Predictive Monitoring of Black-Box Dynamical Systems

Thomas A. Henzinger[1] TAH@IST.AC.AT
Fabian Kresse[1] FABIAN.KRESSE@IST.AC.AT
Kaushik Mallik[2] KAUSHIK.MALLIK@IMDEA.ORG
Emily Yu[1] EMILY.YU@IST.AC.AT
Đorđe Žikelić[3] DZIKELIC@SMU.EDU.SG
[1] *Institute of Science and Technology Austria, Klosterneuburg, Austria*
[2] *IMDEA Software Institute, Madrid, Spain*
[3] *Singapore Management University, Singapore, Singapore*

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

We study the problem of predictive runtime monitoring of black-box dynamical systems with quantitative safety properties. The black-box setting stipulates that the exact semantics of the dynamical system and the controller are unknown, and that we are only able to observe the state of the controlled (aka, closed-loop) system at finitely many time points. We present a novel framework for predicting future states of the system based on the states observed in the past. The numbers of past states and of predicted future states are parameters provided by the user. Our method is based on a combination of Taylor's expansion and the backward difference operator for numerical differentiation. Additionally, we provide upper bound on the prediction error when the controlled system's dynamics are smooth and the maximum magnitudes of the higher order derivatives of the trajectories are known. The predicted states are then used to predict safety violations ahead in time. Our experiments demonstrate practical applicability of our method for complex black-box systems, showing that it is computationally lightweight and yet significantly more accurate than the state-of-the-art predictive safety monitoring techniques.

Keywords: Runtime monitoring, predictive safety monitoring, control systems, black-box control

1. Introduction

A majority of autonomous systems nowadays depend on advanced artificial intelligence (AI) technologies. For instance, in self-driving cars, deep learning is routinely used to design perception modules (Janai et al., 2020) and reinforcement learning has shown great promise for designing navigation controllers (Lillicrap et al., 2016). Although revolutionary, these AI technologies are hard to analyze and may pose serious safety risks in the underlying systems (Amodei et al., 2016).

Towards the safe and trustworthy deployment of AI-powered systems, we study the problem of *predictive runtime monitoring* of continuous-time dynamical systems possibly operated by a learned controller. We consider the *black-box setting*, in which the exact semantics of the system dynamics and the controller are unknown. Rather, one is only able to observe the state of the system at finitely many sampling instances. Our goal is to design a runtime monitoring algorithm which, at each sampling point, takes the past states into account, and *predicts* the future states and resulting future safety status of the system within a given time horizon. Unlike traditional runtime monitoring concerning fulfillment of safety only in the *past* (Bartocci and Falcone, 2018),

our predictive monitors raise safety warnings *before* they actually take place, so that the system can be intervened and steered out of danger in time, e.g., by using a fail-safe backup controller as in shielding (Alshiekh et al., 2018).

Our contributions. We present *Taylor-based Predictive Monitoring (TPM)*, a new framework for predictive runtime monitoring of black-box controlled dynamical systems. For a given time t and samples of past observed states $x_{t-k\tau}, \ldots, x_{t-\tau}, x_t$ with a given sampling time τ, our goal is to predict the next h states $x_{t+\tau}, \ldots, x_{t+h\tau}$. Here, the numbers k and h are parameters that can be chosen by the user.

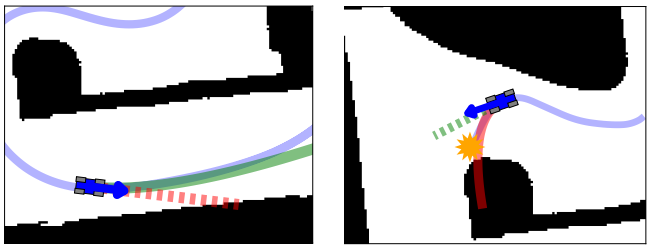

Figure 1: Predictive monitoring using our TPM (solid) and the baseline TTC (dashed). The solid blue line is the ground truth trajectory. Green and red represent predictions of, respectively, safe and unsafe behaviors within the horizon. As can be seen, TMP is more accurate in predicting smooth turns.

TPM consists of a *learning* phase followed by a *prediction* phase. In the learning phase, TPM first uses Taylor's polynomials in order to approximate the true system dynamics and the controller via a polynomial function. Taylor's polynomial expansion is a classical result in mathematical analysis that allows us to represent an arbitrary nonlinear function around a given point using polynomials with arbitrary accuracy (Rudin, 1964). The challenge is that the coefficients of Taylor's polynomials use derivatives of the function, which in our setting are unknown owing to the black-box nature of the system. To address this challenge, we then use the backward difference method (Gear, 1967) to numerically approximate the derivatives from the past state observations. By combining these two ingredients, we obtain an approximate polynomial model of the system in the vicinity of the current time. In the prediction phase, this polynomial model is used to compute the predicted future states $x_{t+\tau}, \ldots, x_{t+h\tau}$. Finally, the predictions can be used to reason about possible future violations of a safety specification of interest. TPM can reason about both *qualitative* and more general *quantitative* safety specifications; see Sec. 2 for the formal definition.

Taylor's polynomials and backward difference method are both standard in numerical analysis. However, to the best of our knowledge, their combined application to monitoring black-box controlled dynamical systems is novel. We also derive a *formal upper bound on the approximation error* induced by TPM, for when the system dynamics and the controller are $(k+1)$-times continuously differentiable functions, and the upper bounds on the first $(k+1)$ derivatives of the state trajectories are known. These bounds may be available from the domain knowledge about the physical model and by minimally accessing the AI controller (Shen et al., 2022), and the requirement of having these bounds is common to several other data-driven approaches with formal guarantees, e.g., (Bunton and Tabuada, 2024).

We implemented and experimentally evaluated TPM on two complex controlled dynamical systems. As a baseline, we compared our method to the time-to-collision (TTC) metric (Vogel, 2003),

a technique for providing runtime safety assurances of autonomous systems. TTC is routinely deployed in autonomous driving to predict on-road safety violation (Wang et al., 2021), and is defined as the time after which the vehicle will violate safety (i.e. cause "collision") if it continues with its current velocity. One can view TTC as a special case of our method where only degree 1 Taylor's polynomial is used. As shown in Fig. 1, reasoning about the first-order derivative only can be too conservative and may lead to failures in correctly predicting future safety violations. This is seen in our experiments as well, with TPM showing superior predictive power compared to TTC.

Our contributions can be summarized as follows:

1. *Predictive runtime monitoring.* We present Taylor-based Predictive Monitoring (TPM), a framework for predictive runtime monitoring of black-box controlled dynamical systems with a given quantitative safety specification. TPM is based on a combination of Taylor's polynomials and the backward difference method of numerical differentiation.
2. *Formal error bound analysis.* We provide formal error bounds for our predictions in the setting when the bounds on the higher-order derivatives of the dynamics are known.
3. *Experiments.* Empirical results demonstrate practical applicability of our framework to complex dynamical systems. TPM shows superior predictive power compared to the baseline TTC, a classical approach for runtime safety assurances of autonomous systems.

Related Work. In the formal methods literature, traditional runtime verification approaches treat the monitored system as a black-box, and output at each time point whether a given specification has been violated or fulfilled *in the past* (Bartocci and Falcone, 2018). There are works on predictive monitoring, which assume that some abstract model of the system is either available (Zhang et al., 2012; Pinisetty et al., 2017) or can be learned at runtime (Stoller et al., 2011; Ferrando and Delzanno, 2023). Our work is close to the latter, but, instead of learning a detailed general purpose model of the system, our TPM "learns" only the essential trend required to predict the future states in the vicinity of the current time. For a more comprehensive overview on data-driven monitoring, we refer to a survey (Taleb et al., 2023). Additionally, black-box monitoring has also been considered in the context of certificate-based neural control (Yu et al., 2025). As discussed in Sec. 1, time-to-collision (TTC) metric (Vogel, 2003) can be viewed as a special case of our method where only degree 1 Taylor's polynomial is used.

Numerical and data-driven inference algorithms are fundamental to many different disciplines, such as time series forecasting in economics (Hyndman, 2018), state estimation of dynamical systems from observed output sequences (Diop et al., 1994; Bunton and Tabuada, 2024), and data-driven online control (De Persis and Tesi, 2019). While our data-driven prediction algorithm has some resemblance to existing techniques, to the best of our knowledge, our work is the first to apply such techniques to the setting of runtime monitoring of controlled dynamical systems.

2. Preliminaries and Problem Statement

Controlled dynamical systems. A *controlled dynamical system*, or *system* in short, is defined via

$$\frac{dx(t)}{dt} = f(x(t), u(t)), \quad x(0) = x_0, \quad \forall t \geq 0 \,.\, u(t) = \pi(x(t)), \tag{1}$$

where $t \in \mathbb{R}_{\geq 0}$ denotes time, $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ and $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ denote the state and the control input at time t, $x_0 \in \mathcal{X}$ is the initial state, $f \colon \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is the (nonlinear) dynamics, and $\pi : \mathcal{X} \to \mathcal{U}$ is the controller which assigns a control input to each state. We will assume that

f and π are Lipschitz continuous, which is the standard assumption (Dawson et al., 2023; Szegedy et al., 2014) required for the existence and uniqueness of the solution of (1), where the solution will be called the *trajectory* of the system, denoted as $\xi\colon \mathbb{R}_{\geq 0} \to \mathbb{R}^n$. Additional assumptions on the higher-order derivatives of ξ will be introduced subsequently during prediction error analysis.

Safety properties. We consider *quantitative safety properties* which are functions of the form $\varphi\colon \mathcal{X} \to \mathbb{R}$, assigning a real valued *safety level* to each system state. We say that the controlled dynamical system under a given controller *satisfies the safety property* if $\varphi(\xi(t)) \geq 0$ for all time steps $t \in \mathbb{R}_{\geq 0}$, i.e., if the safety level remains non-negative along the trajectory. For example, if we are interested in analyzing boolean (or *qualitative*) safety violations, we define $\varphi(x) = -1$ if the state x is unsafe and $\varphi(x) = 0$ otherwise. If we are interested in a quantitative safety properties, like the value of a barrier function B (Prajna et al., 2007), then we define $\varphi(x) = B(x)$ for all states x.

Problem statement. Suppose we are given a *black-box* controlled dynamical system and a quantitative safety property φ; both the dynamics and the controller of the system are unknown but we can observe the resulting trajectory ξ. Let $\tau \in \mathbb{R}_{\geq 0}$ be a given *sampling time* and $h \in \mathbb{N}$ be a given prediction *horizon*. A *predictive runtime monitor*, or a *monitor* in short, observes the trajectory of the system at the sampling instances, and after each new observation, predicts the safety levels in the next h sampling instances. Formally, at each time $t \in \{\tau, 2\tau, 3\tau, \ldots\}$, the monitor takes the input sequence $\ldots, \xi(t-2\tau), \xi(t-\tau), \xi(t)$ in account and outputs either the sequence $\varphi(\xi(t+\tau)), \ldots, \varphi(\xi(t+h\tau))$ or a statistic thereof (e.g., the minimum φ or the first instance when φ becomes negative). We consider the problem of designing a monitor for the given safety property.

Taylor's expansion. Before presenting our monitor, we recall Taylor's polynomial of a $(l+1)$-times continuously differentiable function $g : \mathbb{R} \to \mathbb{R}$. For each $1 \leq i \leq l$, denote by $g^{(i)}$ the i-th derivative of g. For a fixed point $t \in \mathbb{R}$, the *Taylor's polynomial of g of degree l* at point t is

$$P_l(s) = g(t) + \frac{g^{(1)}(t)}{1!}(s-t) + \frac{g^{(2)}(t)}{2!}(s-t)^2 + \ldots + \frac{g^{(l)}(t)}{l!}(s-t)^l. \tag{2}$$

The following theorem is a classical result from mathematical analysis which provides an upper bound on the approximation error of a function via its Taylor's polynomial at a given point.

Theorem 1 (Taylor's theorem (Rudin, 1964)) *Suppose that $g : \mathbb{R} \to \mathbb{R}$ is an $(l+1)$-times continuously differentiable function. Let $t \in \mathbb{R}$ and let P_l be the Taylor's polynomial of g of degree l at point t. Then, for every $s \in \mathbb{R}$, there exists a point $r \in (t, s)$ such that*

$$g(s) - P_l(s) = \frac{g^{(l+1)}(r)}{(l+1)!}(s-t)^{l+1}.$$

Hence, if $B \geq \sup_{r\in(t,s)} |g^{(l+1)}(r)|$, then we have $|g(s) - P_l(s)| \leq \frac{B}{(l+1)!}(s-t)^{l+1}$.

3. Algorithms

The heart of our monitor is a numerical algorithm (Sec. 3.1) for predicting the future states of a given black-box system from the states observed so far along the trajectory. These predicted future states will then be used to obtain the desired predictive runtime monitor (Sec. 3.2).

3.1. A Numerical Algorithm for Predicting Future States

Our (state) prediction algorithm has two phases: (1) In the *learning phase*, for each dimension $i \in [1; n]$ of the system's state space, we use a polynomial with time as its variable to approximate the dimension i of the trajectory ξ. The polynomial approximation function is obtained via a numerical procedure that uses a *finite set* of past states $\xi(t-k\tau), \ldots, \xi(t-2\tau), \xi(t-\tau), \xi(t)$ with an appropriately large k, which we will refer to as the *τ-stencil of length $(k+1)$ ending at t*. We write $x_{-k} = \xi(t-k\tau), \ldots, x_{-1} = \xi(t-\tau), x_0 = \xi(t)$, omitting t whenever it is clear from the context. (2) In the *prediction phase*, the learned polynomial approximation functions are used to compute the predictions of the future states up to the horizon h, denoted as $x_1 = \xi(t+\tau), \ldots, x_h = \xi(t+h\tau)$.

Learning phase. The learning phase independently considers each dimension $i \in [1; n]$ of the system's state space and computes a polynomial approximation for the i-th dimension of the trajectory function ξ. Hence, in what follows, without loss of generality, we assume that $n = 1$ and present our procedure for computing a polynomial approximation to the scalar-valued signal ξ.

We use the Taylor's polynomial P_l of ξ of a given degree l as the polynomial approximation. The challenge in obtaining P_l is that its coefficients depend on the values of the derivatives $\xi^{(1)}, \ldots, \xi^{(l)}$ at time point t, which are unknown to us owing to the black-box nature of the system.

Therefore, we numerically approximate the values of the derivatives using the backward difference (BD) method (Gear, 1967) from the observed stencil $x_{-k}, \ldots, x_0$ ending at time t. In particular, the BD approximation of the i-th derivative at x_0 is obtained as:

$$\nabla^i x_0 := \begin{cases} (x_0 - x_{-1})/\tau & \text{if } i = 1, \\ (\nabla^{i-1} x_0 - \nabla^{i-1} x_{-1})/\tau & \text{otherwise.} \end{cases}$$

The following closed-form expression can be obtained from the inductive definition above:

$$\nabla^i x_0 = \frac{\sum_{j=0}^{i} (-1)^j \binom{i}{j} x_{-j}}{\tau^i}. \tag{3}$$

It can be easily verified that $\nabla^l x_0$ depends on states up to x_{-l} in the past, and therefore the length of the stencil must be $k + 1 \geq l$. The approximation error is formally derived in the following lemma.

Lemma 2 *Let $i > 0$ and suppose that $\xi : \mathbb{R} \to \mathbb{R}$ is an $(i+1)$-times continuously differentiable function. Then, for every given stencil of length $k + 1 \geq i$, the following holds:*

$$|\xi^{(i)}(t) - \nabla^i x_0| \leq \tau \left| \frac{\xi^{(i+1)}(t)}{(i+1)!} \sum_{j=0}^{i} (-1)^j \binom{i}{j} (-j)^{i+1} \right| + \mathcal{O}(\tau^2). \tag{4}$$

The approximation $\nabla^i x_0$ is called the first order approximation, because for small $\tau < 1$, asymptotically, the first order term in τ dominates the error (i.e., the error is $\mathcal{O}(\tau)$). Higher order BD approximations would lead to smaller errors and will be considered in future works.

Proof [Proof of Lem. 2] In (3), if we use the following infinite Taylor's series expansion of x_{-j}

$$x_{-j} = \xi(t - j\tau) = \xi(t) - \frac{j\tau}{1!}\xi^{(1)}(t) + \frac{(j\tau)^2}{2!}\xi^{(2)}(t) - \frac{(j\tau)^3}{3!}\xi^{(3)}(t) + \ldots,$$

we observe that terms with derivatives of ξ of order lower than i cancel out, and we obtain:

$$\nabla^i x_0 = \xi^{(i)}(t) + \sum_{p=i+1}^{\infty} \frac{\tau^{p-i}\xi^{(p)}(t)}{p!} \sum_{j=0}^{i} (-1)^j \binom{i}{j} (-j)^p.$$

The claim is established by separating the dominating term with $p = i+1$ (which results in the first order term in τ) in the sum on the right hand side from the higher order terms. ■

Replacing each $\xi^{(i)}(t)$ in (2) with $\nabla^i x_0$ gives us the *approximated* Taylor's polynomial $\bar{P}_l(\cdot)$.

Remark 3 *The use of Taylor's polynomial is a design choice, and any other polynomial approximation could be used. Since the l-th degree polynomial that interpolates between $l+1$ points is unique, all approaches would provide the same answer. It is possible to choose the stencil length larger than $l+1$, in which case the polynomial is no longer unique, and a "best fit" polynomial can be obtained, e.g., the one that minimizes the mean-squared error. We leave this for future work.*

Prediction phase. The prediction phase of our monitor uses the approximated Taylor's polynomial $\bar{P}_l$ of the trajectory ξ around the current time t in order to compute the predicted future states, denoted as $\bar{x}_1 = \bar{P}_l(t+\tau), \ldots, \bar{x}_h = \bar{P}_l(t+h\tau)$. The following theorem establishes an error bound for the setting when bounds on the $(l+1)$ derivatives of ξ are known.

Theorem 4 *Let $l > 0$ and suppose that $\xi : \mathbb{R} \to \mathbb{R}$ is an $(l+1)$-times continuously differentiable function. Let $\bar{P}_l(s)$ be the approximated Taylor's polynomial obtained from a given stencil of length $k+1 \geq l$ ending at time t and the given sampling time τ. Let $h \in \mathbb{N}$ be a given horizon, and $m \in [1; h]$ be an arbitrary future sampling instance within the horizon. Suppose for every $p \in [1, l+1]$, B_p denotes the upper bound on the p-th derivative of ξ in the interval $(t, t+m\tau)$, i.e., $B_p = \sup_{r \in (t, t+m\tau)} \xi^{(p)}(r)$. Then,*

$$\begin{aligned} |\xi(t+m\tau) - \bar{P}_l(t+m\tau)| &\leq \frac{B_{l+1}}{(l+1)!}(m\tau)^{l+1} + \tau \cdot \sum_{p=1}^{l} \left| \frac{B_{p+1}}{(p+1)!} \sum_{j=0}^{p} (-1)^j \binom{p}{j} (-j)^{p+1} \right| + \mathcal{O}(\tau^2) \\ &= \mathcal{O}((m\tau)^{l+1} + \tau). \end{aligned}$$

Proof Follows by combining Thm. 1 and Lem. 2. ■

Thm. 4 suggests that, for small m and for $\tau < 1$, the prediction error is linear in τ, i.e., $\mathcal{O}(\tau)$. However, for long prediction horizons h that allow $m \in [1; h]$ to become too large, the term $(m\tau)^{l+1}$ may dominate over τ, and therefore the prediction error may increase to $\mathcal{O}((m\tau)^{l+1})$. Hence, considering larger prediction horizon h requires using smaller sampling time τ. This trend is visible in the ablation tests that we present in the experiments section. Finally, we remark that the $(l+1)$-times differentiability assumption is necessary only for our error bound analysis in Thm. 4. Even though the theoretical guarantees cannot be established in the black-box setting without the assumptions, we empirically show that our approach is still useful and sufficiently accurate in practice.

3.2. Taylor-Based Predictive Monitoring (TPM)

We now present TPM, our predictive runtime monitor for safety properties, based on the numerical state prediction algorithm in Sec. 3.1. The monitor continuously observes the samples drawn from the system's trajectory and stores the latest $l + 1$ samples in the FIFO queue Q. Using the states stored in Q as a stencil, TPM first computes the predicted future states using the approximate Taylor's polynomial. Afterwards, it outputs the safety levels of the predicted states or a desired statistic thereof, like the minimum safety level in h steps or the first time instance when safety is violated (i.e., φ drops below zero).

4. Experimental Evaluation

We implemented the algorithms from Sec. 3 in a prototype tool written in Python, and performed experiments to investigate the following two research questions: (i) How does TPM compare to state-of-the-art TTC method in giving early warnings of safety violations? (ii) How accurate is TPM when compared against the ground truth data? Both questions are studied on two environments.

Environment 1: F1Tenth Racing (O'Kelly et al., 2020). A racing car of $1:10$ scale needs to drive around a track while avoiding getting too close to the track boundaries. The state vector comprises of the X-Y coordinate, the rotation, the forward velocity, and the angular velocity. The control inputs are the steering angle and the throttle. A state is safe if its distance to the track boundaries is greater than a predefined threshold, set to $0.5\,\mathrm{m}$ meters in our experiments. We consider 70 differently parameterized controllers (Kresse, 2024), of which 54 are the so-called Pure Pursuit (PP) controllers which track a pre-planned path (Coulter, 1992) and 16 are Follow-The-Gap (FTG) controllers which steer towards the direction where there is the most free space (Sezer and Gokasan, 2012). The sampling time for this environment is fixed at $\tau = 0.01\,\mathrm{s}$.

Environment 2: F-16 Fighter Jet (Heidlauf et al., 2018). A simplified F-16 fighter jet system needs to fly at a safe height above the ground. The 16-dimensional state vector comprises of air speed (v_a), angle of attack (α), angle of sideslip (β), roll (ϕ), pitch, yaw, roll rate, pitch rate, yaw rate, northward displacement, eastward displacement, altitude (alt), engine power lag, upward accel, stability roll rate, and slide accel and yaw rate. The 4 control inputs are acceleration, stability roll rate, the sum of side acceleration and yaw rate, and throttle. A state is safe if the altitude is between $1000\ \mathrm{ft}$ and $45000\ \mathrm{ft}$. The sampling time for this environment is fixed at $\tau = 0.033\,\mathrm{s}$.

4.1. Experiment 1: Comparison to the Classical Time-to-Collision Method

We compared the performance of TPM to the baseline time-to-collision (TTC) (Vogel, 2003), a widely used measure in autonomous driving for predicting on-road safety violations (Wang et al., 2021). The TTC metric represents a *special case* of our approach with $l = 1$, utilizing only the current velocity to estimate the time to collision, assuming the vehicle continues in a straight line along its current orientation.

We design TPM monitors with two different types of outputs, namely boolean safety outputs and quantitative safety outputs, each of which can also be predicted by modifying the TTC algorithm. In the following, we describe the two types of outputs along with ways to measure their accuracy, for which we consider the actual system's trajectory as the ground truth.

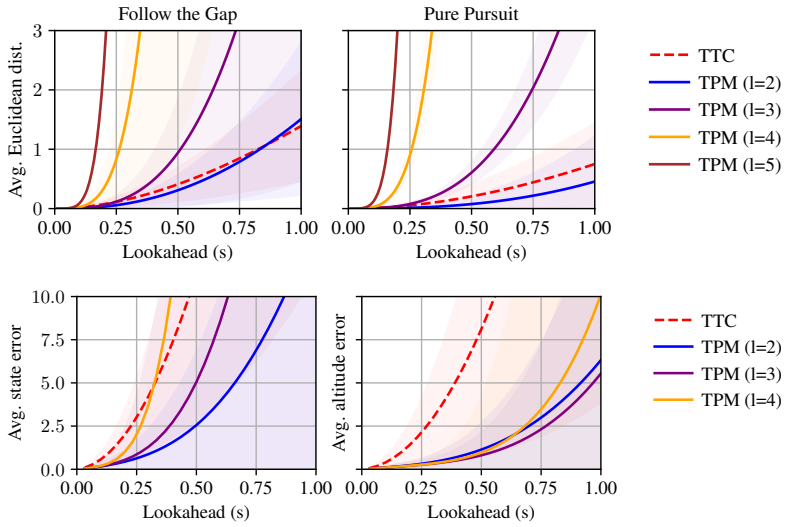

Figure 2: Ablation test results for prediction errors on F1Tenth (top row) and F-16 (bottom row). The lines represent the mean whereas the shaded regions represent the spread. For TPM, the constant l is the degree of the Taylor's polynomial. The lookahead (X-axes) are measured as $h\tau$ for varying h.

Outputs with boolean accuracy metrics. In this case, the monitor raises a warning if a safety violation is predicted within the prediction horizon. We measure the accuracy as follows: A warning is classified as a *true positive* (TP) if it is issued prior to an unsafe state; otherwise, it is a *false positive* (FP), indicating a false alarm. Conversely, if there was no prior warning but a unsafe state occurs, it is categorized as a *false negative* (FN), representing a missed detection; otherwise, it is called a *true negative* (TN). The true positive rate (aka, sensitivity) is defined as TPR = TP/(TP+FN), and the true negative rate (aka, specificity) is defined as TNR = TN/(TN+FP). TPR indicates how well the monitor can predict a real safety violation, whereas TNR indicates how well a monitor can predict the absence of it.

Outputs with quantitative accuracy metrics (Q). In this case, the monitors' outputs are environment-specific. In the F1Tenth environment, the output of the monitor is the same as before, i.e., it raises a warning if an unsafe state is predicted within the prediction horizon. We measure accuracy as the earliest time before entering an unsafe state when a warning is issued: $Q_{\text{F1TENTH}} = (\min_{i \geq t_{\text{UNSAFE}}-1} t^i_{\text{WARNING}}) - t_{\text{UNSAFE}}$, where t_{UNSAFE} is the time when the unsafe state takes place, t^i_{WARNING} is the time i when a warning is issued within the prediction horizon h. For the F-16 environment, the monitor is required to output the *minimum safety distance* within the horizon, defined as $d = \min_{i \in [0,h]} |(\overline{x_i} - alt_{min}) + (alt_{max} - \overline{x_i})|$. We measure accuracy as the difference between the minimum safety distance predicted by the monitor and the ground truth minimum safety distance observed: $Q_{\text{F-16}} = |d_{\text{PREDICTED}} - d_{\text{OBSERVED}}|$, for a fixed prediction horizon set to 50.

The accuracy of the TPM and TTC monitors were measured on random simulations with random initial states of the two systems that we consider: for F1Tenth, we collected $775,300$ simulation steps with FTG controllers and $2,647,570$ with PP controllers, and for F-16, we collected $33,750,000$ simulation steps.

The results are reported in Tab. 1. We observe that, overall, TPM significantly outperforms TTC in all categories except F1Tenth with FTG controller, in which case, owing to the less smooth trajectories, TPM showed lower TNR, i.e., it was more "cautious" and more often predicted safety would be violated when in reality it did not. In all other cases, TPM was more accurate and showed higher TPR, TNR, and Q-value than TTC.

Table 1: Performance comparisons between TPM and TTC on the F1Tenth and F-16 environments. The "%" in bracket in TP, FP, and FN are with respect to the total simulation step counts. The bold numerical entries indicate the which method among TPM and TTC was better.

Env.	h	Metric	**TPM**		TTC	
			FTG	PP	FTG	PP
F1Tenth	50	TP	**98323 (12.68%)**	**246143 (9.30%)**	80023 (10.32%)	235180 (8.88%)
		FP	36329 (4.69%)	**33049 (1.25%)**	**12371 (1.60%)**	100915 (3.81%)
		FN	**711 (0.09%)**	**269 (0.01%)**	19011 (2.45%)	11232 (0.42%)
		TPR	**0.993**	**0.999**	0.808	0.954
		TNR	0.946	**0.986**	**0.982**	0.958
		Q	**47.28 s**	**49.69 s**	39.22 s	46.61 s
	100	TP	**98736 (12.74%)**	**244257 (9.23%)**	90232 (11.64%)	214305 (8.09%)
		FP	177354 (22.88%)	**238744 (9.02%)**	**133671 (17.24%)**	566401 (21.39%)
		FN	**298 (0.04%)**	**2155 (0.08%)**	8802 (1.14%)	32107 (1.21%)
		TPR	**0.997**	**0.991**	0.911	0.870
		TNR	0.738	**0.901**	**0.802**	0.764
		Q	**98.09 s**	**96.67 s**	90.93 s	84.38 s
F-16	50	TP	**2706320 (8.01%)**		2518888 (7.46%)	
		FP	**20309 (0.06%)**		298477 (0.88%)	
		FN	**0 (0.00%)**		**0 (0.00%)**	
		TPR	**1.00**		**1.00**	
		TNR	**0.989**		0.86	
		Q	**72.01 ft**		108.94 ft	

These findings are further confirmed by our experiments shown in Fig. 2. Here, TPM significantly outperforms TTC in the F-16 environment as well as for the PP controllers in the F1Tenth environment. For FTG controllers, the TPM method performs better with a smaller lookahead, up to approximately 0.8 seconds (i.e., $h = 80$).

4.2. Experiment 2: Prediction Accuracy and Ablation Tests

To visually inspect the prediction accuracy of TPM, in Fig. 3, we plot the outputs of several instances of our monitor, with different horizon lengths, alongside the actual trajectory. We observe that the prediction error increases with longer prediction horizons. To further analyze the relationship between prediction error, Taylor polynomial's degree l, and prediction horizon h, we conduct ablation tests whose results are shown in Fig. 2. We observe that for F1Tenth, the configuration with $l = 2$ outperforms the rest, for both PP and FTG agents, with the improvement being particularly significant for PP agents. As the prediction horizon increases beyond 0.8s (80 prediction steps), the configuration with $l = 1$ becomes slightly better for the FTG agents. In the F-16 environment, we observe that $l = 3$ yields the best performance.

Overall, prediction accuracy is correlated with the smoothness of the dynamical system and the used controller. This is not surprising since from Thm. 4, we know that the prediction error increases as the values of the higher order derivatives increase, e.g., when the system has jerky movements.

Monitoring overhead. Our experiments were ran on a personal computer with 12th Gen Intel(R) Core(TM) i9-12900K processor and 32GB RAM. In our experiments, the monitoring overhead per observation consistently remains below 0.001 seconds for $l \leq 14$ and $h = 1$, or for $l \leq 5$ and $h \leq 10$. For longer prediction horizons, the overhead incrementally increases to approximately 0.002 seconds. These results demonstrate the efficiency of our monitor, confirming its lightweight nature and suitability for practical applicability.

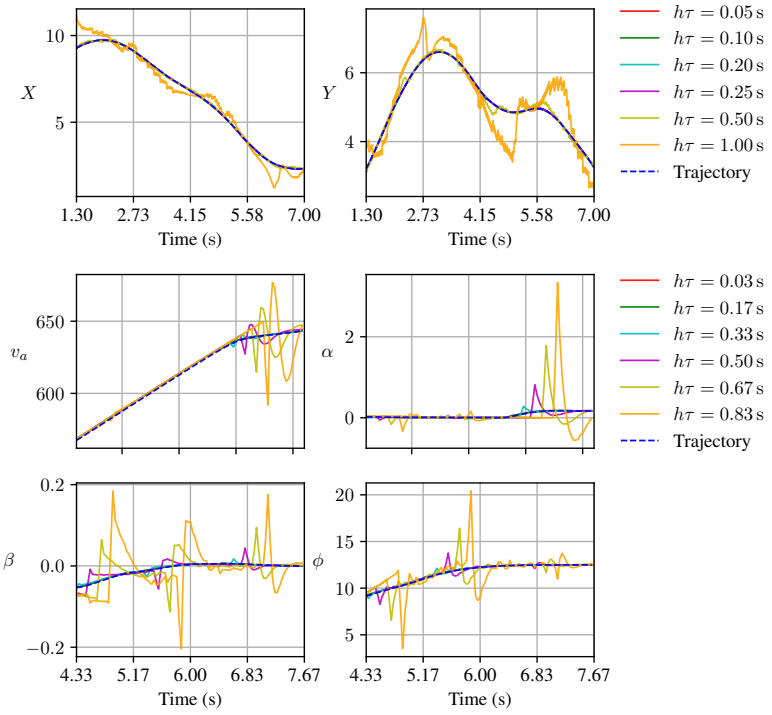

Figure 3: Visualization of the monitor's output as compared to the ground truth trajectories for the first two state dimensions of F1Tenth with $l = 2$ (top row) and four different state dimensions of F-16 with $l = 3$ (middle and bottom rows).

5. Conclusion

We introduced a lightweight *predictive* runtime monitoring framework for black-box controlled dynamical systems, which is able to predict safety violations ahead in time. At each time step, our monitor learns a Taylor-based polynomial approximation of the system's state trajectory from the past observations, which is then used to perform predictions of future states so that safety violations can be predicted. We derive formal upper bounds on the prediction error, given the knowledge of bounds on the derivatives of the trajectory. We present the effectiveness of our monitor on models of a racing car and a fighter aircraft taken from the literature. Future work will focus on studying numerical instabilities, higher order numerical approximations of the derivatives in Taylor's polynomial, different forms of polynomial approximations, as well as extensions to stochastic dynamical systems, multi-agent scenarios, and broader applications beyond safety verification.

Acknowledgments

This work was supported in part by the ERC project ERC-2020-AdG 101020093.

References

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 2669–2678. AAAI Press, 2018. doi: 10.1609/AAAI.V32I1.11797. URL https://doi.org/10.1609/aaai.v32i1.11797.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016. URL http://arxiv.org/abs/1606.06565.

Ezio Bartocci and Yliès Falcone. *Lectures on runtime verification*. Springer, 2018.

Jonathan Bunton and Paulo Tabuada. Confidently incorrect: nonlinear observers with online error bounds. In *2024 American Control Conference (ACC)*, pages 4729–4734. IEEE, 2024.

RC Coulter. Implementation of the pure pursuit path tracking algorithm. 1992.

Charles Dawson, Sicun Gao, and Chuchu Fan. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control. *IEEE Trans. Robotics*, 39(3):1749–1767, 2023. doi: 10.1109/TRO.2022.3232542. URL https://doi.org/10.1109/TRO.2022.3232542.

Claudio De Persis and Pietro Tesi. Formulas for data-driven control: Stabilization, optimality, and robustness. *IEEE Transactions on Automatic Control*, 65(3):909–924, 2019.

SSD Diop, JW Grizzle, PE Moraal, and A Stefanopoulou. Interpolation and numerical differentiation for observer design. In *Proceedings of the American control conference*, volume 2, pages 1329–1329. American Automatic Control Council, 1994.

Angelo Ferrando and Giorgio Delzanno. Incrementally predictive runtime verification. *Journal of Logic and Computation*, 33(4):796–817, 2023.

Charles William Gear. The numerical integration of ordinary differential equations. *Mathematics of Computation*, 21(98):146–156, 1967.

Peter Heidlauf, Alexander Collins, Michael Bolender, and Stanley Bak. Verification challenges in F-16 ground collision avoidance and other automated maneuvers. In *ARCH@ADHS*, volume 54 of *EPiC Series in Computing*, pages 208–217. EasyChair, 2018.

RJ Hyndman. *Forecasting: principles and practice*. OTexts, 2018.

Joel Janai, Fatma Güney, Aseem Behl, and Andreas Geiger. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Found. Trends Comput. Graph. Vis.*, 12(1-3):1–308, 2020. doi: 10.1561/0600000079. URL https://doi.org/10.1561/0600000079.

Fabian Georg Kresse. Deep off-policy evaluation with autonomous racing cars. Master's thesis, Technische Universität Wien, 2024.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1509.02971.

Matthew O'Kelly, Hongrui Zheng, Dhruv Karthik, and Rahul Mangharam. F1TENTH: An open-source evaluation environment for continuous control and reinforcement learning. In *NeurIPS 2019 Competition and Demonstration Track*, pages 77–89. PMLR, 2020.

Srinivas Pinisetty, Thierry Jéron, Stavros Tripakis, Yliès Falcone, Hervé Marchand, and Viorel Preoteasa. Predictive runtime verification of timed properties. *Journal of Systems and Software*, 132:353–365, 2017.

Stephen Prajna, Ali Jadbabaie, and George J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Trans. Autom. Control.*, 52(8):1415–1428, 2007. doi: 10.1109/TAC.2007.902736. URL https://doi.org/10.1109/TAC.2007.902736.

W. Rudin. *Principles of Mathematical Analysis*. International series in pure and applied mathematics. McGraw-Hill, 1964.

Volkan Sezer and Metin Gokasan. A novel obstacle avoidance algorithm:"Follow the Gap Method". *Robotics and Autonomous Systems*, 60(9):1123–1134, 2012.

Siyuan Shen, Tianjia Shao, Kun Zhou, Chenfanfu Jiang, Feng Luo, and Yin Yang. Hod-net: high-order differentiable deep neural networks and applications. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8249–8258, 2022.

Scott D. Stoller, Ezio Bartocci, Justin Seyster, Radu Grosu, Klaus Havelund, Scott A. Smolka, and Erez Zadok. Runtime verification with state estimation. In *RV*, volume 7186 of *Lecture Notes in Computer Science*, pages 193–207. Springer, 2011.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

Rania Taleb, Sylvain Hallé, and Raphaël Khoury. Uncertainty in runtime verification: A survey. *Comput. Sci. Rev.*, 50:100594, 2023.

Katja Vogel. A comparison of headway and time to collision as safety indicators. *Accident analysis & prevention*, 35(3):427–433, 2003.

Chen Wang, Yuanchang Xie, Helai Huang, and Pan Liu. A review of surrogate safety measures and their applications in connected and automated vehicles safety modeling. *Accident Analysis & Prevention*, 157:106157, 2021.

Emily Yu, Đorđe Žikelić, and Thomas A Henzinger. Neural control and certificate repair via runtime monitoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26409–26417, 2025.

Xian Zhang, Martin Leucker, and Wei Dong. Runtime verification with predictive semantics. In *NASA Formal Methods: 4th International Symposium, NFM 2012, Norfolk, VA, USA, April 3-5, 2012. Proceedings 4*, pages 418–432. Springer, 2012.