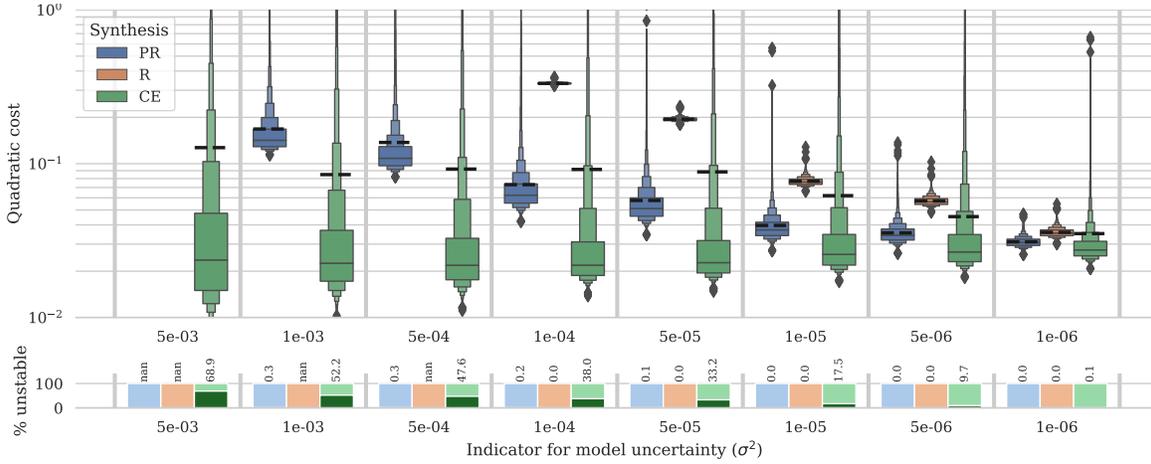


Figure 1: Synthetic example (Sec. 4.1). Distribution over LQR costs (top) and frequency of unstable closed-loop systems (bottom) for a model with different levels of uncertainties based on 10,000 samples. A missing boxplot indicates the problem was infeasible for the given uncertainty. The empirical mean of each distribution is shown as a black dashed line.

experiment, i.e., data collection, GP regression, and controller synthesis, is repeated with a different random seed 25 times, and the results for each experiment are plotted.

#### 4.2.1. SYNTHETIC SYSTEM

The locally linear synthetic system is a variant of the linear system proposed as a challenging LQR benchmark by [Dean et al. \(2020\)](#)

$$x_{k+1} = \begin{bmatrix} 1.01 & 0.01 & 0 \\ 0.01 & 1.01 & 0.01 \\ 0 & 0.01 & 1.01 \end{bmatrix} x_k + \left( \begin{bmatrix} 0.3 & 0.0 & 0.0 \\ 0.3 & 0.3 & 0.0 \\ 0.3 & 0.3 & 0.3 \end{bmatrix} x_k \right)^{\circ 3} + Iu_k + \omega_k, \quad (11)$$

where  $\circ$  denotes the Hadamard power. We set  $Q = I_3$  and  $R = I_3$ . One rollout for this system consists of six samples. Due to the added non-linearity, a linear estimator might fail to identify the system. We evaluate the stability by checking the spectral radius w.r.t. the linear part of (11) and check that all states are smaller than 1000. The results are shown in Fig. 2(a). The PR synthesis outperforms the R synthesis in expectation and is feasible more often at 3 and 5 rollouts. All feasible cases lead to a stable controller. In this experiment, the CE controller performs better than the robust methods, suggesting that our GP model might overestimate the uncertainty. For 8 rollouts the PR and CE controller are on par with a controller designed for the linearization of the true non-linear system (T).

#### 4.2.2. ROTARY PENDULUM

The rotary pendulum dynamics are based on the equations of motion derived by [Iwashiro et al. \(1996\)](#) and discretized with a sample time of 20 ms. The state-space of the rotary pendulum is

4-dimensional and consists of the angle and angular velocity of the rotary arm and the pendulum. The operating point is the unstable upright position. We collected 30 samples per rollout. Since our contribution is not focused on model learning, we reduced the difficulty of the learning task. The data is recorded around the operating point using an a-priori stabilizing controller, which allowed us to have a relatively accurate GP posterior with comparatively little data.

The results are shown in Fig. 2(b). The PR synthesis yields a controller that performs better, on average, than both the controllers from the CE and R synthesis. In this setting, the PR controller performs better than the CE controller, which we attribute to the more informative data-set and a better tuned GP model, in contrast to the synthetic system.

We want to remark that if we use a Bayesian uncertainty estimate, the performance critically depends on the chosen prior, especially in the low data regime. However, the robustness properties hold as long as the prior contains the true system with sufficient probability. In principle, the same results apply to uncertainties obtained from frequentist methods.

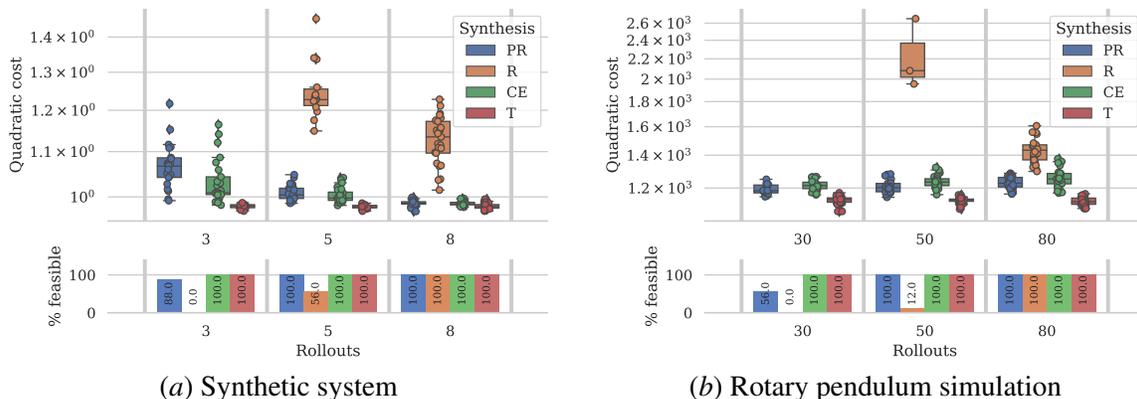


Figure 2: Synthesis on GP model learned from data Sec. 4.2. Distribution over (stable) LQR costs (top) on the true system. We plot the frequency of feasibility for each method (bottom). All feasible problems resulted in stable controllers. We conducted 25 experiments for each number of rollouts.

## 5. Conclusion

We introduce a novel probabilistic robust control method for linearized GP models that leverages posterior distributions over model parameters. Instead of optimizing for the worst-case, we optimize the controller for the specific distribution. Thus, we achieve significantly better performance while still ensuring comparable robustness guarantees.

From a theoretical point of view, our method only requires a posterior distribution over potential system parameters. Here, we obtain these by linearizing a GP, a model class with a lot of recent attention for dynamics modeling. While this is already relevant for many applications, we want to address more general problems, also beyond set-point control, in future work. Additional research is required to use the framework of probabilistic controller synthesis for partial state observations, where one would like to design an observer alongside the controller, both based on data.

## Acknowledgments

The authors thank F. Solowjow, C. Fiedler, A. R. Geist and S. Heim for their helpful comments and discussions. This work was supported in part by the Cyber Valley Initiative and the Max Planck Society. The authors thank the International Max Planck Research School for Intelligent Systems for supporting A. von Rohr.

## References

- F. Berkenkamp and A. P. Schoellig. Safe and robust learning control with gaussian processes. In *European Control Conference*, pages 2496–2501, 2015.
- B. Boulet and Y. Duan. The fundamental tradeoff between performance and robustness. *IEEE Control Systems Magazine*, 27(3):30–44, June 2007.
- Mona Buisson-Fenet, Friedrich Solowjow, and Sebastian Trimpe. Actively learning gaussian process dynamics. *Proceedings of Machine Learning Research*, pages 5–15. PMLR, 10–11 Jun 2020.
- Marco C. Campi, Simone Garatti, and Maria Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149 – 157, 2009. ISSN 1367-5788.
- Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, pages 633–679, 2020.
- MP. Deisenroth and CE. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Int. Conf. on Machine Learning*,, pages 465–472, 2011.
- Andreas Doerr, Christian Daniel, Duy Nguyen-Tuong, Alonso Marco, Stefan Schaal, Marc Toussaint, and Sebastian Trimpe. Optimizing long-term predictions for model-based policy search. In *Conference on Robot Learning*, pages 227–238, November 2017.
- Roger Frigola, Yutian Chen, and Carl Edward Rasmussen. Variational gaussian process state-space models. In *Advances in neural information processing systems*, pages 3680–3688, 2014.
- Andreas Geist and Sebastian Trimpe. Learning constrained dynamics with gauss’ principle adhering gaussian processes. *Proceedings of Machine Learning Research*, pages 225–234. PMLR, 10–11 Jun 2020.
- L. Hewing, J. Kabzan, and M. N. Zeilinger. Cautious model predictive control using gaussian process regression. *IEEE Transactions on Control Systems Technology*, pages 1–8, 2019.
- M. Iwashiro, K. Furuta, and K. J. Astrom. Energy based control of pendulum. In *IEEE International Conference on Control Applications*, pages 715–720, 1996.
- Achin Jain, Truong Nghiem, Manfred Morari, and Rahul Mangharam. Learning and control using gaussian processes. In *IEEE Int. Conf. on Cyber-Physical Systems*, pages 140–149, 2018.

- Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration. In *IEEE Conference on Decision and Control*, pages 6059–6066, 2018.
- Horia Mania, Stephen Tu, and Benjamin Recht. Certainty equivalence is efficient for linear quadratic control. In *Advances in Neural Information Processing Systems*, pages 10154–10164, 2019.
- Duy Nguyen-Tuong and Jan Peters. Model learning for robot control: a survey. *Cognitive processing*, pages 319–340, 2011.
- Chris J Ostafew, Angela P Schoellig, Timothy D Barfoot, and Jack Collier. Learning-based non-linear model predictive control to improve vision-based mobile robot path tracking. *Journal of Field Robotics*, pages 133–152, 2016.
- CE. Rasmussen and CKI. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- Laura Ryan Ray and Robert F. Stengel. A monte carlo approach to the analysis of control system robustness. *Automatica*, pages 229 – 236, 1993.
- H. Schlüter, F. Solowjow, and S. Trimpe. Event-triggered learning for linear quadratic control. *IEEE Transactions on Automatic Control*, 2020.
- Eduardo D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems (2nd Ed.)*. Springer-Verlag, Berlin, Heidelberg, 1998.
- Andreas Svensson, Arno Solin, Simo Särkkä, and Thomas Schön. Computationally efficient bayesian learning of gaussian process state space models. In *Artificial Intelligence and Statistics*, pages 213–221, 2016.
- Jack Umenberger and Thomas B Schön. Learning convex bounds for linear quadratic control policy synthesis. In *Advances in Neural Information Processing Systems*, pages 9561–9572, 2018.
- J. Umlauf, T. Beckers, M. Kimmel, and S. Hirche. Feedback linearization using gaussian processes. In *IEEE Conference on Decision and Control*, pages 5249–5255, 2017.
- J. Umlauf, T. Beckers, and S. Hirche. Scenario-based optimal control for gaussian process state space models. In *European Control Conference*, pages 1386–1392, 2018.
- Julia Vinogradska, Bastian Bischoff, Jan Achterhold, Torsten Koller, and Jan Peters. Numerical quadrature for probabilistic policy search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 164–175, 2018.
- Kemin Zhou, John Comstock Doyle, Keith Glover, et al. *Robust and optimal control*, volume 40. Prentice hall New Jersey, 1996.