Learning Disturbances Online for Risk-Aware Control: Risk-Aware Flight with Less Than One Minute of Data

Prithvi Akella[1] PAKELLA@CALTECH.EDU
Skylar X. Wei[1] SWEI@CALTECH.EDU
Joel W. Burdick[1] JWB@ROBOTICS.CALTECH.EDU
Aaron D. Ames[1] AMES@CALTECH.EDU
[1] *1200 E California Blvd MC 104-44, Pasadena, CA 91101*

Editors: N. Matni, M. Morari, G. J. Pappas

Abstract

Recent advances in safety-critical risk-aware control are predicated on *apriori* knowledge of the disturbances a system might face. This paper proposes a method to efficiently learn these disturbances online, in a risk-aware context. First, we introduce the concept of a *Surface-at-Risk*, a risk measure for stochastic processes that extends Value-at-Risk — a commonly utilized risk measure in the risk-aware controls community. Second, we model the norm of the state discrepancy between the model and the true system evolution as a scalar-valued stochastic process and determine an upper bound to its *Surface-at-Risk* via Gaussian Process Regression. Third, we provide theoretical results on the accuracy of our fitted surface subject to mild assumptions that are verifiable with respect to the data sets collected during system operation. Finally, we experimentally verify our procedure by augmenting a drone's controller and highlight performance increases achieved via our risk-aware approach after collecting less than a minute of operating data.

Keywords: Value-at-Risk, Risk-Aware Control, Gaussian Process, Scenario Optimization

1. Introduction

The models we use for control synthesis are useful, though oftentimes inaccurate. To wit, reduced order models are heavily utilized for controller synthesis for complex robotic systems, *e.g.* quadrupeds, bipeds, drones, *etc* (Bouman et al. (2020); Fan et al. (2021); Ubellacker et al. (2021); Xiong (2021)). However, these models require robustification to disturbances (e.g. to compensate for the gap between the reduced and full order models) to function reliably on these complex systems (Thieffry et al. (2018); Kim et al. (2020); Alan et al. (2021); Kolathaya and Ames (2018); Ahmadi et al. (2020)). As a result, recent studies on the robust control of nonlinear systems center around input-to-state-safe control (Kolathaya and Ames (2018); Romdlony and Jayawardhana (2016); Taylor et al. (2020)) and risk-aware control (Ahmadi et al. (2020); Lindemann et al. (2021); Majumdar and Pavone (2020); Dixit et al. (2021); Akella et al. (2022a)) among other techniques. These methods typically assume *apriori* knowledge of a model and possible disturbances (or at least the magnitude thereof) and employ control techniques designed to reject those known disturbances. On the other hand, learning-based approaches attempt to identify the underlying model (Buisson-Fenet et al. (2020); Nguyen-Tuong and Peters (2011); Jain et al. (2018); Berkenkamp and Schoellig (2015); Folkestad et al. (2022); Westenbroek et al. (2021); Wang et al. (2018)), in many cases through Gaussian Process Regression (GPR) (Williams and Rasmussen (2006)).

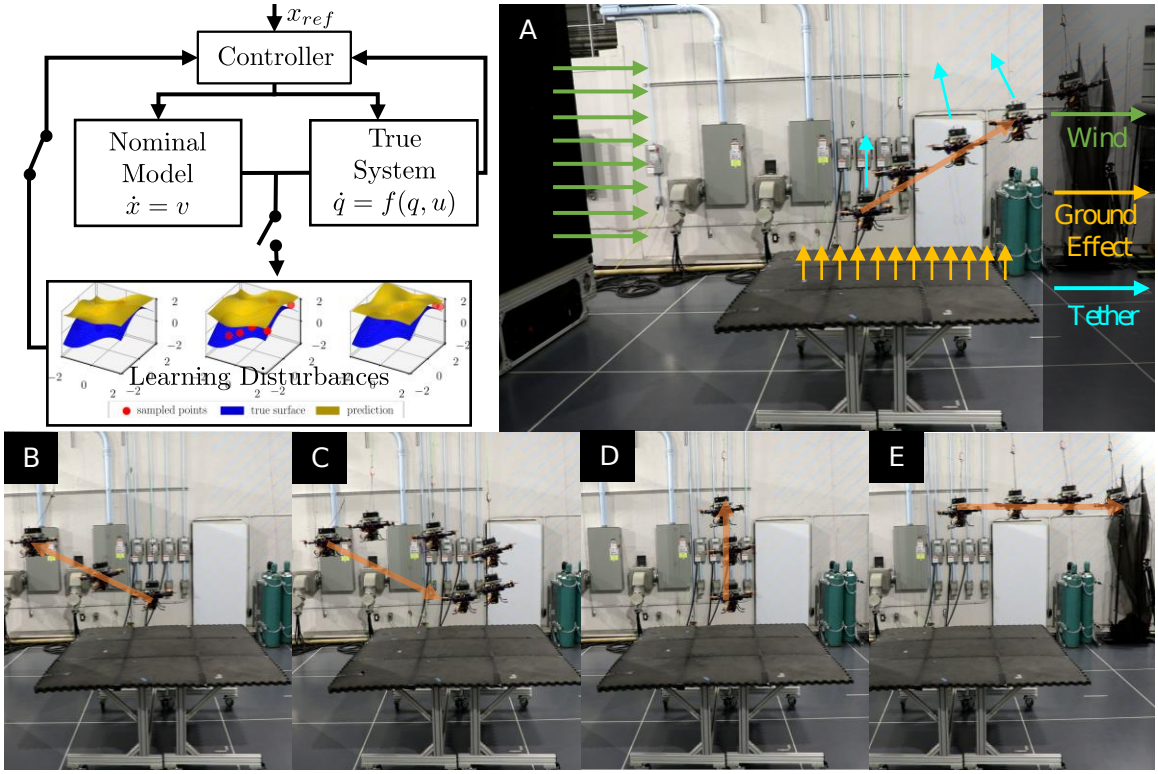

Figure 1: (Top Left) A general overview of our procedure, (Top Right) a photo of our experimental setup, and (Bottom) snippets of flight paths taken by the drone during the second set of experiments run — the experiments depicted on the left in Figure 3. Our procedure has two parts. First, we implement a nominal controller and calculate norm discrepancies between predicted model evolution and true system evolution. Then, we fit, via gaussian process regression, a risk-aware disturbance model for the disturbances that the nominal system experiences. We show in Section 4 how our procedure dramatically improves baseline controller performance and provide a statement on the theoretical accuracy of our model in Section 3.

However, assuming *apriori* knowledge of disturbances might not be accurate in real-world settings, and gaussian process regression for model determination tends to be sample-complex and only uncover expected system behavior. While learning expected behavior is indeed useful, control predicated on expected models of system behavior might yield problematic behavior in safety-critical settings where risk-sensitive approaches are preferable (Ahmadi et al. (2021); Ono et al. (2018)). Skipping the model identification step, recent work in Bayesian Optimization and Reinforcement Learning aims to identify such risk-aware policies in a model-free fashion (Cakmak et al. (2020); Makarova et al. (2021); Heger (1994); Chow et al. (2017); Mihatsch and Neuneier (2002); Geibel and Wysotzki (2005)). However, these prior works assume an ability to sample disturbances directly, assume *apriori* knowledge of disturbances, or are sample-complex.

Our Contribution: We propose a risk-aware model augmentation approach via learning disturbance models online that does not require *apriori* disturbance knowledge. Our approach is sample-efficient as shown in Section 4, where we require less than a minute of flight data to make risk-aware

control improvements on a drone mid-flight. Furthermore, by building off prior work (Akella et al. (2022b,a)), we both define and ensure that our learned disturbance surface is a *Surface-at-Risk* for the stochastic process accounting for the discrepancy between model and true system evolution. Hence, augmenting the controller with our learned disturbance model yields an efficient risk-aware controller as we demonstrate experimentally.

Structure: Section 2.1 provides a brief background on gaussian process regression, and Section 2.2 formally defines a *Surface-at-Risk* for a stochastic process. Section 3 presents the problem of upper-bounding such a surface and provides a theoretical statement on the accuracy of our procedure with respect to identifying such an upper bound. Finally, Section 4 showcases the utility of our procedure for risk-aware control of a drone with online disturbance learning.

2. Mathematical Preliminaries and Definitions

2.1. A Brief Aside on Gaussian Process Regression

A key concept in our approach is the notion of *Surfaces-at-Risk* which we fit via GPR as part of our procedure. GPR typically assumes the existence of an unknown function $f : X \to \mathbb{R}$ that we aim to represent by taking noisy samples y of f at points $x \in X$ where the noise ξ is typically assumed to be sub-Gaussian (Srinivas et al. (2009); Chowdhury and Gopalan (2017); Williams and Rasmussen (2006)). Let $\mathbb{X} = \{x_i\}_{i=1}^N$ be a set of N points $x \in X$ and $\mathbb{Y}$ be the corresponding set of noisy observations, *i.e.* $\mathbb{Y} = \{y_i = f(x_i) + \xi,\ \forall\ x_i \in \mathbb{X}\}$. Furthermore, let $k : X \times X \to \mathbb{R}$ be a positive-definite *kernel function*. Then, a *gaussian process* is uniquely defined by its mean function $\mu : X \to \mathbb{R}$ and its variance function $\sigma : X \to \mathbb{R}$. These functions are defined as follows, with $k_N(x) = [k(x, x_i)]_{x_i \in \mathbb{X}}$, $\mathbb{K} = [k(x_i, x_j)]_{x_i, x_j \in \mathbb{X}}$, $y_{1:N} = [y_i]_{y_i \in \mathbb{Y}}$, and $\lambda = (1 + \frac{2}{N})$:

$$\mu_N(x) = k_N(x)^T \left(\mathbb{K} + \lambda I_N\right)^{-1} y_{1:N}, \quad \sigma_N(x) = k_N(x, x), \tag{1}$$
$$k_N(x, x') = k(x, x') - k_N(x)^T \left(\mathbb{K}_N + \lambda I\right)^{-1} k_N(x').$$

Lastly, each kernel function has a space of functions it can reproduce to point-wise accuracy, it's Reproducing Kernel Hilbert Space (RKHS). Under the assumption that the function to-be-fitted f has bounded norm in the RKHS of the chosen kernel k, GPR guarantees high-probability representation of f as formalized in the theorem below, taken from Chowdhury and Gopalan (2017):

Theorem 1 *Let $f : X \to \mathbb{R}$, $\mathbb{X} = \{x_i\}_{i=1}^N$ be a set of N points $x \in X$, $\mathbb{Y} = \{y_i = f(x_i) + \xi\}_{x_i \in \mathbb{X}}$ be a set of noisy observations y_i of $f(x_i)$ with R sub-gaussian noise ξ, and $k : X \times X \to \mathbb{R}$ be a positive-definite kernel function. If f has B-bounded RKHS norm for some $B > 0$, i.e. $\|f\|_{RKHS} \leq B$, then, with μ_N and σ_N as per (1) and with minimum probability $1 - \delta$,*

$$|\mu_N(x) - f(x)| \leq \left(B + R\sqrt{2 \ln \frac{\sqrt{\det\left((1 + \frac{2}{N})I_N + \mathbb{K}_N\right)}}{\delta}} \right) \sigma_N(x),\ \forall\ x \in X.$$

2.2. Surfaces-at-Risk for Scalar Stochastic Processes

This section formally defines a *Surface-at-Risk* for a scalar stochastic process — the specific structure we aim to fit via GPR. Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with Ω a sample space, $\mathcal{F}$ a σ-algebra

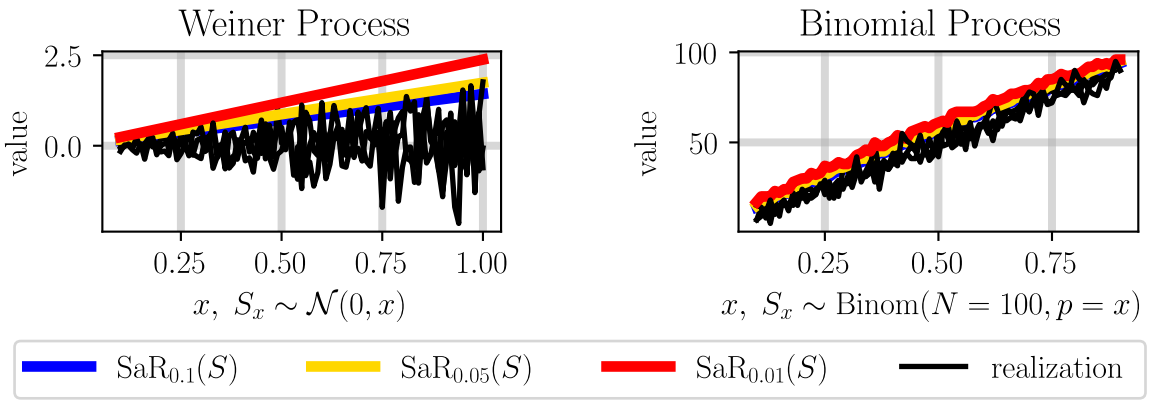

Figure 2: Example Surfaces-at-Risk at risk-levels $\epsilon \in [0.1, 0.05, 0.01]$ for a Weiner Process (Left) and Binomial Process (Right). Distributions for the indexed scalar random variables S_x comprising each process S are provided on the axes. Sample realizations of the stochastic processes are shown in black, with Surfaces-at-Risk shown via colored lines.

over Ω defining events, and $\mathbb{P}$ a probability measure, we define a scalar *stochastic process* S over the indexed space $\mathcal{X}$ as a collection of scalar random variables $S_x : \Omega \to \mathbb{R}$, *i.e.* $S = \{S_x\}_{x \in \mathcal{X}}$. Here, each scalar random variable S_x has a (perhaps) different distribution $\pi_x : \mathbb{R} \to [0, 1]$ such that probability of S_x taking values in $A \subseteq \mathbb{R}$, *i.e.* $\mathbb{P}_{\pi_x}[S_x \in A \subseteq \mathbb{R}]$, is well-defined.

Risk-measures are functions of these scalar random variables, and Value-at-Risk is a specific type of risk-measure stemming from the financial literature (Linsmeier and Pearson (2000)).

Definition 2 *The Value-at-Risk level $\epsilon \in [0, 1]$ of a scalar random variable X defined over the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with distribution π is defined as the $(1 - \epsilon) - th$ quartile of X, i.e.*

$$\text{VaR}_\epsilon(X) \triangleq c \ \text{ s.t. } \ c = \inf\{z \in \mathbb{R} \mid \mathbb{P}_\pi[X \leq z] \geq 1 - \epsilon\}.$$

Then, the *Surface-at-Risk* for a scalar stochastic process is a similar collection of the Values-at-Risk of the underlying scalar random variables constituting the scalar stochastic process.

Definition 3 *The Surface-at-Risk level $\epsilon \in [0, 1]$ of a scalar stochastic process S indexed by the set $\mathcal{X}$ is the indexed collection of the Values-at-Risk level ϵ of each random variables S_x comprising S:*

$$\text{SaR}_\epsilon(S, x) = \text{VaR}_\epsilon(S_x).$$

Figure 2 shows a few examples of Surfaces-at-Risk for varying risk-levels ϵ overlaid on realizations of common stochastic processes.

3. Learning Disturbances

3.1. The Risk-Aware Disturbance-Norm Identification Problem

From a risk-aware standpoint, we aim to identify a *Surface-at-Risk* as per Definition 3 for a scalar stochastic process S indexed over the model state-space $\hat{\mathcal{X}}$. Sample realizations of this process correspond to disturbance norms the system might experience at any given model state $\hat{x} \in \hat{\mathcal{X}}$. To

formally state this problem, we will first denote our true system via x and sim model via $\hat{x}$, *i.e.* $\forall\ k, j = 0, 1, 2, \ldots,$ (perhaps) different state and input spaces, and process noise ξ with (unknown and perhaps) state-dependent distribution π

$$\begin{aligned} &\textbf{True:} \quad x_{k+1} = f(x_k, u_k) + \xi, \quad x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ \xi \sim \pi, \\ &\textbf{Sim:} \quad \hat{x}_{j+1} = \hat{f}(\hat{x}_j, \hat{u}_j), \qquad\qquad\qquad \hat{x}_j \in \hat{\mathcal{X}},\ \hat{u}_j \in \hat{\mathcal{U}}. \end{aligned} \tag{SYS}$$

As an example consistent with the demonstration to follow, the true system would be a drone, with our reduced-order simulator model a single integrator. The true state would be the drone's position and orientation, and the true input would be the rotor torques. Meanwhile, the model state would be the drone's position in 3-space, and the model input would be the desired velocity.

To identify the discrepancy between the systems in (SYS), we define two maps - M_x which projects the true state x to the model state $\hat{x}$ and M_u which extends the model input $\hat{u}$ to the true input u, *e.g.* M_u provides rotor torques to realize the desired velocity in 3-space:

$$M_x : \mathcal{X} \to \hat{\mathcal{X}}, \quad M_u : \hat{\mathcal{U}} \times \mathcal{X} \to \mathcal{U}. \tag{MAPS}$$

To note, we only assume the existence of these maps and the ability to use them, we do not assume that they are unique, we know their analytic form, *etc*. To put these maps in the context of our drone example, the drone's underlying controller operates at 1 kHz making the true-system time step 1 ms. Since we aim to provide model inputs at 50 Hz, $K = 20$. M_x is just the projection of our drone's position in 3-space, and M_u is the on-board controller that takes in a commanded 3-space velocity — model input $\hat{u}$ — and updates rotor speeds at 1 kHz to achieve that velocity. These maps will be further explained in Section 4. Finally, we assume that after some amount of true system time-steps $K > 0$, we can observe projected true system evolution. We denote K as the *time-dilation parameter* and the observation function O is defined as follows:

$$x_{k+1} = f(x_k, M_u(\hat{u}, x_k)),\ O(x_0, \hat{u}) = M_x(x_K). \tag{OBS}$$

These maps let us formally state the projected evolution of our true system, *i.e.* evolution of $\hat{x}_j = M_x(x_{Kj})$, when driven by a feedback controller $U : \hat{\mathcal{X}} \to \hat{\mathcal{U}}$. Comparing projected and sim model evolution results in the discrepancy d we aim to learn:

$$\hat{x}_{j+1} = \hat{f}(M_x(x_{Kj}), U(M_x(x_{Kj}))) + \underbrace{O(x_{Kj}, U(M_x(x_{Kj}))) - \hat{f}(M_x(x_{Kj}), U(M_x(x_{Kj})))}_{d,\text{ and }\ \delta = \|d\| \text{ has distribution } \pi_{\hat{x}} : \mathbb{R} \to [0,1]}. \tag{2}$$

Then, inspired by input-to-state-safe barrier and input-to-state-stable Lyapunov works whose robust controllers only require information on the 2-norm of this disturbance d, we aim to learn a probabilistic upper bound on $\|d\|$ by taking samples of indexed random variables $S_{\hat{x}}$ comprising a disturbance-norm stochastic process S indexed by $\hat{\mathcal{X}}$ as in (SYS).

Definition 4 *The disturbance-norm stochastic process $S = \{S_{\hat{x}}\}_{\hat{x} \in \hat{\mathcal{X}}}$ where samples of each random variable $S_{\hat{x}}$ correspond to norms δ of disturbances d as defined in equation (2). The variability in norm samples δ arises through the assumed process noise ξ in the true system dynamics in (SYS).*

Remark on Residuals: If we only consider a deterministic discrepancy between the true and sim models, then the disturbances d as per (2) would correspond to residual dynamics, and our procedure would fit a surface to the norm of the residual dynamics (learning residual dynamics has a

well-studied history, see Saveriano et al. (2017); Johannink et al. (2019); Schperberg et al. (2022); Zeng et al. (2020) and citations within). The discrepancy between these approaches and ours is that we also learn a probabilistic bound on the norm of any stochastic, model-state-dependent disturbances that affect the system during operation. This is why we represent the discrepancies as a stochastic process and fit a Surface-at-Risk, which provides a natural way to reason about risk-aware disturbance rejection in a context including model errors and stochastic uncertainty.

Furthermore, we assume our disturbance-norm stochastic process is indexed over the model state space $\hat{\mathcal{X}}$ as opposed to the true state space $\mathcal{X}$ as we only assume the ability to measure the projected state $\hat{x}_j = M_x(x_{Kj})$. Therefore, we can only correspond sampled disturbance norms δ to points in the projected state space $\hat{\mathcal{X}}$. Then, our goal is to identify a "close" upper bound to the *Surface-at-Risk* for this disturbance-norm stochastic process at some risk-level $\epsilon \in [0, 1]$.

Problem 1 *Identify an upper bound to the Surface-at-Risk at some risk-level $\epsilon \in [0, 1]$ for the disturbance-norm stochastic process S as per Definition 4 with Surfaces-at-Risk as defined in Definition 3. Specifically, identify an estimate $\mathbb{SR}_\epsilon$ such that,*

$$\mathbb{SR}_\epsilon(S, \hat{x}) \geq \mathrm{SaR}_\epsilon(S, \hat{x}),\ \forall\ \hat{x} \in \hat{\mathcal{X}}. \tag{3}$$

While the aforementioned upper bound $\mathbb{SR}_\epsilon$ could be arbitrarily large and satisfy (3), we aim to find a "close" upper bound to the true Surface-at-Risk level ϵ to facilitate risk-aware control.

3.2. Fitting a Disturbance-Norm Surface-at-Risk

For identifying such an upper bound $\mathbb{SR}_\epsilon$, we first note that even for stochastic processes whose sample realizations are non-differentiable, their Surfaces-at-Risk are relatively smooth — see Figure 2 for examples. Intuitively, we expect the disturbance norms δ_i, δ_j at "close" model states $\hat{x}_i, \hat{x}_j \in \hat{\mathcal{X}}$ are similarly "close":

Assumption 1 *For the disturbance-norm stochastic process S in Definition 4, the Surface-at-Risk at a given risk-level $\epsilon \in [0, 1]$ has bounded discrepancy. I.e. $\exists\ \alpha, \beta \in \mathbb{R}_{\geq 0}$ such that,*

$$\forall\ \hat{x}_i, \hat{x}_j \in \hat{\mathcal{X}},\ \|\hat{x}_i - \hat{x}_j\| \leq \alpha \implies |\,\mathrm{SaR}_\epsilon(S, \hat{x}_i) - \mathrm{SaR}_\epsilon(S, \hat{x}_j)| \leq \beta.$$

Notably, this assumption only implies a bounded discrepancy, and not continuity, *e.g.* a bounded piecewise continuous function would have bounded variance as per our assumption. We will verify that this assumption holds for the data set we collect in Section 4.

Second, we need to take (perhaps noisy) unbiased samples of $\mathbb{SR}_\epsilon(S, \hat{x})$ for a given model state $\hat{x} \in \hat{\mathcal{X}}$. By equation (3), $\mathbb{SR}_\epsilon(S, \hat{x}) \geq \mathrm{VaR}_\epsilon(S_{\hat{x}})$, and we can define one sample δ_j of $S_{\hat{x}_j}$ as follows, where O is as per (OBS), and M_x is as per (MAPS):

$$\delta_j = \|O(x_{Kj}, U(M_x(x_{Kj})) - \hat{f}(M_x(x_{Kj}), U(M_x(x_{Kj}))\|, \quad \hat{x}_j = M_x(x_{Kj}). \tag{4}$$

Then, we can group multiple samples δ_j for sequential model states visited during operation, *i.e.* $\delta_j, \delta_{j+1}, \ldots$ for $\hat{x}_j, \hat{x}_{j+1}, \ldots$ to produce an upper bound to at least one Value-at-Risk level ϵ of a sampled random variable, *i.e.* $\mathrm{VaR}_\epsilon(S_{\hat{x}_j}), \mathrm{VaR}_\epsilon(S_{\hat{x}_{j+1}}), \ldots$. To do so, we require the following proposition, stated for N scalar random variables X with (perhaps) different distributions π.

Algorithm 1: Fitting a Disturbance-Norm Surface-at-Risk

Data: α, β for Assumption 1, an integer $\mathrm{N_{RV}} > 0$ for Proposition 2 corresponding to the number of random variables to sample, time-step dilation parameter $K > 0$ between true system evolution and model evolution as per (OBS), and $k : \hat{\mathcal{X}} \times \hat{\mathcal{X}} \to \mathbb{R}$ a kernel function

Initialize: $\mathrm{s} = 0$, $\mathbb{X} = []$, $\mathbb{Y} = []$

References: Disturbance Norm samples δ_j as per (4) and projector M_x as per (MAPS)

while *True* **do**

  Initialize empty data-set, *i.e.* $\mathcal{D}_\mathrm{s} = [\,]$

  for $j = \mathrm{N_{RV}} \cdot \mathrm{s}, \mathrm{N_{RV}} \cdot \mathrm{s}+1, \ldots, \mathrm{N_{RV}}(\mathrm{s}+1) - 1$ **do**

    Collect state-indexed disturbance norm samples, *i.e.* $\mathcal{D}_\mathrm{s} \leftarrow \mathcal{D}_\mathrm{s} \cup (\delta_j, \hat{x}_j = M_x(x_{Kj}))$

  end

  Augment GP state dataset with $\mathcal{D}_\mathrm{s}$: $\mathbb{X} \leftarrow \mathbb{X} \cup \hat{x}_{\mathrm{N_{RV}}(\mathrm{s}+1)-1}$

  Augment GP norm dataset with $\mathcal{D}_\mathrm{s}$: $\mathbb{Y} \leftarrow \mathbb{Y} \cup \max\{\delta_\ell \in \mathcal{D}\} + \beta$

  Fit $\mu_\mathrm{s}, \sigma_\mathrm{s}$ as per (1) with data sets $\mathbb{X}, \mathbb{Y}$. $\mathrm{s}{+}{+}$

end

Proposition 2 *Let $\{X_i\}_{i=1}^N$ be a collection of N scalar random variables with (perhaps) different distributions $\{\pi_i\}_{i=1}^N$, and let $\{x_i\}_{i=1}^N$ be a set of N samples of these random variables, one sample per each random variable, i.e. x_i is a sample of X_i. Then, for any $\epsilon \in [0,1]$, the probability that at least one sample $x_\ell \in \{x_i\}_{i=1}^N$ is greater than the Value-at-Risk level ϵ of its corresponding random variable X_ℓ is equivalent to $1-(1-\epsilon)^N$, i.e. with* VaR *as per Definition 2 and $\forall\ \epsilon \in [0,1]$,*

$$\mathbb{P}_{\pi_1,\pi_2,\ldots,\pi_N}\left[\exists\, x_\ell \in \{x_i\}_{i=1}^N \text{ s.t. } x_\ell \geq \mathrm{VaR}_\epsilon(X_\ell)\right] \geq 1-(1-\epsilon)^N.$$

Proof: Consider a random variable $X_\ell \in \{X_i\}_{i=1}^N$. The probability of taking a sample x_ℓ of X_ℓ such that $x_\ell \geq \mathrm{VaR}_\epsilon(X_\ell)$ is less than or equal to ϵ by Definition 2. The same line of reasoning holds $\forall\ X_\ell \in \{X_i\}_{i=1}^N$. As such, the probability that no sample $x_\ell \in \{x_i\}_{i=1}^N$ is greater than the corresponding Value-at-Risk level ϵ is less than or equal to $(1-\epsilon)^N$, yielding our result. ■

Our procedure for generating unbiased samples of the upper bound $\mathbb{SR}_\epsilon$ stems directly from Proposition 2 and Assumption 1. First, we let the system evolves for $\mathrm{N_{RV}}$ model time-steps and collect one norm sample δ_j per model state $\hat{x}_j$ visited during operation. This norm sample δ_j is calculated as per (4). Second, Proposition 2 guarantees that the largest norm sample δ_j^* is greater than the Value-at-Risk level ϵ for its corresponding indexed random variable $S_{\hat{x}_j^*}$ with some minimum probability. Third, if all norm samples were drawn from indexed random variables $S_{\hat{x}_j}$ whose indices $\hat{x}_j$ were "close", *i.e.* $\|\hat{x}_s - \hat{x}_r\| \leq \alpha\ \forall\ \hat{x}_r \neq \hat{x}_s \in \{\hat{x}_{j+i}\}_{i=0}^{N-1}$ and for some $\alpha > 0$, we can use Assumption 1 to augment the largest norm sample δ_j^* by a constant $\beta > 0$. The sum is, with minimum probability $1-(1-\epsilon)^N$, an unbiased, non-noisy sample of $\mathbb{SR}_\epsilon(S, \hat{x}_j)$. Algorithm 1 formalizes this procedure and our main theoretical result follows.

Theorem 2 *Let $\alpha, \beta, \mathrm{N_{RV}}, \mathrm{s}, \mu_\mathrm{s}, \sigma_\mathrm{s}$, and k be as defined in Algorithm 1, let $B > 0$, let* SaR *be the Surface-at-Risk measure as per Definition 3 for some risk-level $\epsilon \in [0,1]$, let S be the disturbance-norm stochastic process as per Definition 4, and let Assumption 1 hold for each data set $\mathcal{D}_\mathrm{s}$ in lines 5-7 of Algorithm 1 with respect to the given parameters α, β. If $\|\mathbb{SR}_\epsilon(S)\|_{RKHS} \leq B$, then with minimum probability $\left(1-(1-\epsilon)^{\mathrm{N_{RV}}}\right)^\mathrm{s}$ the following holds $\forall\ \hat{x} \in \hat{\mathcal{X}}$ and $\forall\ \mathrm{s} = 1, 2, \ldots$:*

$$|\mu_\mathrm{s}(\hat{x}) - \mathbb{SR}_\epsilon(S, \hat{x})| \leq B\sigma_\mathrm{s}(\hat{x}), \qquad \mu_\mathrm{s}(\hat{x}) + B\sigma_\mathrm{s}(\hat{x}) \geq \mathrm{SaR}_\epsilon(S, \hat{x}).$$

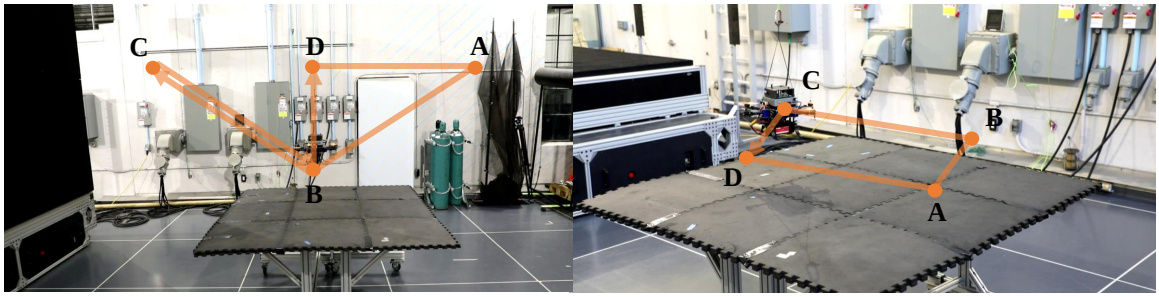

Figure 3: Depictions of the two types of periodic trajectories implemented in our drone experiments described in Section 4. These trajectories approximate difficult types of behaviors commonly asked of drones,

Proof: First, by the assumptions above, we know that for each data set $\mathcal{D}_{\mathrm{s}}$ in lines 5-7 of Algorithm 1, we have taken one sample δ_j of $\mathrm{N_{RV}}$ (potentially) different random variables $S_{\hat{x}_j}$. By Proposition 2, we know that with minimum probability $1-(1-\epsilon)^{\mathrm{N_{RV}}}$, the maximum sample $\delta_j^* \triangleq \max\{\delta_\ell \in \mathcal{D}\}$ is greater than the Value-at-Risk of its corresponding random variable $\mathrm{VaR}_\epsilon(S_{\hat{x}_j^*})$ (VaR is defined in Definition 2). Since we assume Assumption 1 holds for each such set of random variables, then we know that with minimum probability $1-(1-\epsilon)^{\mathrm{N_{RV}}}$, the sum $\delta_j^*+\beta$ is greater than the value-at-risk level ϵ of any sampled random variable, *i.e.* the sum $\delta_j^*+\beta$ is a non-noisy estimate of $\mathbb{SR}_\epsilon(S,\hat{x})$, $\forall\, \hat{x} \in \mathcal{D}_{\mathrm{s}}$. Hence, repeating this same argument for each data point in $\mathbb{X}, \mathbb{Y}$ and setting $R=0$, as each sampled point is a non-noisy sample of our upper-bounding surface, we recover the results of Theorem 1 with minimum probability $(1-(1-\epsilon)^{\mathrm{N_{RV}}})^{\mathrm{s}}$:

$$|\mu_{\mathrm{s}}(\hat{x}) - \mathbb{SR}_\epsilon(S,\hat{x})| \leq B\sigma_{\mathrm{s}}(\hat{x}),\ \forall\, \hat{x} \in \hat{\mathcal{X}}. \tag{5}$$

Our final result holds by unraveling the absolute-value inequality in (5), as $\mathbb{SR}_\epsilon(S)$ is an upper-bounding surface for $\mathrm{SaR}_\epsilon(S)$. ■

4. Learning Disturbances Mid-Flight for Risk-Aware Control

4.1. Implementation Specifics

All flight tests are performed at the Caltech Center for Autonomous Systems and Technology arena which is equipped with an Optitrack motion capture system that samples and streams the rotor-craft pose at 190 Hz. We belay a safeguard tether to the drone (weights 2.46 kg) with a $\sim$200 g passive weight attached on the other end to partially eliminate tether slack, which is another source of uncertainty. Figure 3 depicts the two types of flight paths taken, wherein we aimed to realize complex behaviors commonly asked of drones, *e.g.* ascent and descent with both headwind and tailwind, circulating low to the ground, and taking off vertically in the presence of transverse wind. All disturbing winds were realized by The Caltech Real Weather Wind Tunnel, and windspeed information was not made available to the baseline controller to-be-augmented. This baseline controller was developed against a single integrator model, and as such, it outputs 3-space velocities at 50 Hz for the drone to follow. The velocities provided by this controller are tracked by the drone's onboard flight controller, a Hex Cube Orange running a PX4 autopilot Meier et al. (2015).

With respect to the mathematical setting in Section 3.1 then, we do not know our true system dynamics, though we model the system as a single integrator:

$$\hat{x}_{j+1} = \hat{x}_j + \hat{u}_j(\Delta t = 0.02),\ \hat{x}_j \in \underbrace{[-2,2]^2 \times [1.2,2]}_{\hat{\mathcal{X}}},\ \hat{u}_j \in \underbrace{[-0.8,0.8]^2 \times [-0.5,0.5]}_{\hat{\mathcal{U}}}.$$

The state projection map M_x as in (MAPS) reads the drone's position in 3-space. The input map M_u corresponds to the onboard PX4 controller that maps true drone states $x \in \mathcal{X}$ and commanded 3-space velocities $\hat{u} \in \hat{\mathcal{U}}$ to rotor speeds at 1 kHz. As we update these desired velocities at 50 Hz, our time-dilation parameter $K = 20$ for Algorithm 1. Finally, our observation function O as per (OBS) outputs the projected true-system 3-space position after K true-system time-steps, and our disturbance-norm samples δ as per (4) are defined as follows:

$$\delta_j = \|O(x_{Kj}, U(M_x(x_{Kj})) - (\hat{x}_j + U(M_x(x_{Kj}))\Delta t\|, \quad \hat{x}_j = M_x(x_{Kj}).$$

The baseline controller $U : \hat{\mathcal{X}} \to \hat{\mathcal{U}}$ is a discrete-time Lyapunov controller designed to send the single-integrator system to a provided waypoint, and does not take into account complex aerodynamic effects, *e.g.* ground effects, transverse wind, and tethered disturbances, which are challenging to model and can degrade flight performance when ignored (O'Connell et al. (2022); Folkestad et al. (2022)). Furthermore, the number of random variables sampled per data-collection step $\mathrm{N_{RV}} = 60$ was kept constant, and we used the squared-exponential kernel function with length-scale parameter $\ell = 1.0$ for all experiments as well.

Our desired outcomes were twofold. First, we fit an upper bound to the disturbance-norm *Surface-at-Risk* level $\epsilon = 0.05$ over the course of one traversal of the desired flight path. In this initial flight path, we only implement the baseline controller and augment this controller if the system takes longer than 10 seconds to reach within 0.1 m of the subsequent waypoint along the desired path. As each path comprises fewer than 6 waypoints, this ensures that our learned model considers less than a minute of data for all experiments on both flight paths. These cutoff times were specifically chosen to highlight the efficiency of our method with limited data. Second, on all subsequent flight paths, we provide from our fitted surface the norm of disturbances that the Lyapunov controller should reject while providing velocity commands. As such, we expect performance improvements from our augmented controller in the form of traversal time speedups through the series of waypoints, as subsequent waypoints are provided once the drone reaches within 0.1 m of the current, commanded waypoint, and the drone's controller should account for the vast majority of disturbances caused by wind, ground, and tether effects as we fitted an upper bound to the disturbance-norm *Surface-at-Risk* level $\epsilon = 0.05$.

4.2. Discussion of Results

We performed four sets of experiments: (A) Hovering and moving while maintaining a 0.15 m height above ground (see right in Figure 3); (B) Ascent, descent, and vertical take-off without any wind (see left in Figure 3); (C) The same flight path as (B) but with a 0.6 m/s transverse wind. The wind flows from left to right when looking at the setup in Figure 3. A graphical example is also provided in Figure 1; (D) The same flight path as (B) and (C) but with a 2 m/s transverse wind.

Figure 4 shows the fitted $\mathrm{SaR}_{\epsilon=0.05}$ for each of the four experiments (A)-(D) ran on the drone, as labeled prior. As mentioned, in all cases we see at least a $2\times$ speedup in flight path times when

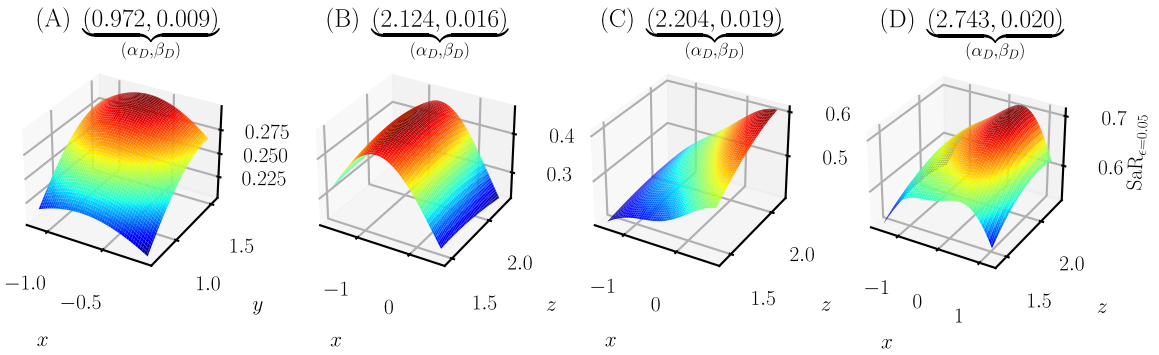

Figure 4: Fitted $\mathrm{SaR}_{\epsilon=0.05}$ for the four experiments depicted in Figure 3, with α_D the maximum distance between two sampled states for GPR, and β_D the maximum discrepancy between two sampled disturbance norms. Over all four experiments, we see a consistent $2\times$ speedup in flight path times after implementation of the augmented controller — a qualitative result we expect as per Theorem 2, as we fit an upper bound to disturbance norms at 95% probability. This information is further explained in Section 4.2.

implementing the augmented controller, with as much as a $5\times$ speedup in the hovering case (A). Furthermore, we were also able to verify Assumption 1 with respect to the data sets we collected for each experiment. Specifically, for (A), we assumed that for states within $\alpha = 1$m their Values-at-Risk level $\epsilon = 0.05$ would not change by more than $\beta = 0.05$. As can be seen in the title of the associated subfigure in Figure 4, the reported values from data are smaller than their assumed counterparts, indicating that Assumption 1 held over this experiment, at least with respect to the collected data. For the remaining experiments, the assumed α, β values were as follows: (B) $\alpha = 3$m, $\beta = 0.05$; (C) $\alpha = 3$m, $\beta = 0.1$; (D) $\alpha = 3$m, $\beta = 0.2$. Therefore, as we can see from the associated titles in Figure 4, we are similarly able to verify that Assumption 1 held over each of these cases as well — at least with respect to the data collected. As such, we expected a significant increase in performance according to Theorem 2 as was realized in all four cases with respect to flight path time speedups. All experiments can also be seen in our supplementary video here: vid.

5. Concluding Remarks and Future Work

Our results were threefold. First, we defined Surfaces-at-Risk, an extension of Value-at-Risk to scalar-valued stochastic processes. Second, we defined the discrepancy between simulator and true-system evolution as a stochastic process, and provided a method to fit an upper bound to this process's *Surface-at-Risk*. Third, we provided a theoretical statement on the accuracy of our proposed approach with respect to fitting such an upper bound. Finally, we showcased the utility of our procedure in facilitating risk-aware control by implementing our procedure on a drone mid-flight and exhibiting dramatic performance improvements as a result. In future work, we hope to integrate our procedure with existing works in safety-critical control, to create a pipeline for online, adaptive, safety-critical risk-aware control.

Acknowledgments

The work of Prithvi Akella was supported by the Air Force Office of Scientific Research, grant FA9550-19-1-0302, and the National Science Foundation, grant 1932091. The work of Skylar Wei was supported in part by DARPA, through the Learning and Introspective Control program. We would also like to thank the Caltech Center for Autonomous Systems and Technologies for the use of the wind tunnel in our experiments.

References

Video. URL https://youtu.be/4i2GNU8ahSU.

Mohamadreza Ahmadi, Xiaobin Xiong, and Aaron D Ames. Risk-sensitive path planning via cvar barrier functions: Application to bipedal locomotion. *arXiv preprint arXiv:2011.01578*, 2020.

Mohamadreza Ahmadi, Ugo Rosolia, Michel D Ingham, Richard M Murray, and Aaron D Ames. Constrained risk-averse markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11718–11725, 2021.

Prithvi Akella, Anushri Dixit, Mohamadreza Ahmadi, Joel W Burdick, and Aaron D Ames. Sample-based bounds for coherent risk measures: Applications to policy synthesis and verification. *arXiv preprint arXiv:2204.09833*, 2022a.

Prithvi Akella, Wyatt Ubellacker, and Aaron D Ames. Safety-critical controller verification via sim2real gap quantification. *arXiv preprint arXiv:2209.09337*, 2022b.

Anil Alan, Andrew J Taylor, Chaozhe R He, Gábor Orosz, and Aaron D Ames. Safe controller synthesis with tunable input-to-state safe control barrier functions. *IEEE Control Systems Letters*, 6:908–913, 2021.

Felix Berkenkamp and Angela P Schoellig. Safe and robust learning control with gaussian processes. In *2015 European Control Conference (ECC)*, pages 2496–2501. IEEE, 2015.

Amanda Bouman, Muhammad Fadhil Ginting, Nikhilesh Alatur, Matteo Palieri, David D Fan, Thomas Touma, Torkom Pailevanian, Sung-Kyun Kim, Kyohei Otsu, Joel Burdick, et al. Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2518–2525. IEEE, 2020.

Mona Buisson-Fenet, Friedrich Solowjow, and Sebastian Trimpe. Actively learning gaussian process dynamics. In *Learning for dynamics and control*, pages 5–15. PMLR, 2020.

Sait Cakmak, Raul Astudillo Marban, Peter Frazier, and Enlu Zhou. Bayesian optimization of risk measures. *Advances in Neural Information Processing Systems*, 33:20130–20141, 2020.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1):6070–6120, 2017.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.

Anushri Dixit, Mohamadreza Ahmadi, and Joel W Burdick. Risk-sensitive motion planning using entropic value-at-risk. In *2021 European Control Conference (ECC)*, pages 1726–1732. IEEE, 2021.

David D Fan, Kyohei Otsu, Yuki Kubo, Anushri Dixit, Joel Burdick, and Ali-Akbar Agha-Mohammadi. Step: Stochastic traversability evaluation and planning for risk-aware off-road navigation. *arXiv preprint arXiv:2103.02828*, 2021.

Carl Folkestad, Skylar X Wei, and Joel W Burdick. Koopnet: Joint learning of koopman bilinear models and function dictionaries with application to quadrotor trajectory tracking. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1344–1350. IEEE, 2022.

Peter Geibel and Fritz Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 24:81–108, 2005.

Matthias Heger. Consideration of risk in reinforcement learning. In *Machine Learning Proceedings 1994*, pages 105–111. Elsevier, 1994.

Achin Jain, Truong Nghiem, Manfred Morari, and Rahul Mangharam. Learning and control using gaussian processes. In *2018 ACM/IEEE 9th international conference on cyber-physical systems (ICCPS)*, pages 140–149. IEEE, 2018.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6023–6029. IEEE, 2019.

Youngmin Kim, Richard Allmendinger, and Manuel López-Ibáñez. Safe learning and optimization techniques: Towards a survey of the state of the art. In *International Workshop on the Foundations of Trustworthy AI Integrating Learning, Optimization and Reasoning*, pages 123–139. Springer, 2020.

Shishir Kolathaya and Aaron D Ames. Input-to-state safety with control barrier functions. *IEEE control systems letters*, 3(1):108–113, 2018.

Lars Lindemann, George J Pappas, and Dimos V Dimarogonas. Reactive and risk-aware control for signal temporal logic. *IEEE Transactions on Automatic Control*, 2021.

Thomas J Linsmeier and Neil D Pearson. Value at risk. *Financial Analysts Journal*, 56(2):47–67, 2000.

Anirudha Majumdar and Marco Pavone. How should a robot assess risk? towards an axiomatic theory of risk in robotics. In *Robotics Research*, pages 75–84. Springer, 2020.

Anastasia Makarova, Ilnura Usmanova, Ilija Bogunovic, and Andreas Krause. Risk-averse heteroscedastic bayesian optimization. *Advances in Neural Information Processing Systems*, 34: 17235–17245, 2021.

Lorenz Meier, Dominik Honegger, and Marc Pollefeys. Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6235–6240, 2015. doi: 10.1109/ICRA.2015.7140074.

Oliver Mihatsch and Ralph Neuneier. Risk-sensitive reinforcement learning. *Machine learning*, 49 (2):267–290, 2002.

Duy Nguyen-Tuong and Jan Peters. Model learning for robot control: a survey. *Cognitive processing*, 12(4):319–340, 2011.

Masahiro Ono, Matthew Heverly, Brandon Rothrock, Eduardo Almeida, Fred Calef, Tariq Soliman, Nathan Williams, Hallie Gengl, Takuto Ishimatsu, Austin Nicholas, et al. Mars 2020 site-specific mission performance analysis: Part 2. surface traversability. In *2018 AIAA SPACE and Astronautics Forum and Exposition*, page 5419, 2018.

Michael O'Connell, Guanya Shi, Xichen Shi, Kamyar Azizzadenesheli, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural-fly enables rapid learning for agile flight in strong winds. *Science Robotics*, 7(66):eabm6597, 2022. doi: 10.1126/scirobotics.abm6597.

Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. On the new notion of input-to-state safety. In *2016 IEEE 55th conference on decision and control (CDC)*, pages 6403–6409. IEEE, 2016.

Matteo Saveriano, Yuchao Yin, Pietro Falco, and Dongheui Lee. Data-efficient control policy search using residual dynamics learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4709–4715. IEEE, 2017.

Alexander Schperberg, Yusuke Tanaka, Feng Xu, Marcel Menner, and Dennis Hong. Real-to-sim: Deep learning with auto-tuning to predict residual errors using sparse data. *arXiv preprint arXiv:2209.03210*, 2022.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for safety-critical control with control barrier functions. In *Learning for Dynamics and Control*, pages 708–717. PMLR, 2020.

Maxime Thieffry, Alexandre Kruszewski, Christian Duriez, and Thierry-Marie Guerra. Control design for soft robots based on reduced-order model. *IEEE Robotics and Automation Letters*, 4 (1):25–32, 2018.

Wyatt Ubellacker, Noel Csomay-Shanklin, Tamas G Molnar, and Aaron D Ames. Verifying safe transitions between dynamic motion primitives on legged robots. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8477–8484. IEEE, 2021.

Li Wang, Evangelos A Theodorou, and Magnus Egerstedt. Safe learning of quadrotor dynamics using barrier certificates. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2460–2465. IEEE, 2018.

Tyler Westenbroek, Ayush Agrawal, Fernando Castañeda, S Shankar Sastry, and Koushil Sreenath. Combining model-based design and model-free policy optimization to learn safe, stabilizing controllers. *IFAC-PapersOnLine*, 54(5):19–24, 2021.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Xiaobin Xiong. *Reduced Order Model Inspired Robotic Bipedal Walking: A Step-to-step Dynamics Approximation based Approach*. PhD thesis, California Institute of Technology, 2021.

Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4): 1307–1319, 2020.