Control Of Uncertain Nonlinear Systems Using Non-parametric Machine Learning

K. Saisamhith[1] and Ketan Detroja[2]

Abstract— **Accurately modeling nonlinear dynamical systems is challenging due to model inaccuracies and uncertainties, which can significantly degrade controller performance. Designing stabilizing controllers for nonlinear systems under such uncertainties remains a challenging problem. A wide range of stabilizing control algorithms have been proposed in the existing literature; however, their effectiveness relies on the accuracy of system dynamics. This manuscript addresses the issue of stabilizing controller design for nonlinear systems, particularly when the drift vector field is uncertain, by integrating non-parametric machine learning techniques to estimate the unknown component. In this work, a framework is proposed in which Gaussian process regression (GPR) is employed to estimate the unknown drift vector field and integrate it into the control synthesis. Moreover, an event-triggered condition is employed to collect data online as needed. The proposed approach enables the construction of control laws that practically stabilize the system, despite incomplete system knowledge. Rigorous theoretical guarantees for the proposed method are provided, and the effectiveness of the framework is demonstrated through numerical simulation studies.**

I. INTRODUCTION

Nonlinear systems are encountered in many engineering domains, like chemical processing industries, power systems, robotics, etc. Designing stabilizing controllers for these systems is essential to their operation. Historically, control design for nonlinear systems has been based on linearizing them around the operating point. However, these control algorithms don't work well if the system deviates far from the operating point. Consequently, controllers are designed directly for nonlinear systems, using Lyapunov-based methods to overcome the limitations of linearization-based approaches. Among the most popular methods are feedback linearization [1], control Lyapunov function (CLF) based methods [2], Passivity-based control [3], and back-stepping control [4]. Feedback linearization is a differential geometric method: the nonlinear system is transformed into an equivalent linear system via a diffeomorphism, and the controller is designed on the transformed space. CLF-based methods provide an alternative framework, wherein an explicit feedback law, such as those proposed in [2], [5], is designed to guarantee asymptotic stabilization of nonlinear systems. The back-stepping methods exploit special system structures(i.e, strict feedback form, pure feedback form) to achieve stabilization through recursive design procedures. Whereas, Passivity-based control offers another powerful framework by leveraging the energy properties of nonlinear systems. More recently, nonlinear model predictive control(N-MPC) [6] techniques have been developed to handle system constraints and performance objectives. N-MPC recursively solves an optimization problem over an N-step prediction horizon to find a feasible control input. Collectively, these methods enable controller synthesis directly from nonlinear dynamics while providing rigorous stability guarantees and applicability across diverse engineering domains. The nonlinear control design methods described above rely on a perfect model of the system. However, in practice these systems are subjected to model uncertainties. Historically, these uncertainties have been modeled as parametric structures, or approximated using neural networks to enable adaptive control techniques. In recent years, non-parametric machine learning models like Gaussian process regression (GPR) have gained traction for approximating the unmodeled inaccuracies. In GPR, the unmodeled components are taken as samples of a Gaussian process, and updated according to the collected data. The advantages of GPR over parametric models, such as neural networks in modelling uncertainties are well documented in [7]. Unlike neural networks, GPs provide analytical error bounds without assuming a predefined structure on unknown function, making them valuable in control tasks. GPR has already been utilized in many control design problems, like feedback linearization [8], [9] of certain class of systems (i.e, in explicit single input Brunovsky canonical form), Model-reference-adaptive control [10], and minimum norm controller [11], etc. In [8], [11], the uncertainty is learned from the large amount of data that is collected offline, in an episodic manner. In [11] probabilistic guarantees are provided for the performance of the control law, which in turn depend explicitly on the amount of data collected. Many extensions of Gaussian process regression are formulated, like sparse GPR [12], recursive GPR [13] to increase the scalability. Furthermore, the articles [14], [9] used a predefined condition for data acquisition strategy that judiciously collect data online only when necessary. This article follows a similar philosophy: it collects essential offline data for initialization and hyperparameter tuning of the GPR model, and then utilizes an event-triggered condition to collect additional data when necessary to update the model.

The main contributions of this article are summarized as follows:

- This article addresses the *control design problem* for general control-affine nonlinear systems with an *unmodeled component in the drift vector field*, without assuming any parametric structure or prior knowledge of the unknown dynamics.
- The unmodeled dynamics are learned from data using *Gaussian Process regression (GPR)*, and the learned model is integrated into control design.
- An event trigger condition is proposed using the deterministic error bound of Gaussian process to collect additional data online, when it is necessary. Hence, the guarantees provided in this paper are not probabilistic.

*This work was not supported by any organization

[1]Ketan Detroja is with Department of Electrical Engineering IIT Hyderabad, Kandi, Telangana, India ketan@ee.iith.ac.in

[2]K. Saisamhith is with the Department of Electrical Engineering, IIT Hyderabad, Kandi, Telangana, India ee23resch01002@iith.ac.in

A. Notations:

$L_f E = \nabla E(x)^T f(x)$, $\mathbb{R}^n$ is real domain of dimension n, norms considered in this article are 2-norms, i.e $||x|| \equiv ||x||_2$, $\mathcal{C}^1$ is the class of Continuously differentiable functions.

II. BACKGROUND

A. Gaussian process

Definition 1. (Gaussian Process)[15] : Let $\mathcal{I}$ be an index set. A *Gaussian process (GP)* is a stochastic process $\{X_t\}_{t\in\mathcal{I}}$ such that for any finite collection $\{t_1, t_2, \ldots, t_N\} \subset \mathcal{I}$, the random vector $\boldsymbol{X} = (X_{t_1}, X_{t_2}, \ldots, X_{t_N})^\top$ follows a multivariate Gaussian distribution.

A GP is fully specified by:

1) A *mean function* $m : \mathcal{I} \to \mathbb{R}$, where
$$m(t) = \mathbb{E}[X_t], \tag{1}$$
2) A *covariance (kernel) function* $k : \mathcal{I} \times \mathcal{I} \to \mathbb{R}$, where
$$k(t_i, t_j) = \text{Cov}(X_{t_i}, X_{t_j}) \tag{2}$$
$$= \mathbb{E}\left[(X_{t_i} - m(X_{t_i}))(X_{t_j} - m(X_{t_j}))\right]. \tag{3}$$

We denote a Gaussian process as:
$$X_t \sim \mathcal{GP}\big(m(t), k(t, t')\big). \tag{4}$$

$k(t, t')$ is also called as kernel function, some of the well-known kernel functions are squared Exponential Kernel (SEK): $k(x, x') = \rho^2 \exp\left(-\frac{\|x-x'\|^2}{2\ell^2}\right)$, Linear Kernel: $k(x, x') = \rho^2(x^\top x') + c$ etc, where ρ, l, c are hyper parameters. The kernel functions characterize the extent to which X_t and $X_{t'}$ are correlated.

B. GPR

GPR [15] belongs to a class of non-parametric machine learning models, which work under Bayesian framework. Due to their flexibility in representation, they outperform other regression models on small and medium data sets. GPR has theoretical guarantees to quantify uncertainty around their prediction. Of late, GPR became very popular among the control and machine learning communities.

Consider a regression model
$$\mathbf{y} = f(\mathbf{X}) + \boldsymbol{\epsilon} \tag{5}$$
where $f : \mathbb{R} \to \mathbb{R}$ is the function to be estimated with observed data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_{n_e}^2 \mathbf{I})$.
In GPR, the latent function f is modeled as a GP $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, and updated in Bayesian sense with the collected training data .

From the definition of GP, the joint distribution of observed output values $\mathbf{y} = [y_1, y_2, ..., y_n]^T$ and the predicted value y_* at test point $\mathbf{x}_*$ obey a multivariate Gaussian distribution:
$$\begin{bmatrix}\mathbf{y}\\ y_*\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}\\ \mu_*\end{bmatrix}, \begin{bmatrix}\mathbf{K} + \sigma_{n_e}^2\mathbf{I} & \mathbf{k}_*\\ \mathbf{k}_*^\top & k(\mathbf{x}_*, \mathbf{x}_*)\end{bmatrix}\right), \tag{6}$$

where $X = [x_1, x_2, .., x_n]^T$ is the input training data, $\boldsymbol{\mu} = m(\mathbf{X})$ and $\mu_* = m(\mathbf{x}_*)$ are the mean vectors, $\mathbf{K}$ is the $n \times n$ covariance matrix of training inputs with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{k}_*$ is the covariance vector between test point $\mathbf{x}_*$ and training inputs with components $[\mathbf{k}_*]_i = k(\mathbf{x}_*, \mathbf{x}_i)$, and $k(\mathbf{x}_*, \mathbf{x}_*)$ represents the process variance at the test point. The noise variance $\sigma_{n_e}^2$ accounts for observation noise in the training data.

The predictive distribution for y_* at test point $\mathbf{x}_*$ given observed data $(\mathbf{X}, \mathbf{y})$ is obtained through the conditional Gaussian distribution:
$$p(y_* \mid \mathbf{x}_*, \mathbf{X}, \mathbf{y}) \sim \mathcal{N}(\mu_*, \Sigma_*), \tag{7}$$
where the posterior mean and variance are given by:
$$\mu_* = m(\mathbf{x}_*) + \mathbf{k}_*^\top(\mathbf{K} + \sigma_{n_e}^2\mathbf{I})^{-1}(\mathbf{y} - m(\mathbf{X})) \tag{8}$$
$$\Sigma_* = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top(\mathbf{K} + \sigma_{n_e}^2\mathbf{I})^{-1}\mathbf{k}_* \tag{9}$$

In Eq (8) $m(\bullet)$ is prior mean. Prior knowledge of the unknown function such as its sign over a compact set or its values at specific points can be incorporated into Gaussian Process Regression (GPR) through an appropriate choice of the prior mean function $m(\mathbf{X})$, as discussed in [8] . $m(\mathbf{X})$ is taken as 0 when there is no prior knowledge of the unknown function. The hyper-parameters of the Gaussian Process model are tuned by maximizing the marginal likelihood of the observed training data. The hyperparameters denoted by $\boldsymbol{\psi}$ comprises length-scales and signal variances of the squared exponential kernels used to model the unknown functions. Assuming a zero prior mean, the unknown function is modeled as $f(x) \sim \mathcal{GP}(0, k(x, x'))$, where the kernel $k(\cdot, \cdot)$ is parameterized by $\boldsymbol{\psi}$. Given a training data $\mathcal{D} = \{x_i, y_i\}_{i=1}^{i=N}$, the marginal likelihood is given by $p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\psi}) = \mathcal{N}\big(\mathbf{y} \mid \mathbf{0},\, K(\mathbf{X}, \mathbf{X}) + \sigma_{n_e}^2 I\big)$, where $K_{ij} = k(x_i, x_j)$. Equivalently, the log marginal likelihood admits the closed-form expression
$$\log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\psi}) = -\tfrac{1}{2}\mathbf{y}^\top(K + \sigma_{n_e}^2 I)^{-1}\mathbf{y} - \tfrac{1}{2}\log|K + \sigma_{n_e}^2 I| - \tfrac{N}{2}\log(2\pi),$$
and the optimal hyperparameters $\boldsymbol{\psi}^*$ are obtained by maximizing this expression using standard gradient-based numerical optimization methods.
The posterior variance of the estimate decreases with increase in training data points, which was proven in [8]. Similar concept applies to a vector valued function, where each component is dealt similarly as above. Under some mild assumptions, GPR has analytical error bounds between the prediction and the actual function. The error bounds are of two kinds, probabilistic as discussed in [16], and deterministic as presented in [17]. This article utilizes deterministic error bounds to design the event triggering condition. The deterministic error bound is given as.

Lemma 1 (GP Prediction Error Bound)**.** *Under Assumption 1 in [17], Let Ω be a compact set, $\mathcal{D} = \{x_i, y_i\}_{i=1}^{i=N}$ be the training data set, and $\max_{1\le i\le N} ||f(x)_i||_{h_k} < B < \infty$. The Gaussian process regression trained on dataset $\mathcal{D}$, satisfies the following error bound:*
$$\|\boldsymbol{\mu}(\mathbf{x}) - \mathbf{f}(\mathbf{x})\| \le \eta(\mathbf{x}) = \sqrt{\text{tr}(\mathbf{B}\boldsymbol{\Sigma}(\mathbf{x}))}$$
$\forall x \in \Omega$, *where:*

- *$\mu(x)$ is the posterior mean, $f(x) : \mathbb{R}^n \to \mathbb{R}^n$ is the unknown function.*
- *$\mathbf{\Sigma}(\mathbf{x}) = diag(\sigma_1(x)^2, ..\sigma_n(x)^2)$ is the posterior variance matrix. Where, $\sigma_i(x)^2$ is the variance of the i^{th} component.*
- *Where $||f||_{h_k}$ is reproducing kernel Hilbert norm associated with respect to reproducing kernel Hilbert space induced by the kernel function used in the regression.*
- $\mathbf{B} = \text{diag}(\beta_1, \ldots, \beta_n)$ *with* $\beta_i = B_i^2 - \mathbf{y}_{\mathcal{D}}^\top(\mathbf{K}_{\mathcal{D}} + \sigma_{n_e}^2 \mathbf{I}_M)^{-1}\mathbf{y}_{\mathcal{D}} + M$, *where* $||f(x)_i||_{h_k} \leq B_i$, $\mathbf{y}_{\mathcal{D}} = [y_1, y_2, ...y_n]^T$, $[\mathbf{K}_{\mathcal{D}}]_{ij} = k(x_i, x_j)$

For the scalar function, the error bound becomes $|f(x) - \mu(x)| < \sqrt{\beta}\sigma(x)$. As observed in [17], the term $\eta(x)$ decreases as the number of training data points increases. Thus, as the number of training samples increases, especially within the region of interest the error bound $\eta(x)$ contracts.

C. Definitions:

Consider an affine nonlinear system of the form:

$$\dot{x} = f(x) + g(x)u,$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $f(x)$ and $g(x)$ are smooth functions with $f(0) = 0$.

Definition 2. A **Control Lyapunov Function (CLF)** is a proper, positive-definite function $E \in C^1(\mathbb{R}^n, \mathbb{R}_+)$ such that there exist class $\mathcal{K}_\infty$ functions α_1, α_2 satisfying

$$\alpha_1(\|x\|) \leq E(x) \leq \alpha_2(\|x\|), \quad \forall x \in \mathbb{R}^n,$$

and

$$\inf_{u \in \mathbb{R}^m} \nabla E(x)\,[f(x) + g(x)u] < 0, \quad \forall x \in \mathbb{R}^n \setminus \{0\}.$$

The feedback control law can be of the form given in [2], or in [5], these control laws are smooth every where except at the origin. Moreover $u(x)$ is continuous at origin if the CLF has small control property defined in [2] or satisfies strong control Lyapunov function condition, defined in [5].

Definition 3 (Practical Stability)**.** Consider the system $\dot{x} = f(x)$. The equilibrium $x = 0$ is said to be *practically stable* if there exists a constant $r > 0$ such that, for every initial condition $x(0)$ in a given domain, the system trajectories satisfy

$$\limsup_{t \to \infty} \|x(t)\| \leq r,$$

that is, the state is ultimately bounded inside the ball, given as

$$\mathcal{B}_r := \{x \in \mathbb{R}^n : \|x\| \leq r\}.$$

III. Problem Formulation:

A. System description:

Consider a system with uncertainty in drift vector field.

$$\dot{x} = l(x) + h(x) + g(x)u \tag{10}$$

where $l(x) \in \mathcal{C}^1, g(x) \in \mathcal{C}^1$ and $f(x) = l(x) + h(x)$. $l(x)$ constitute the nominal components of the system., and $h(x) \in C^1$ is the unknown component of the drift vector field, and $\mathcal{X}$ be the compact operating domain of the operating system.

In this article a paradigm is designed that ensures ultimately boundedness of the system in a compact neighborhood around the origin, i.e $\{x : E(x) \leq \Delta\}$ despite the presence of uncertainties.

Assumption 1. Let $\max_{1 \leq i \leq N} ||h(x)_i||_{h_k} < B$ and B is assumed to be known, which is a standard assumption taken in previous GPR based learning frameworks, e.g [9]. Moreover, B characterizes the complexity of the functions belonging to the reproducing kernel Hilbert space (RKHS) induced by the chosen kernel, Consequently, a larger value of B corresponds to an RKHS capable of representing more complex function classes.

Assumption 2. In this work, explicit knowledge of the unknown components is not necessary for CLF design. Rather, we assume that the nominal model possesses sufficient structural information to enable CLF construction, which is a standard assumption in the study of uncertain systems [18], [19]. Notably, [19] adopts the same assumption.

B. Data collection:

In this article The Lie derivative of $E(x)$ w.r.t $h(x)$ that is $L_h E$ is estimated using GPR. A certain amount of data required for hyperparameter tuning is collected offline, and the rest is collected online based on an event trigger condition. The training data is $D = \{Y_i, X_i\} = \{L_{\dot{x}_i}E - L_{l(x_i)}E - [L_{g(x_i)}E]^T u(x_i), x_i\}$. The collected data points are assumed to be noise free ($\sigma_{n_e} = 0$). Moreover, according to assumption-II, the Control Lyapunov Function (CLF) is constructed from the nominal model. consequently, there exist regions of the state space where the Lie derivative of the CLF along the true system dynamics is strictly negative. In particular, on the set $\Gamma = \{x \in \mathcal{X} : L_g E(x) = 0\} \cap \mathcal{X}$ the stabilization relies solely on the drift dynamics. Following the arguments of section III-B in [8], this prior knowledge about the sign of $L_f E$ in $\{\Gamma \cap \{x : E(x) \leq \Delta\}\}$ can be integrated into GPR to estimate the unknown drift component. Alternatively, This can also be achieved by collecting a sufficient amount of offline training data in the region $\{\Gamma \cap \{x : E(x) \leq \Delta\}\}$. In particularly, by incorporating an SMT verifier such as dReal [20] in the data collection loop using the falsifying logical condition as $\Big((x \in \Gamma \cap E(x) \leq \Delta) \wedge \tilde{L}_{\hat{f}} E(x) + \eta(x) \geq 0\Big)$, and augmenting the training dataset with counterexamples returned by the verifier, followed by appropriately initializing the GPR. Such an initialization ensures consistency of the learned drift dynamics in regions where control authority vanishes.

IV. Main results

Consider the control law [1]

$$u(x) = \begin{cases} \frac{-a - \sqrt{a^2 + ||b||^4}}{||b||^2} b & \text{for } E(x) \geq \Delta,\ ||b|| \neq 0 \\ 0 & otherwise \end{cases} \tag{11}$$

where $\hat{L}_f E = L_l E + \hat{L}_h E$, $\hat{L}_h E$ is a GPR estimate of $L_h E$, $\gamma > 0$ is tuning constant $a = \hat{L}_f E$, $b = L_g E$, Δ is some non-zero positive number. The above control is smooth every-where except at the origin, as discussed in [2]. Where as $\hat{L}_h E = \mu(x)$ and the variance of the estimation is $\sigma(x)^2$. In the remainder of this article, define $\Psi(x) = \sqrt{a^2 + ||b||^4} > 0$ in $E(x) \geq \Delta$.

[1]The proposed control law is discontinuous on the surface of $E(x) = \Delta$. Since the vector fields are measurable, locally bounded on the surface, and admits a nonempty, convex Filippov set on the surface. consequently,the closed-loop dynamics are well posed in the Filippov sense.

Theorem 1. *Let $X \subset \mathbb{R}^n$ be a compact operating domain and define*

$$\mathcal{E}_c := \{x \in \mathbb{R}^n : E(x) \leq c\} \subset X$$

for some constants $c > \Delta > 0$. Consider the control law $u(x)$ in (11)*. If the following condition is satisfied*

$$\sigma(x) < \frac{\Psi(x)}{\hat{\beta}}, \quad \forall x \in \{x \in \mathcal{E}_c : E(x) \geq \Delta,\ b(x) \neq 0\}$$

where $\hat{\beta} = \sqrt{\beta}$, then

$$\dot{E}(x) < 0$$

for all such x.

Proof. For $x \in \mathcal{E}^1$,

$$\begin{aligned}\dot{E}(x) &= L_f E + (L_g E)^T u \\ &= L_f E - L_{\hat{f}} E - \Psi(x) \\ &\leq |L_{f(x)} E - L_{\hat{f}(x)} E| - \Psi(x)\end{aligned}$$

From Lemma 1, we have

$$\begin{aligned}&\leq \hat{\beta}\sigma(x) - \Psi(x) \\ &< 0 \quad \text{if } \sigma(x) < \frac{\Psi(x)}{\hat{\beta}}\end{aligned}$$

■

From the condition derived in the previous theorem, and using not more than N data points to train. The following triggering rule is designed to collect the data online.

$$t_{k+1} = \inf\left\{t > t_k \,\middle|\, \hat{\beta}\sigma(x) \geq \Psi(x) \quad \& \quad E(x) \geq \Delta, b \neq 0\right\} \tag{12}$$

Theorem 2. *Consider the closed-loop system under the control law defined in* (11) *and the triggering condition given in* (12)*. Suppose the condition of Theorem 1 is satisfied for all $x \in \mathcal{E}_c$ with $E(x) \geq \Delta$.*

Then the origin is practically stable. In particular, the system trajectories are ultimately bounded in the set

$$\mathcal{R} := \left\{x \in \mathbb{R}^n : \|x\| \leq \alpha_1^{-1}(\Delta)\right\},$$

where α_1 is the class $\mathcal{K}_\infty$ function associated with the control Lyapunov function $E(x)$.

Proof. From theorem 1 the triggering rule ensures $\dot{E} < 0$ in the region $\{x : E(x) \geq \Delta\}$. The ultimate boundedness can be established through contradiction, as shown. Assume, for contradiction that x(t) doesn't reach $\mathcal{R}$ ultimately, it stays in the region $\mathcal{W} = \{x : c_1 \leq E(x) \leq c_2\}$, where $\mathcal{R} \subset \{x : E(x) \leq c_1\}$. Since, $\dot{E}$ is continuous and $\mathcal{E}$ is compact, minimum of $\dot{W}$ exists in that region and denote it as M. now

$$E(t) = E(t_0) + \int_{t_o}^{t} \dot{E}dt \tag{13}$$

$$\leq E(t_o) - M(t - t_0) \tag{14}$$

hence $E(t)$ will be negative for large values of t, this leads to a contradiction. Hence,the trajectory is ultimately bounded in $\mathcal{R}' = \{x : E(x) \leq \Delta\}$. From the Definition 1

$$\alpha_1(||x||) \leq E(x) \leq \Delta \tag{15}$$

$$\implies ||x|| \leq \alpha_1^{-1}(\Delta) \tag{16}$$

Therefore, $\mathcal{R}' \subset \mathcal{R}$. Hence, the trajectories are ultimately bounded in $\mathcal{R}$ ■

The implementation of the proposed framework is summarized in the following algorithm 1:

Algorithm 1 GP-Based Controler

Require: Initial state $x(0)$, CLF $E(x)$, and required design parameters.
Ensure: ultimate boundedness

```
Initialize GP model, hyper-parameters using offline training data
Set t ← 0
while system is running do
    Measure current state x(t)
    Compute CLF E(x)
    Compute a(x) = L_f̂ E(x) and b(x) = L_g E(x)
    if E(x) ≥ Δ and ‖b(x)‖ ≠ 0 then
        Compute control input
        u(x) ← −(a(x) + Ψ(x))/‖b(x)‖² b(x)
    else
        u(x) ← 0
    end if
    Apply control input u(x) to the system
    if β̂σ(x) ≥ Ψ(x)  &  E(x) ≥ Δ then
        Collect data point (x, ẋ)
        Update GP model using at most N recent data points
    end if
    t ← t + δ
end while
```

Theorem 3. *The triggering condition proposed in 12 guarantees a strictly positive inter-event time, that is, the trigger instances $\{t_k\}_{k=0}^{\infty}$ do not accumulate in finite time.*

Proof. Let us prove this by contradiction. suppose that accumulation happens in finite time,i.e, $\lim_{k\to\infty} t_k = T$. Assume the trigger happens at the k-th time, i.e., data is collected at t_k then $\sigma(x(t_k) = 0$. Let t_{k+1} be the next event time. Then from mean value theorem.

$$\sigma(x(t_{k+1})) - \sigma(x(t_k)) = [\frac{\partial\sigma(p)}{\partial x}]^T(x(t_{k+1}) - x(t_k)),$$

where p **is some point between** $x(t_k)$ & $x(t_{k+1})$

$$\begin{aligned}&\leq ||\frac{\partial\sigma(p)}{\partial x}||.||(x(t_{k+1}) - x(t_k)|| \\ &\leq M\int_{t_k}^{t_{k+1}} ||f(x) + g(x)u(x)||dt \\ &\leq M.M^{'}[t_{k+1} - t_k]\end{aligned}$$

where as, $max_{x\in\mathcal{E}}||\hat{f}(x)+g(x)u(x)|| = M^{'}$, and $||\frac{\partial\sigma(p)}{\partial x}|| = M$.

$$\implies \frac{\Psi(x(t_{k+1}))}{\hat{\beta}} \leq \sigma(x(t_{k+1})) \leq M.M^{'}[t_{k+1} - t_k]$$

$$\implies \tau = \frac{\min\limits_{t\in[0,T]} \Psi(x(t))}{\hat{\beta}.M.M'} \leq [t_{k+1} - t_k]$$

$$\implies [t_{k+1} - t_k] \geq \tau > 0$$

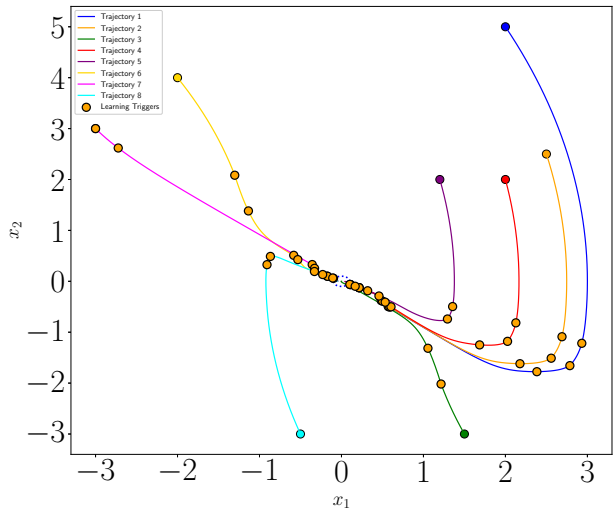

Fig. 1: Simulation for example-1

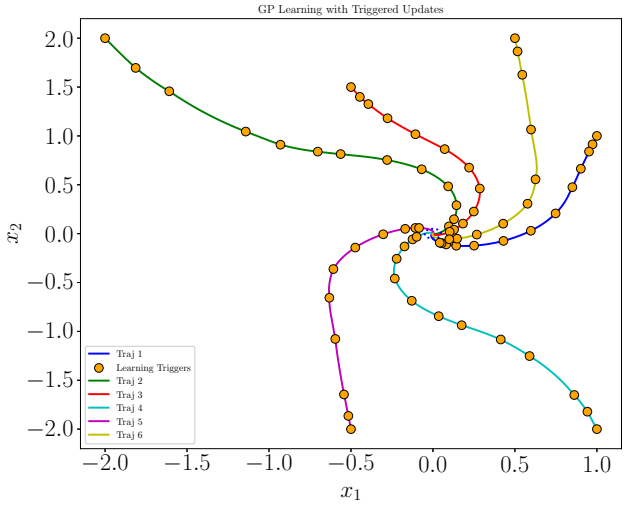

Fig. 2: Simulation for example-2

Then, for any finite time $T > 0$, the maximum number of triggering events that can occur in the interval $[0, T]$ is finite and bounded by $k \leq \lceil \frac{T}{\tau} \rceil$. Hence, all triggering instants t_k with $k \leq \lceil \frac{T}{\tau} \rceil$ satisfy $t_k \leq T$. Eventually, if $k \geq \lceil \frac{T}{\tau} \rceil + 1$, then $t_k > T$, which contradicts the assumption that there exists a finite $T > 0$ such that $\lim_{k\to\infty} t_k = T$. ■

V. Simulation Results

In this section, numerical simulations are presented to validate the proposed Gaussian process based control framework. The simulations are demonstrated on two nonlinear systems.

A. Example-1

Consider the Non-linear system.

$$\dot{x}_1 = x_2 + x_2^3, \quad \dot{x}_2 = -x_1 - f(x_2) + u, \tag{17}$$

where $x = [x_1, x_2]^\top \in \mathbb{R}^2$ is the state vector with $\mathcal{X} = [-10, 10] \times [-10, 10]$ and $u \in \mathbb{R}$ is the control input. The term $f(x_2)$ is given as

$$f(x_2) = (0.8 + 0.2\sin(x_2))\tanh(10x_2) + x_2, \tag{18}$$

which is assumed to be completely unknown to the controller and treated as an unmodeled component of the drift vector field. The control Lyapunov function is chosen as Control Lyapunov Function (CLF) is selected as $E(x) = x_1^2 + 1.5x_1x_2 + x_2^2$, and $\Delta = 0.1$.

For the GPR, a square exponential kernel is selected and B is assumed to be 10, and the GPR is trained with at most the latest $N = 5$ data points. For hyperparameter initialization is done by choosing some 30 random data points in the operating domain $\mathcal{X}$, and kept constant during online training. The simulation results are shown in the Fig 1 for 8 randomly chosen initial conditions.

B. Example-2

Consider the following system.

$$\dot{x} = \underbrace{\begin{bmatrix} x_2 \\ -x_1 \end{bmatrix}}_{\text{known drift}} + \underbrace{\begin{bmatrix} x_1(r^2-1) \\ \zeta(x_2)(r^2-1) \end{bmatrix}}_{\text{unknown drift}} + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} u, \tag{19}$$

where $x = [x_1, x_2]^\top \in \mathbb{R}^2$ is the sate vector with $\mathcal{X} = [-3, 3] \times [-3, 3]$, and $u \in \mathbb{R}$. where $r^2 = x_1^2 + x_2^2$, and $\zeta(x_2) = \left(0.8 + 0.2e^{-100|x_2|}\right)\tanh(x_2)$. The control Lyapunov function is takes as $E(x) = \frac{1}{2}x^\top x = \frac{1}{2}(x_1^2 + x_2^2)$, Δ is taken as 0.05. For the GPR, a square exponential kernel is selected and B is assumed to be 10, and the GPR is trained with atmost the latest $N = 5$ data points. For hyperparameter initialization is done by choosing some 10 random data points in the operating domain $\mathcal{X}$ and kept constant during online training. The simulation results are shown in the Fig 2 for some 6 randomly chosen initial conditions

The simulations are conducted in python 3.8 on two nonlinear system examples which do not belong to the class of systems considered in [9] . It is observed that the trajectories originating from the initial conditions successfully converge to a prescribed neighbourhood around the origin. The trigger instants at which the event-based data is sampled are also illustrated in Figs. 1,2, highlighting the evolution of the event-triggering mechanism along each trajectory.

VI. Conclusion

This article presented a GPR-based control framework for control-affine nonlinear systems with unknown drift dynamics. The unknown component is estimated using GPR, and the estimate is utilized in control design ensuring practical stability of the closed-loop system. An event-triggered data acquisition mechanism was employed to judiciously update the GP model when required. Rigorous theoretical analysis was performed for the proposed method, and numerical simulations are conducted on two nonlinear systems with uncertainties to demonstrate the effectiveness of the proposed approach. Future work will focus

on extending the framework to incorporate actuator constraints, and noise in data.

REFERENCES

[1] A. Isidori, *Nonlinear control systems: an introduction*. Springer, 1985.

[2] E. D. Sontag, "A 'universal'construction of artstein's theorem on nonlinear stabilization," *Systems & control letters*, vol. 13, no. 2, pp. 117–123, 1989.

[3] A. Van der Schaft, *L2-gain and passivity techniques in nonlinear control*. Springer, 2000.

[4] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos, *Nonlinear and adaptive control design*. John Wiley & Sons, Inc., 1995.

[5] J. Tsinias, "Sufficient lyapunov-like conditions for stabilization," *Mathematics of control, Signals and Systems*, vol. 2, no. 4, pp. 343–357, 1989.

[6] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[7] M. Liu, G. Chowdhary, B. C. Da Silva, S.-Y. Liu, and J. P. How, "Gaussian processes for learning and control: A tutorial with examples," *IEEE Control Systems Magazine*, vol. 38, no. 5, pp. 53–86, 2018.

[8] J. Umlauft, T. Beckers, M. Kimmel, and S. Hirche, "Feedback linearization using gaussian processes," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 5249–5255.

[9] J. Umlauft and S. Hirche, "Feedback linearization based on gaussian processes with event-triggered online learning," *IEEE Transactions on Automatic Control*, vol. 65, no. 10, pp. 4154–4169, 2019.

[10] G. Cho, G. Chowdhary, A. Kingravi, J. How, and A. Vela, "A bayesian nonparametric approach to adaptive control using gaussian processes," in *CDC*, 2013.

[11] T. Westenbroek, F. Castañeda, A. Agrawal, S. S. Sastry, and K. Sreenath, "Learning min-norm stabilizing control laws for systems with unknown dynamics," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 737–744.

[12] F. Leibfried, V. Dutordoir, S. John, and N. Durrande, "A tutorial on sparse gaussian processes and variational inference," *arXiv preprint arXiv:2012.13962*, 2020.

[13] M. F. Huber, "Recursive gaussian process: On-line regression and learning," *Pattern Recognition Letters*, vol. 45, pp. 85–91, 2014.

[14] T. Norton, G. Stagg, D. Ward, and C. K. Peterson, "Decentralized sparse gaussian process regression with event-triggered adaptive inducing points," *Journal of Intelligent & Robotic Systems*, vol. 108, no. 4, p. 72, 2023.

[15] J. Wang, "An intuitive tutorial to gaussian process regression," *Computing in Science & Engineering*, vol. 25, no. 4, pp. 4–11, 2023.

[16] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," *IEEE transactions on information theory*, vol. 58, no. 5, pp. 3250–3265, 2012.

[17] S. Zhang, D.-H. Zhai, X. Dai, T.-y. Huang, Y. Xia, and S. Hirche, "Learning-based parameterized barrier function for safety-critical control of unknown systems," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*. IEEE, 2024, pp. 8805–8810.

[18] M. F. AL-Sunni, H. Almubarak, and J. M. Dolan, "Safety embedded adaptive control using barrier states," *arXiv preprint arXiv:2504.15423*, 2025.

[19] T. Westenbroek, A. Agrawal, F. Castañeda, S. S. Sastry, and K. Sreenath, "Combining model-based design and model-free policy optimization to learn safe, stabilizing controllers," *IFAC-PapersOnLine*, vol. 54, no. 5, pp. 19–24, 2021.

[20] S. Gao, S. Kong, and E. M. Clarke, "dreal: An smt solver for nonlinear theories over the reals," in *International conference on automated deduction*. Springer, 2013, pp. 208–214.