Learning a Stability Filter for Uncertain Differentially Flat Systems using Gaussian Processes

Melissa Greeff, Adam W. Hall and Angela P. Schoellig

***Abstract*— Many physical system models exhibit a structural property known as differential flatness. Intuitively, differential flatness allows us to separate the system's nonlinear dynamics into a linear dynamics component and a nonlinear term. In this work, we exploit this structure and propose using a nonparametric Gaussian Process (GP) to learn the unknown nonlinear term. We use this GP in an optimization problem to optimize for an input that is most likely to feedback linearize the system (i.e., cancel this nonlinear term). This optimization is subject to input constraints and a stability filter, described by an uncertain Control Lyapunov Function (CLF), which probabilistically guarantees exponential trajectory tracking when possible. Furthermore, for systems that are control-affine, we choose to express this structure in the selection of the kernel for the GP. By exploiting this selection, we show that the optimization problem is not only convex but can be efficiently solved as a second-order cone program. We compare our approach to related works in simulation and show that we can achieve similar performance at much lower computational cost.**

I. INTRODUCTION

There is a growing interest in increased autonomy of safety-critical but uncertain and nonlinear systems, such as self-driving vehicles, unmanned aerial vehicles (UAVs), and mobile manipulators. This has motivated bridging formal safety analysis with the flexibility of machine learning to cope with large prior uncertainties.

Gaussian Processes (GPs) have gained popularity within the control community as a nonparametric machine learning approach that quantifies the uncertainty in its prediction. This uncertainty can be used to generate a probabilistic upper bound for the difference between the true and learned function value based on distance to the training data [1].

GPs are often used to learn the dynamics model which is then included in a model predictive control (MPC) framework. The GP uncertainty quantification can be used to tighten state and input constraints [2], [3]. A major limitation is that the resulting optimization problem is generally non-convex, and slow and expensive to solve. Furthermore, this approach provides no stability guarantees for the controlled system.

Stability guarantees, for example, asymptotic stability or tracking, have been combined with GPs by exploiting two structural assumptions about the dynamics: 1) the system is control-affine and 2) the system is either fully actuated [4] or differentially flat [5]. Intuitively, differential flatness allows us to separate the nonlinear model into a linear dynamics component and a nonlinear term, see Fig. 1 [6]. Given that these assumptions (control-affine and differential flatness) are true for many first-principle models of physical systems, for example, quadrotors [8], cranes, and manipulators, they are not limiting in practice.

The authors are with the Dynamic Systems Lab (www.dynsyslab.org) at the University of Toronto Institute for Aerospace Studies (UTIAS) and the Vector Institute for Artificial Intelligence, Toronto. Adam W. Hall is jointly with the Space and Terrestrial Autonomous Robotic Systems Laboratory. Email: melissa.greeff@mail.utoronto.ca, adam.hall@robotics.utias.utoronto.ca, schoellig@utias.utoronto.ca

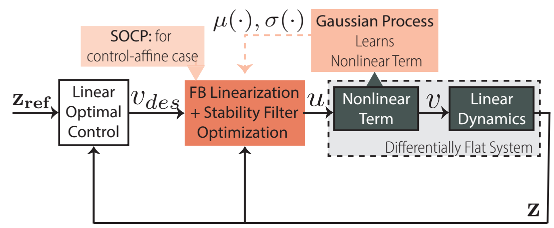


Fig. 1. Overview of the proposed learning-based control architecture. Our proposed approach learns the nonlinear term as a Gaussian Process (GP). We combine feedback linearization with a stability filter by using the learned GP model to 1) optimize for an input u that is most likely to feedback linearize the system (i.e., cancel the nonlinear term) 2) guarantee stability with high probability through a stability filter. For systems that are control-affine, we show that the resulting optimization is a convex second-order cone program (SOCP).

However, previous work typically includes one of two additional limiting assumptions: 1) the actuation function is fully known [4], [9] (i.e., the unknown disturbance is only a function of the state) or 2) there are no actuation or input constraints [5], [7]. Practically, there is often uncertainty in the actuation function as a result of delays, error in the system parameters (e.g., mass or inertia), or unaccounted dynamics of low-level controllers. Input constraints can represent physical limitations or may be added for user safety.

Differential flatness is commonly used in feedback (FB) linearization controllers, which attempt to cancel the nonlinear term such that outer-loop linear controllers can be designed based on the linear dynamics alone. In [5] and [7], the nonlinear term (or inverse nonlinear term) is learnt as a GP. The predicted GP mean is used to update the inverse nonlinear term (or FB linearization) while the uncertainty is used in a robust outer-loop linear controller. In [5], the control-affine structure is leveraged when taking the derivative of the learnt GP. In [7], similar to the approach we propose here, the control-affine structure is exploited in the selection of the GP kernel structure. Both approaches can guarantee asymptotic trajectory tracking. However, because the robustness is accounted for in the outer-loop controller, it is difficult to account for actuation or input constraints and, therefore, they are often neglected.

Using the strong assumption that the actuation function is known, in [4], a Lyapunov function is proposed that maximizes the probability of asymptotic stability of a known equilibrium accounting for input constraints. The approach is limited to fully actuated systems and can only be applied to stabilizing a known equilibrium. Another approach that makes this assumption, [7], combines a known Control Lyapunov Function (CLF), to encourage stability, with input constraints in an optimization framework that can be solved as a quadratic program (QP). Control Lyapunov Functions (CLFs) have traditionally been used in minimum input-norm controllers to stabilize an equilibrium [10].

Motivated by the work in [11], our proposed approach uses the idea that, for systems that are FB linearizable, we can exploit their linear tracking error dynamics to construct a Lyapunov function to ensure exponential tracking convergence. Similar to [11], we propose using such a Lyapunov function as a CLF. However, unlike [11], we do not need to learn in an episodic fashion. Instead, we propose using a GP, with carefully selected kernel structure, to learn the nonlinear term and then leverage its quantified uncertainty in our controller.

In this paper, we develop a method to efficiently handle robust tracking guarantees and input constraints in the presence of model uncertainty. As illustrated in Fig. 1, our proposed approach uses the GP to learn the uncertain nonlinear term and combines feedback linearization, commonly applied to differentially flat systems, with a stability filter, described by a CLF, in an optimization framework. The three key contributions of this work are:

- We provide a novel approach that uses a GP to learn the uncertain nonlinear term and then use the GP in an optimization problem to optimize for a control input u that is 1) most likely to cancel the nonlinear term while 2) guaranteeing a stability filter condition with high probability and 3) adhering to input constraints.
- We show that for control-affine systems, by exploiting this structure in the GP kernel selection, the resulting optimization is not only convex but can be solved efficiently as a second-order cone program (SOCP).
- We demonstrate, in simulation, a significant reduction in average computation over previous robustness methods using GPs, while still achieving high tracking performance. This makes our proposed approach suitable for online and onboard implementation in high-rate feedback loops, for example, on autonomous UAVs.

II. PROBLEM STATEMENT

Consider a single-input, single-output (SISO), control-affine system with state $\mathbf{x} \in \mathbb{R}^n$ and input $u \in \mathbb{R}$:

$$\dot{\mathbf{x}} = f(\mathbf{x}, u), \tag{1}$$

where $f(\mathbf{x}, u)$ is an *unknown* function and the dimension of the state n is known. The input is subject to bound constraints, that is, $u_{min} \leq u \leq u_{max}$ where u_{min} and u_{max} are known minimum and maximum input constraints.

Assumption 1: The system (1) is differentially flat in the *known* output $y = h(\mathbf{x}) \in \mathbb{R}$.

Under Assumption 1, our goal is to design a control law u that:

O1 Achieves high trajectory tracking performance;
O2 Can be efficiently computed online;
O3 Guarantees that the closed-loop system satisfies robust stability (in the sense that tracking errors are bounded);
O4 Accounts for actuation/input constraints.

III. BACKGROUND

A. Differential Flatness

Definition 1 (Differential Flatness [6]): A SISO nonlinear system (1) is differentially flat in output y if there exist smooth, invertible functions such that: $\mathbf{x} = \phi(\mathbf{z})$, $u = \psi^{-1}(\mathbf{z}, v)$, where $\mathbf{z} = [y, \dot{y}, ..., y^{(n-1)}]^T$, $v = y^{(n)}$.

Lemma 1: If system (1) is differentially flat in output y, then it is equivalent to:

$$v = \psi(\mathbf{z}, u), \tag{2}$$

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}v, \tag{3}$$

where the linear dynamics (3) are an integrator chain of degree n and the nonlinear term is given by (2) [5], [7]. Moreover, if (1) is *control-affine*, the nonlinear term (2) is also *control-affine*, that is, (2) can be written as:

$$v = \alpha(\mathbf{z}) + \beta(\mathbf{z})u. \tag{4}$$

Note that since our dynamics $f(\mathbf{x}, u)$ is unknown, the linear dynamics (3) can be recovered using Assumption 1 but the nonlinear term (2) is still unknown.

B. Control Lyapunov Functions

When the nonlinear term (2) is known, we can exploit the structure of differentially flat systems (2)-(3) to construct a Control Lyapunov Function (CLF). We will use this CLF as a stability filter when the nonlinear term is uncertain.

Using standard feedback linearization, we can design the controller u to cancel the nonlinear term (2) and then select the input to the linear dynamics v such that the resulting linear error dynamics are Hurwitz. Consider a smooth reference state $z_{ref}(t)$ and reference input $v_{ref}(t)$, we can write the tracking error dynamics of (2)-(3) as:

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{z} + \mathbf{B}\psi(\mathbf{z}, u) - \dot{\mathbf{z}}_{ref}, \tag{5}$$

where the tracking error is $\mathbf{e} = \mathbf{z} - \mathbf{z}_{ref}$ and $\dot{\mathbf{z}}_{ref} = [\dot{y}_{ref}, ..., y_{ref}^{(n-1)}, v_{ref}]^T$.

Consider a standard feedback linearization controller $u = \psi^{-1}(\mathbf{z}, v_{nom})$ where

$$v_{nom} = -\tilde{\mathbf{K}}\mathbf{e} + v_{ref} \tag{6}$$

is designed such that the closed-loop error dynamics are stable. Under this control law, it follows that the error dynamics in (5) simplifies to:

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{B}\tilde{\mathbf{K}})\mathbf{e},$$

where $\tilde{\mathbf{K}}$ is selected such that $(\mathbf{A} - \mathbf{B}\tilde{\mathbf{K}})$ is Hurwitz.

Given the Hurwitz stability of the closed-loop system it follows from converse stability theorems that we can construct a Lyapunov function that guarantees exponential tracking convergence [11]. Specifically, we can select the Lyapunov function $V(\mathbf{e}) = \mathbf{e}^T\mathbf{P}\mathbf{e}$ where $\mathbf{P} > 0$ is a positive definite matrix that satisfies the algebraic Riccati equation:

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = \mathbf{0},$$

for selected positive definite matrices $\mathbf{Q} > 0$, $\mathbf{R} > 0$.

Definition 2: A function $V : \mathbb{R}^n \to \mathbb{R}_+$ is a Control Lyapunov Function (CLF) for (5) certifying exponential stability if there exists positive constants $c_1, c_2, c_3 > 0$ such that:

$$c_1||\mathbf{e}||^2 \leq V(\mathbf{e}) \leq c_2||\mathbf{e}||^2,$$
$$\dot{V}(\mathbf{e}) \leq -c_3 V(\mathbf{e}).$$

We can use the previously constructed Lyapunov function $V(\mathbf{e}) = \mathbf{e}^T\mathbf{P}\mathbf{e}$ as a CLF with positive constants $c_1 = \lambda_{min}(\mathbf{P}), c_2 = \lambda_{max}(\mathbf{P})$ and $c_3 = \frac{\lambda_{min}(\mathbf{S})}{\lambda_{max}(\mathbf{P})}$ where $\mathbf{S} = \mathbf{Q} + \tilde{\mathbf{K}}^T\mathbf{R}\tilde{\mathbf{K}}$. In the absence of a chosen controller u, the decreasing time derivative condition $\dot{V}(\mathbf{e}) \leq -c_3V(\mathbf{e})$ in Definition 2 becomes:

$$\mathbf{e}^T\mathbf{P}(\mathbf{A}\mathbf{z} + \mathbf{B}\psi(\mathbf{z}, u) - \dot{\mathbf{z}}_{ref}) \leq -c_3\mathbf{e}^T\mathbf{P}\mathbf{e}. \quad (7)$$

C. Gaussian Processes (GPs)

GP regression is a nonparametric approach that is used to approximate a nonlinear map, $\psi(\mathbf{a}) : \mathbb{R}^{\dim(\mathbf{a})} \to \mathbb{R}$, from the input $\mathbf{a}$ to the function value $\psi(\mathbf{a})$. It does this by assuming that the function values $\psi(\mathbf{a})$, associated with different inputs $\mathbf{a}$, are random variables and that any finite number of these random variables have a joint Gaussian distribution. This nonparametric approach still requires us to define two priors: a prior mean function of $\psi(\mathbf{a})$, generally set to zero, and a covariance or kernel function $k(\cdot,\cdot)$ which encodes, for two input points, how similar their respective function values are. For example, a common kernel function is the squared-exponential (SE) function:

$$k(\mathbf{a}_i, \mathbf{a}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{a}_i - \mathbf{a}_j)^T\mathbf{L}^{-2}(\mathbf{a}_i - \mathbf{a}_j)\right) + \delta_{ij}\sigma_\eta^2,$$

which is characterized by three types of hyperparameters: the prior variance σ_f^2, observation noise σ_η^2, where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise, and the length scales, or the diagonal elements of the diagonal matrix $\mathbf{L}$, which encode a measure of how quickly the function $\psi(\mathbf{a})$ changes with respect to $\mathbf{a}$. These hyperparameters can be optimized by solving a maximum log-likelihood problem [12].

This GP framework can be used to predict the function value at any query point $\mathbf{a}$ based on N noisy observations, $\mathcal{D} = \{\mathbf{a}_i, \hat{\psi}(\mathbf{a}_i)\}_{i=1}^N$. The predicted mean and variance at the query point $\mathbf{a}$ conditioned on the observed data $\mathcal{D}$ are [12]:

$$\mu(\mathbf{a}) = \mathbf{k}(\mathbf{a})\mathbf{K}^{-1}\hat{\mathbf{\Psi}}, \quad (8)$$

$$\sigma^2(\mathbf{a}) = k(\mathbf{a}, \mathbf{a}) - \mathbf{k}(\mathbf{a})\mathbf{K}^{-1}\mathbf{k}^T(\mathbf{a}), \quad (9)$$

where $\hat{\mathbf{\Psi}} = [\hat{\psi}(\mathbf{a}_1), ..., \hat{\psi}(\mathbf{a}_N)]^T$ is the vector of observed function values, the covariance matrix has entries $\mathbf{K}_{(i,j)} = k(\mathbf{a}_i, \mathbf{a}_j), \quad i, j \in 1, ..., N$, and $\mathbf{k}(\mathbf{a}) = [k(\mathbf{a}, \mathbf{a}_1), ..., k(\mathbf{a}, \mathbf{a}_N)]$ is the vector of the covariances between the query point $\mathbf{a}$ and the observed data points in $\mathcal{D}$.

Kernel selection for control-affine systems: For control-affine systems, the nonlinear map $\psi(\mathbf{a}) = \psi(\mathbf{z}, u)$ is affine in the control input, i.e., we can write $\psi(\mathbf{z}, u) = \alpha(\mathbf{z}) + \beta(\mathbf{z})u$. We can encode this structure in the selection of the kernel of the GP as:

$$k(\mathbf{a}_i, \mathbf{a}_j) = k_\alpha(\mathbf{z}_i, \mathbf{z}_j) + u_i k_\beta(\mathbf{z}_i, \mathbf{z}_j)u_j + \delta_{ij}\sigma_\eta^2, \quad (10)$$

where σ_η^2 is the observation noise and $k_\alpha(\cdot,\cdot)$ and $k_\beta(\cdot,\cdot)$ are often selected to be common kernel functions (e.g., SE functions).

Assumption 2: k_α and k_β are positive definite kernels.

Assumption 3: k_α and k_β are bounded kernels.

Lemma 2: Given *Assumption 2*, then the kernel (10) is also positive definite. Moreover, if *Assumption 3* holds, then the kernel (10) is also a bounded kernel.

Proof: It follows from [13] that the affine dot product compound kernel, i.e., $u_i k_\beta(\mathbf{z}_i, \mathbf{z}_j)u_j$, is positive definite and bounded provided that $k_\beta(\mathbf{z}_i, \mathbf{z}_j)$ is positive definite and bounded. Consequently, the kernel (10) is also positive definite and bounded as it is the addition of two positive definite and bounded kernels. ■

Using this kernel structure (10), the predicted mean $\mu(\mathbf{a})$ and variance $\sigma^2(\mathbf{a})$ at any query point $\mathbf{a} = (\mathbf{z}, u)$, conditioned on N noisy observations $\mathcal{D} = \{\mathbf{a}_i, \hat{\psi}(\mathbf{a}_i)\}_{i=1}^N$, are linear and quadratic in u, respectively, or more explicitly:

$$\mu(\mathbf{a}) = \gamma_1(\mathbf{z}) + \gamma_2(\mathbf{z})u, \quad (11)$$

$$\sigma^2(\mathbf{a}) = \gamma_3(\mathbf{z}) + \gamma_4(\mathbf{z})u + \gamma_5(\mathbf{z})u^2, \quad (12)$$

where:

$$\gamma_1(\mathbf{z}) = \mathbf{k}_\alpha(\mathbf{z})\mathbf{K}^{-1}\hat{\mathbf{\Psi}}, \quad \gamma_2(\mathbf{z}) = \mathbf{k}_\beta(\mathbf{z})\mathbf{K}^{-1}\hat{\mathbf{\Psi}},$$
$$\gamma_3(\mathbf{z}) = \left(k_\alpha(\mathbf{z}, \mathbf{z}) - \mathbf{k}_\alpha(\mathbf{z})\mathbf{K}^{-1}\mathbf{k}_\alpha^T(\mathbf{z})\right),$$
$$\gamma_4(\mathbf{z}) = -\left(\mathbf{k}_\beta(\mathbf{z})\mathbf{K}^{-1}\mathbf{k}_\alpha^T(\mathbf{z}) + \mathbf{k}_\alpha(\mathbf{z})\mathbf{K}^{-1}\mathbf{k}_\beta^T(\mathbf{z})\right),$$
$$\gamma_5(\mathbf{z}) = \left(k_\beta(\mathbf{z}, \mathbf{z}) - \mathbf{k}_\beta(\mathbf{z})\mathbf{K}^{-1}\mathbf{k}_\beta^T(\mathbf{z})\right).$$

The vectors $\mathbf{k}_\alpha(\mathbf{z}) = [k_\alpha(\mathbf{z}, \mathbf{z}_1), ..., k_\alpha(\mathbf{z}, \mathbf{z}_N)]$ and $\mathbf{k}_\beta(\mathbf{z}) = [k_\beta(\mathbf{z}, \mathbf{z}_1)u_1, ..., k_\beta(\mathbf{z}, \mathbf{z}_N)u_N]$.

IV. METHODOLOGY

In the proposed approach, we exploit the differential flatness structure and learn (2) using a GP as described in Section III-C. We combine feedback linearization with a stability filter by using the learned GP model to 1) optimize for an input u that most likely feedback linearizes the system (2)-(3) while 2) guaranteeing that the stability filter, described by the CLF (7) is decreasing with high probability. We combine feedback linearization with a stability filter in an optimization framework, which can also account for input constraints. Further, we show that for control-affine systems, by carefully selecting the kernel (10), the resulting optimization can be described by a second-order cone program that can be solved in polynomial time by standard interior point methods. Our proposed approach has three key components:

Probabilistic Feedback Linearization - see Section IV-A: Based on our learned GP model for (2), we optimize for an input u such that the predicted output of the GP is likely to match the designed input to the linear dynamics v_{des}, typically computed by a linear optimal controller based on the linear dynamics (3).

Probabilistic Stability Filter - see Section IV-B: Using the learned GP model for (2), we include a stability filter that guarantees that (7) is decreasing with high probability.

Linearization & Stability Filter Optimization - see Section IV-C: We combine the *probabilistic feedback linearization* with the *probabilistic stability filter*. At each time step, we propose to solve an optimization problem that finds the input u that is mostly likely to result in a unity mapping between v_{des} and the input seen by the linear dynamics (3) while ensuring robust tracking stability and input constraint satisfaction. For control-affine systems, we exploit the encoded structure and show that the resulting optimization is a second-order cone program.

A. Probabilistic Feedback Linearization

The objective of feedback linearization is to create a unity mapping between the desired input v_{des}, determined by some outer-loop linear controller, and the input $v = \psi(\mathbf{z}, u)$ seen by the linear dynamics. While $\psi(\mathbf{z}, u)$ is unknown, we have approximated this mapping from data using a GP. We propose finding an input u that is most likely to result in such a feedback linearization. More precisely, we compute an input u that minimizes the expected squared distance between $\psi(\mathbf{z}, u)$ and v_{des}:

$$\min_u \mathbb{E}(||\psi(\mathbf{z}, u) - v_{des}||^2).$$

Given that we have learnt the function $\psi(\mathbf{z}, u)$ using a GP, we can infer its value at any given query point $\mathbf{a}^* = (\mathbf{z}, u)$ conditioned on the data $\mathcal{D}$ as a Gaussian, i.e., $\psi(\mathbf{a}^*)|\mathcal{D} = \mathcal{N}(\mu(\mathbf{a}^*), \sigma^2(\mathbf{a}^*))$ where the mean and covariance are given by (8) and (9). Using this, we can rewrite the optimization problem as:

$$\min_u (\mu(\mathbf{z}, u) - v_{des})^2 + \sigma^2(\mathbf{z}, u). \tag{13}$$

Probabilistic feedback linearization for control-affine systems: For control-affine systems, we can exploit the kernel structure in (10). This allows us to rewrite the mean $\mu(\mathbf{z}, u)$ as a linear function of input u, using (11), and the covariance $\sigma^2(\mathbf{z}, u)$ as a quadratic function of input u, using (12). Plugging in (11) and (12) into (13), we rewrite the optimization problem, neglecting constant terms with respect to the optimization variable u, as a convex quadratic program in u:

$$\min_u (\gamma_2^2(\mathbf{z}) + \gamma_5(\mathbf{z}))u^2 + (2\gamma_1(\mathbf{z})\gamma_2(\mathbf{z}) - 2\gamma_2(\mathbf{z})v_{des} + \gamma_4)u. \tag{14}$$

Remark: The optimization (14) is convex because $\gamma_2^2(\mathbf{z}) + \gamma_5(\mathbf{z}) \geq 0$ since the function $\gamma_5(\mathbf{z}) \geq 0$ is the predicted covariance of $\beta(\mathbf{z})$ in (4) conditioned on the data $\mathcal{D}$.

Theorem 1: The functions $\gamma_i(\mathbf{z})$ are real-valued and Lipschitz continuous on the compact set $\mathbf{z} \in \mathcal{Z}$. The functions $\gamma_2(\mathbf{z})$ and $\gamma_5(\mathbf{z})$ are never both zero. The desired input $v_{des}(\mathbf{z})$ is also real-valued and Lipschitz continuous on the compact set $\mathbf{z} \in \mathcal{Z}$. Under these assumptions, the input $u(\mathbf{z})$ computed using (14) is also Lipschitz continuous on the compact set $\mathbf{z} \in \mathcal{Z}$.

Proof: The solution of (14) has a closed-form solution:

$$u = \frac{-\gamma_1(\mathbf{z})\gamma_2(\mathbf{z}) + \gamma_2(\mathbf{z})v_{des}(\mathbf{z}) - \frac{1}{2}\gamma_4(\mathbf{z})}{\gamma_2^2(\mathbf{z}) + \gamma_5(\mathbf{z})}.$$

The numerator is Lipschitz continuous on $\mathcal{Z}$ since it is a linear combination of the products of Lipschitz continuous functions that are bounded on $\mathcal{Z}$. Similarly, the denominator is also Lipschitz continuous on $\mathcal{Z}$. Since, $\gamma_2^2(\mathbf{z}) + \gamma_5(\mathbf{z}) > 0$, it follows that the resulting quotient is also Lipschitz continuous on $\mathcal{Z}$. ■

While the resulting control law (14) is Lipschitz continuous, it cannot guarantee robust stability and tracking convergence. We propose extending the probabilistic feedback linearization formulation (14) by also including a stability filter that guarantees tracking convergence with high probability.

B. Probabilistic Stability Filter

We probabilistically bound the error between the true nonlinear function value (2) and the learnt mean value (8).

Assumption 4: The function $\psi(\mathbf{a})$ has a bounded reproducing kernel Hilbert space (RKHS) norm $||\psi(\mathbf{a})||_k$ with respect to the kernel $k(\mathbf{a}_i, \mathbf{a}_j)$ of the GP, and the observation noise η is uniformly bounded by σ_η.

Theorem 2: [1] Given Assumption 4. Let $\delta \in (0, 1)$, then:

$$\Pr\{\forall \mathbf{a} \in \mathcal{A}, |\mu(\mathbf{a}) - \psi(\mathbf{a})| \leq \beta^{1/2}\sigma(\mathbf{a})\} \geq 1 - \delta,$$

where $\Pr\{\cdot\}$ is the probability, $\mathcal{A}$ is compact, $\mu(\mathbf{a})$ is the GP mean, $\sigma^2(\mathbf{a})$ is the GP covariance and

$$\beta = 2||\psi(\mathbf{a})||_k + 300\gamma \ln^3((N+1)/\delta),$$

where $\gamma \in \mathbb{R}$ is the maximum information gain.

We now exploit Theorem 2 to rewrite the decreasing CLF condition in (7) using the learnt GP mean (8) and variance (9) such that this condition holds with high probability.

We can rewrite the decreasing CLF condition in (7) using $\mathbf{e}^T\mathbf{P}(\mathbf{A}\mathbf{z}+\mathbf{B}\psi(\mathbf{z}, u)+\mathbf{B}v_{nom}-\mathbf{B}v_{nom}-\dot{\mathbf{z}}_{ref}) \leq -c_3\mathbf{e}^T\mathbf{P}\mathbf{e}$, where v_{nom} comes from (6). Using the algebraic Riccati equation, this simplifies to $-\mathbf{e}^T\mathbf{S}\mathbf{e} + 2\mathbf{e}^T\mathbf{P}\mathbf{B}(\psi(\mathbf{z}, u) - v_{nom}) \leq -c_3\mathbf{e}^T\mathbf{P}\mathbf{e}$.

We have learnt $\psi(\mathbf{z}, u)$ as a GP. Defining $w := \mathbf{e}^T\mathbf{P}\mathbf{B}$ and recalling that the query $\mathbf{a}$ comprises of the state and input, i.e., $\{\mathbf{z}, u\}$, we use Theorem 2 to obtain:

$$\Pr\{w(\psi(\mathbf{z}, u) - v_{nom}) \leq w(\mu(\mathbf{z}, u) - v_{nom}) + |w|\beta^{1/2}\sigma(\mathbf{z}, u)\} \geq 1 - \delta.$$

We use this probabilistic condition to rewrite (7):

$$-\mathbf{e}^T\mathbf{S}\mathbf{e}+2w(\mu(\mathbf{z}, u)-v_{nom})+2|w|\beta^{1/2}\sigma(\mathbf{z}, u) \leq -c_3\mathbf{e}^T\mathbf{P}\mathbf{e}, \tag{15}$$

where $c_3 = \frac{\lambda_{min}(\mathbf{S})}{\lambda_{max}(\mathbf{P})}$. While the constraint (15) is not necessarily convex, for systems that are control-affine, we propose exploiting this structure in the kernel selection of the GP (10).

Probabilistic stability filter for control-affine systems: For control-affine systems, we choose to exploit the kernel structure in (10). By Lemma 2, the kernel (10) is bounded and positive definite and, therefore, we can construct a corresponding RKHS. We make a similar assumption to *Assumption 4* for control-affine systems and kernel (10) such that we can similarly apply Theorem 2.

Assumption 5: The control-affine nonlinear function (4) has a bounded RKHS norm with respect to kernel (10) used in the GP, and the observation noise η is uniformly bounded by σ_η.

Under *Assumption 5*, we apply Theorem 2 and can rewrite the probabilistic decreasing CLF condition (15) using the mean (11), which is linear in control input u, and covariance (12), which is quadratic in control input u. Plugging (11) and (12) into (15), the filter condition becomes:

$$\begin{aligned} &2w(\gamma_1(\mathbf{z}) + \gamma_2(\mathbf{z})u - v_{nom}) \\ &\quad + 2|w|\beta^{1/2}\sqrt{\gamma_3(\mathbf{z}) + \gamma_4(\mathbf{z})u + \gamma_5(\mathbf{z})u^2} \\ &\quad \leq -c_3\mathbf{e}^T\mathbf{P}\mathbf{e} + \mathbf{e}^T\mathbf{S}\mathbf{e}. \end{aligned} \tag{16}$$

C. Linearization and Stability Filter Optimization

We combine feedback linearization with the safety filter by optimizing (13) subject to the probabilistically robust decreasing CLF condition (15). We can write this as an optimization problem including input constraints:

$$\begin{aligned} \min_{u,d} \quad & (\mu(\mathbf{z},u) - v_{des})^2 + \sigma^2(\mathbf{z},u) + \rho d^2 \\ \text{s.t.} \quad & 2w\left(\mu(\mathbf{z},u) - v_{nom}\right) + 2|w|\beta^{1/2}\sigma(\mathbf{z},u) \\ & \qquad \leq \mathbf{e}^T(-c_3\mathbf{P} + \mathbf{S})\mathbf{e} + d, \\ & u_{min} \leq u \leq u_{max}, \end{aligned} \tag{17}$$

where d is a slack variable added to ensure feasibility of the above optimization problem and ρ is a large weight. This optimization problem is not necessarily convex.

Optimization for control-affine systems: For control-affine systems (4), we use the simplications made in Sections IVA-B such that we can rewrite the optimization (17) as a second-order cone program (SOCP) as stated below in Theorem 3.

Theorem 3: Given *Assumptions 1, 2, 3* and *5*, the optimization problem (17) can be written as a convex optimization problem. Moreover, it is a SOCP.

Proof: We can rewrite both the CLF constraint (16), including the slack variable d, and the convex quadratic optimization problem (14) as second-order cone (SOC) constraints.

In (16), we can rewrite:

$$\sqrt{\gamma_3(\mathbf{z}) + \gamma_4(\mathbf{z})u + \gamma_5(\mathbf{z})u^2} = \left|\left|\begin{bmatrix} \sqrt{\gamma_5(\mathbf{z})}u + \frac{\gamma_4(\mathbf{z})}{2\sqrt{\gamma_5(\mathbf{z})}} \\ \sqrt{\gamma_3(\mathbf{z}) - \frac{\gamma_4^2(\mathbf{z})}{4\gamma_5(\mathbf{z})}} \end{bmatrix}\right|\right|_2$$

where $||\cdot||_2$ denotes the L_2-norm. This is possible because $\gamma_3(\mathbf{z}) - \frac{\gamma_4^2(\mathbf{z})}{4\gamma_5(\mathbf{z})} \geq 0$ since the covariance in (12) is positive. We can, therefore, rewrite the first SOC constraint in the standard form:

$$||\bar{\mathbf{A}}_1\bar{\mathbf{u}} + \bar{\mathbf{b}}_1||_2 \leq \bar{\mathbf{c}}_1\bar{\mathbf{u}} + \bar{\mathbf{d}}_1, \tag{18}$$

where the optimization variables $\bar{\mathbf{u}} = [u, d, f]^T$ include the input u, the slack variable d and a dummy variable f. The matrix $\bar{\mathbf{A}}_1 = \text{diag}(|w|\sqrt{\gamma_5(\mathbf{z})}, 0, 0)$ and the vector $\bar{\mathbf{b}}_1 = [|w|\frac{\gamma_4(\mathbf{z})}{2\sqrt{\gamma_5(\mathbf{z})}}, |w|\sqrt{\gamma_3(\mathbf{z}) - \frac{\gamma_4^2(\mathbf{z})}{4\gamma_5(\mathbf{z})}}, 0]^T$. By rewriting (16), the vectors $\bar{\mathbf{c}}_1 = [\frac{-w\gamma_2(\mathbf{z})}{\beta^{1/2}}, \frac{1}{2\beta^{1/2}}, 0]$, and $\bar{\mathbf{d}}_1 = \frac{w}{\beta^{1/2}}(v_{nom} - \gamma_1(\mathbf{z})) + \frac{1}{2\beta^{1/2}}\mathbf{e}^T(\mathbf{S} - c_3\mathbf{P})\mathbf{e}$.

We rewrite the optimization (14) including a dummy variable $f \geq (\gamma_2^2(\mathbf{z}) + \gamma_5(\mathbf{z}))u^2 + \rho d^2$ where ρ is a large weight and d is the slack variable. It follows that $0 \geq 4((\gamma_2^2(\mathbf{z}) + \gamma_5(\mathbf{z}))u^2 + \rho d^2) - 4f$ which can be rewritten as $(1+f)^2 \geq 4((\gamma_2^2(\mathbf{z}) + \gamma_5(\mathbf{z}))u^2 + \rho d^2) + (1-f)^2$. Since both sides of the inequality are positive, we can rewrite this condition as:

$$\left|\left|\begin{bmatrix} 2\sqrt{\gamma^2(\mathbf{z}) + \gamma_5(\mathbf{z})}u \\ 2\rho^{1/2}d \\ 1-f \end{bmatrix}\right|\right|_2 \leq 1-f,$$

which allows us to write the second SOC constraint in the standard form as:

$$||\bar{\mathbf{A}}_2\bar{\mathbf{u}} + \bar{\mathbf{b}}_2||_2 \leq \bar{\mathbf{c}}_2\bar{\mathbf{u}} + \bar{\mathbf{d}}_2, \tag{19}$$

where $\bar{\mathbf{A}}_2 = \text{diag}(2\sqrt{\gamma^2(\mathbf{z}) + \gamma_5(\mathbf{z})}, 2\rho^{1/2}, -1)$, $\bar{\mathbf{b}}_2 = [0,0,1]^T$, $\bar{\mathbf{c}}_2 = [0,0,1]^T$ and $\bar{\mathbf{d}}_2 = 1$. We rewrite the optimization problem in standard SOCP form:

$$\begin{aligned} \min_{\bar{\mathbf{u}}} \quad & \begin{bmatrix} 2\gamma_1(\mathbf{z})\gamma_2(\mathbf{z}) - 2\gamma_2(\mathbf{z})v_{des} + \gamma_4 & 0 & 1 \end{bmatrix}\bar{\mathbf{u}} \\ & \text{s.t. SOC constraints (18) \& (19),} \\ & \quad u_{min} \leq u \leq u_{max}, \end{aligned} \tag{20}$$

where we recall that $\bar{\mathbf{u}} = [u, d, f]^T$ includes the input u, the slack variable d and the dummy variable f. ■

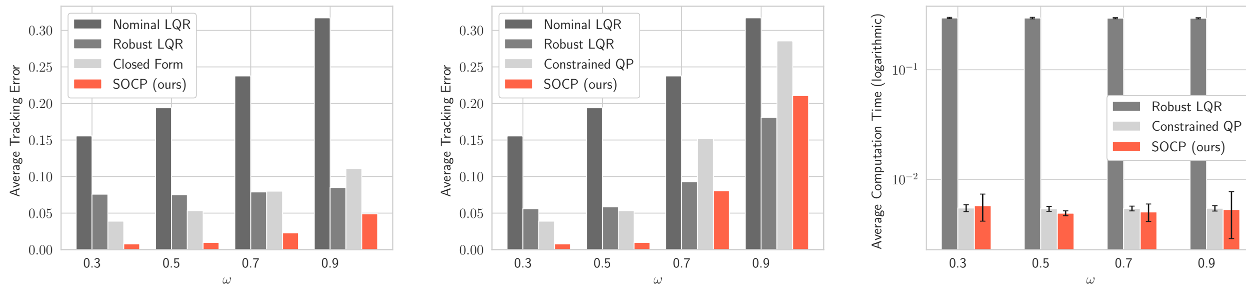
Remark: SOCPs can be solved in polynomial time by interior point methods [14].

At each time step, we solve the SOCP (20), which efficiently solves for an input u that balances feedback linearization objectives with robust stability requirements and input constraints.

V. SIMULATION RESULTS

We compare our proposed *SOCP* (20) method, with similar GP learning-based methods on a SISO 1-D quadcoptor moving in the horizontal direction. The dynamics follow [5], with $\ddot{x} = T\sin(\theta) - \gamma\dot{x}$ and $\dot{\theta} = \frac{1}{\tau}(u - \theta)$, where x is the horizontal position, θ is the pitch angle, and the input u is the commanded pitch angle.

To compare the algorithms in the unconstrained case, input constraints are neglected, and our proposed *SOCP* method is compared against the *closed-form* solution of (14), a *nominal LQR* that uses an inaccurate prior model, and a learned *robust LQR* from [5]. To compare the algorithms in the input-constrained case, input constraints are included, our proposed *SOCP* method is compared against a *constrained QP* that optimizes (14) subject to input constraints, *nominal LQR* that uses an inaccurate prior model and saturates the inputs at the



(a) Average tracking error w/o input constraints (b) Average tracking error w/ input constraints (c) Average computation time (s)

Fig. 2. Average tracking error (a) without input constraints and (b) with input constraints for *nominal LQR* (no learning), *robust LQR* [5], *constrained QP* optimizing (14), and our proposed *SOCP* approach (red). We compare the average computation times in (c) where the error bars correspond to one standard deviation.

constraints, and a learned *robust LQR* [5] from prior work with saturated inputs.

For all simulations, we use $\beta^{1/2} = 2$ and $\rho = 625$. The true dynamics have a time constant $\tau = 0.2$, thrust $T = 10$, and drag $\gamma = 0.3$. The nominal model has estimated parameters $\hat{\tau} = 0.15$, $\hat{T} = 7$, and $\hat{\gamma} = 0$. Both these models are differentially flat in the output $y = x$. A GP, using the kernel function (10), is used to learn the unknown nonlinear term $\psi(\mathbf{z}, u)$. All GP parameters are optimized to minimize the GP's log-likelihood over the training data. The training trajectory is generated using the *nominal LQR*, with gains $\mathbf{Q} = \text{diag}(20, 15, 5)$ and $\mathbf{R} = 0.1$, and feedback linearization, based on the nominal model, to track $y_{ref} = At\sin(\omega t)$ with $A = 1$, $\omega = 1$ for 5 seconds.

To compare closed-loop performance, each controller's tracking performance is assessed along 4 different trajectories with $\omega = 0.3,\ 0.5,\ 0.7,\ 0.9$ and $A = 0.2$. The average output tracking error and computation times are compared in Figure 2.

In the unconstrained case, our proposed *SOCP* method provides up to an 90% decrease in average tracking error as compared with *robust LQR*, with larger decreases in error occurring when the trajectory is further away from the training data. In the constrained case, using input constraints $-15 \leq u \leq 15$, the average tracking error of our proposed *SOCP* approach is nearly as good as the *robust LQR*, and up to 85% less for trajectories further away from the training trajectory. We note that the tracking errors of all the compared approaches increase significantly and are comparable in the constrained cases when $\omega = 0.7$ and $\omega = 0.9$. This is due to the input constraints being reached for a significant portion of the trajectory, dominating the tracking error. Additionally, our *SOCP* approach has an average computational time nearly two orders of magnitude smaller than the *robust LQR* case and is comparable to a standard QP, significantly reducing the required computational power.

VI. CONCLUSION

Our proposed approach (SOCP) has significantly better performance than the constrained QP approach at a similar computational cost. While a Robust LQR can outperfom us in rare cases (i.e., when the trajectory is infeasible for the given input constraints) it comes at a significantly higher computational cost. Our proposed approach efficiently handles robust tracking stability and input-based constraints in the presence of model uncertainty by exploiting the structure of control-affine differentially flat systems. As future work, we will investigate a data-driven CLF selection and extend our approach to include Control Barrier Functions [15].

REFERENCES

[1] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Trans. Inf. Theory*, 2012.

[2] C. J. Ostafew, A. P. Schoellig and T. D. Barfoot, "Robust constrained learning-based NMPC enabling reliable mobile path-tracking," *International Journal of Robotics Research*, 2016.

[3] C. D. McKinnon and A. P. Schoellig, "Learning probabilistic models for safe predictive control in unknown environments," in *Proc. of the European Control Conference (ECC)*, 2019.

[4] J. Umlauft, L. Pöhler and S. Hirche, "An Uncertainty-Based Control Lyapunov Approach for Control-Affine Systems Modeled by Gaussian Process," *IEEE Control Systems Letters*, 2018.

[5] M. Greeff and A. P. Schoellig, "Exploiting Differential Flatness for Robust Learning-Based Tracking Control Using Gaussian Processes," *IEEE Control Systems Letters*, 2021.

[6] M. Fliess, J. Lévine, P. Martin and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *International Journal of Control*, 1995.

[7] J. Umlauft, T. Beckers, M. Kimmel and S. Hirche, "Feedback linearization using Gaussian processes," in *Proc. IEEE Conference on Decision and Control (CDC)*, 2017.

[8] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[9] L. Zheng, R. Yang, J. Pan, H. Cheng and H. Hu, "Learning-Based Safety-Stability-Driven Control for Safety-Critical Systems under Model Uncertainties," in *Proc. International Conference on Wireless Communications and Signal Processing (WCSP)*, 2020.

[10] E. Sontag. "A 'universal' construction of Artstein's theorem on nonlinear stabilization," *Systems & Control Letters*, 1989.

[11] A. Taylor, V. Dorobantu, H. Le, Y. Yue, and A. Ames. "Episodic learning with control lyapunov functions for uncertain robotic systems," in *Proc. International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[12] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. Cambridge, MA: MIT Press, 2006.

[13] F. Castaneda, J. J. Choi, B. Zhang, C. J. Tomlin, and K. Sreenath, "Gaussian process-based min-norm stabilizing controller for control-affine systems with uncertain input effects," *arXiv preprint*, arXiv:2011.07183, 2020.

[14] Y. Nesterov and A. Nemirovski, *Interior-point polynomial algorithms in convex programming*, Siam studies in applied mathematics: 1994.

[15] D. D. Fan, J. Nguyen, R. Thakker, N. Alatur, A. A. Agha-mohammadi and E. A. Theodorou, "Bayesian Learning-Based Adaptive Control for Safety Critical Systems," in *Proc. International Conference on Robotics and Automation (ICRA)*, 2020.