Predictive Runtime Monitoring of Vehicle Models Using Bayesian Estimation and Reachability Analysis

Yi Chou, Hansol Yoon and Sriram Sankaranarayanan
University of Colorado, Boulder, USA.

***Abstract*— We present a predictive runtime monitoring technique for estimating future vehicle positions and the probability of collisions with obstacles. Vehicle dynamics model how the position and velocity change over time as a function of external inputs. They are commonly described by discrete-time stochastic models. Whereas positions and velocities can be measured, the inputs (steering and throttle) are not directly measurable in these models. In our paper, we apply Bayesian inference techniques for real-time estimation, given prior distribution over the unknowns and noisy state measurements. Next, we pre-compute the set-valued reachability analysis to approximate future positions of a vehicle. The pre-computed reachability sets are combined with the posterior probabilities computed through Bayesian estimation to provided a predictive verification framework that can be used to detect impending collisions with obstacles. Our approach is evaluated using the coordinated-turn vehicle model for a UAV using on-board measurement data obtained from a flight test of a Talon UAV. We also compare the results with sampling-based approaches. We find that precomputed reachability analysis can provide accurate warnings up to 6 seconds in advance and the accuracy of the warnings improve as the time horizon is narrowed from 6 to 2 seconds. The approach also outperforms sampling in terms of on-board computation cost and accuracy measures.**

I. Introduction

As unmanned robotic vehicles are increasingly common in crowded urban spaces, it is increasingly important to provide a rigorous monitoring framework that can track these vehicles over time in order to detect and prevent collisions with buildings, pedestrians and other vehicles. However, such a monitor has to be able to predict the possible future positions of the vehicle over a future time horizon and use these predictions to detect possible collisions. To be precise, these predictions have to account for the dynamics of the vehicle which are often nonlinear. Also, to be effective, these predictions have to be fast: much faster than the time horizons over which the prediction is given.

In this paper, we explore a real-time safety monitoring framework for vehicles that relies on periodic position and velocity measurements. We assume a nonlinear discrete-time dynamical model that models positions and velocities over time as a function of control inputs (throttle and steering) and exogenous forces (wind drag and friction). In particular, our approach focuses on the planar coordinated turn (CT) model that has been widely used in target tracking [1], and collision detection for air vehicles [2]. Our approach uses a combination of Bayesian inference for tracking unmeasured aspects of the vehicle's dynamics that include the control inputs applied to the vehicle at each time. This is represented as a posterior distribution over the possible control inputs, and updated as new vehicle state measurements are obtained.

We combine the posterior distribution with a forward set-valued reachability analysis that computes rigorous set-valued bounds on the possible reachable states over a given time horizon (Cf. [3] for a survey). In particular, we use the Taylor model approach that represents the sets of states as low degree polynomial over uncertain noise symbols representing various unknown inputs to the model [4], [5]. However, reachability analysis approaches are too computationally expensive in terms of time, memory and power for potential use inside a real-time monitoring framework. Instead, our approach precomputes reachable sets exploiting the invariance of the vehicle models to translation and rotations of the coordinate frames. We demonstrate how such invariance properties reduce a potentially infinite number of possible instances to a finite number (a few thousands) of reachability computations that are performed offline and stored in a table for lookup during runtime monitoring.

The combination of Bayesian inference and precomputed reachability analysis results allows us to place upper bounds on the probability of collisions with an obstacle over a fixed time horizon. We evaluate our approach on experimental flight data from a test flight of a Talon UAV using the Pixhawk autopilot over the Pawnee national grasslands in eastern Colorado [6]. This data includes accurate GPS positions, the vehicle pose and heading at 0.4 second intervals. We use this data to conduct a "retrospective" real-time monitoring placing various fictitious obstacles at various distances from the ground truth trajectories. Our experiments demonstrate that real-time monitoring that combines reachability analysis with Bayesian inference is effective in terms of computation time. Also, the predictive accuracy is superior to sampling-based approaches. We also note that our approach provides warnings as early as 6 seconds in advance although these warnings have a high false alarm rate, but without any false negatives. At the same time, the false alarms resolve themselves into more accurate warnings for smaller time horizons of 4 and 2 seconds, respectively.

A. Related Work

The use of real-time monitors to predict and act against imminent property violations form the basis for runtime assurance using L1-Simplex architectures that switch between a lower performance but formally validated control when an impending failure is predicted [7]. However, the key issue lies in how impending failures are to be predicted. Often

predicting failures involves computing control invariant sets or solving reachability problems in real time [8], [9]. Our previous work uses a game-theoretic viability monitoring approach for predicting impending property violations for linear systems [10], [11]. The approach solves a viability problem that asks whether there exists any control input that can be used to avoid an impending collision. However, this approach is restricted to linear systems and essentially ignores the controller for the vehicle. Both of these limitations restrict the applicability of the viability monitor.

The use of tables to precompute information is common in approaches such as the ACAS-Xu collision avoidance system for UAVs [12]. However, the ACAS approach uses an encounter model which is a finite-state Markov decision process and stores the optimal policy in a table. The dynamics of the encounter model are much simpler than the approach considered here in this paper. Deep neural networks for collision avoidance have been proposed by Julian et al. by using neural networks to "compress" the ACAS-X tables [13]. Phan et al. demonstrate the use of neural network classifiers combined with offline statistical hypothesis testing to predict if the current state is likely to violate a property during a future time horizon [14]. However, neural networks are black boxes whose predictions must be currently trusted without the underlying evidence pending the development of efficient verification approaches.

Lygeros and Prandini (along with coworkers) have investigated the stochastic reachability analysis approaches for detecting and avoiding collisions between aircrafts [15], [16]. Similarities included the use of stochastic models to predict future positions with uncertainties and the use of reachability analysis to estimate the probability of collision. However, the use of Bayesian inference to process data during runtime and the precomputation for efficiency are unique to this paper.

Our approach is closely related that of Althoff and Dolan [17] which employs reachability analysis to verify that an autonomous ground vehicle can be driven in a collision free manner over some time horizon, in the presence of other vehicles and pedestrians (moving obstacles). However, there are many notable differences: we assume no knowledge of the vehicle state which is estimated in our approach from noisy observations. Althoff et al conservatively linearize the model in order to compute reachable sets, whereas we use nonlinear reachability analysis, employing invariance under reference frame transformations to perform offline precomputations. These invariance properties are generally treated as symmetries of the underlying dynamical systems. Recent approaches have studied more general frameworks for using symmetry reductions in reachability analysis [18], [19].

The use of the coordinated turn model to predict future aircraft positions and compute collision probabilities has been explored in the work of Hwang et al. [2], [20]. The similarities include the use of state estimation to track the unknown turn rate at each step. However, Hwang et al. also linearize the models in order to make the computation of the future states of the vehicle easier, whereas we work with nonlinear models. On the other hand, Hwang et al. use a multimodal switched system wherein the modes represent straight line flight versus turns in different directions.

Our work is also related to the recent work of Fisac et al. that track and predict the future states of human pedestrians in an environment consisting of cooperating autonomous UAVs [21]. They use Bayesian estimation to estimate a scalar "temperature parameter" that makes the model more uncertain of the future positions of the pedestrian. Since the dynamics of the pedestrian themselves are kept simple, this makes it easy to compute reachability estimates over a small time horizon in their approach. In our work, we use a nonlinear dynamical model with multiple state variables. Also, we do not adjust our confidence in our model predictions based on measurements. Instead, the measurements update the model's unknown inputs.

II. Preliminaries

We will now present discrete-time dynamical models for vehicles that model positions and velocities over time [22]. Let $\mathbf{x} \in \mathbb{R}^n$ denote the vector of state variables, and $\mathbf{u} \in \mathbb{R}^m$ denote control inputs. The state variables $\mathbf{x}(t)$ are updated at time t by a continuous function $f(\mathbf{x}(t), \mathbf{u}(t))$, i.e., $\mathbf{x}(t+1) := f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t))$, wherein $\mathbf{w}(t)$ is a stochastic disturbance input drawn from a known distribution $\mathcal{W}$.

These models are commonly used in robotics for performing path and motion planning as well as low level control input generation for driving the vehicle along a desired trajectory [23]. Depending on the application, the models may simply abstract away the vehicle itself as a particle, wherein the forces on the vehicle and its direction of travel are predicted by the model. At the other extreme, "hi-fidelity" models can capture aspects of the vehicle's construction and forces that act upon individual parts of the vehicle.

Example 2.1 (Dubins Model): The (planar) Dubins vehicle model treats a vehicle as a particle with the state variables (x, y, v, ψ), where (x, y) is the position of the center of mass of a vehicle, v is the speed (the magnitude of the velocity vector) of a vehicle, and ψ is the angle between the velocity vector and the x-axis. The control inputs include the throttle (u_v) and steering input (u_ψ). The motion of the vehicle obeys the following equations:

$$\begin{aligned} x(t+h) &= x(t) + hv\cos(\psi(t)) + \tfrac{h^2}{2}u_v\cos(\psi(t)) \\ y(t+h) &= y(t) + hv\sin(\psi(t)) + \tfrac{h^2}{2}u_v\sin(\psi(t)) \\ v(t+h) &= v(t) + hu_v \\ \psi(t+h) &= \psi(t) + hu_\psi \end{aligned}$$

Here, the control inputs (u_v, u_ψ) represent the rate of change of speed and the angular speed, which satisfies $u_v \in [u_{v,min}, u_{v,max}]$ and $u_\psi \in [u_{\psi,min}, u_{\psi,max}]$, respectively. Often, we may add exogenous disturbances to this model to capture drag forces and friction.

Example 2.2 (The coordinated-turn (CT) Model): The CT model describes a horizontal motion of aircraft flying at constant speeds along an arc. The model is described by $\mathbf{x}: (x, v_x, y, v_y)$, wherein the positions (x, y) and velocities

(v_x, v_y) can be measured accurately. The variable Ω refers to the unknown turning rate of the model, and is taken to be a time-varying input to this model. The equations are given by $\mathbf{x}(t+h) = A(\Omega, h)\mathbf{x}(t)$, wherein

$$A(\Omega, h): \begin{bmatrix} 1 & \frac{\sin(\Omega h)}{\Omega} & 0 & \frac{\cos(\Omega h)-1}{\Omega} \\ 0 & \cos(\Omega h) & 0 & -\sin(\Omega h) \\ 0 & \frac{1-\cos(\Omega h)}{\Omega} & 1 & \frac{\sin(\Omega h)}{\Omega} \\ 0 & \sin(\Omega h) & 0 & \cos(\Omega h) \end{bmatrix}$$

The control input Ω is bounded by the interval $[\Omega_{min}, \Omega_{max}]$. The model is assumed to have a fixed velocity over time. We relax this assumption by updating the velocity input with an exogenous disturbance input $w_v(t)$ at each time step.

Given a system $\mathcal{S}$, the trajectories are a sequence of states $\mathbf{x}(0), \mathbf{x}(h), \cdots$, wherein $\mathbf{x}(t+h) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t))$ for given control inputs $\mathbf{u}(t)$ and disturbances $\mathbf{w}(t)$.

III. Problem Statement and Overall Approach

Let $\mathbf{x}(t)$ denote the vehicle state which includes its position $(x(t), y(t))$, and velocity $(v_x(t), v_y(t))$. The vehicle's motion is affected by unknown control inputs $\mathbf{u}(t)$ (e.g., throttle and steering inputs) as well as exogenous disturbances. Finally, (noisy) measurements of the vehicle's position (x, y) and velocity (v_x, v_y) are available periodically at discrete time instances $t = 0, h, 2h, \cdots$. The measurement errors are assumed to be normal distributions with 0 mean and known standard deviations. The unknown control inputs $\mathbf{u}(t)$ belong to a bounded range $\mathcal{U}$.

Problem Setup: The inputs to the problem setup include:

1) Vehicle model with state variables $\mathbf{x}$, inputs $\mathbf{u}$ and exogenous inputs $\mathbf{w}$. We assume that the state variables $\mathbf{x}$ are known at each time step through noisy measurements with known error distributions, but the inputs $\mathbf{u}(t)$ are not measured. At the same time, we assume the exogenous disturbances have a known distribution.
2) Data from the observations (measurements) at some fixed time step $h > 0$.

Given this information, at some time instant t_0, we wish to know the distribution of possible states $\mathbf{x}(t_0 + Nh)$ for some time horizon $N > 0$. More specifically, we wish to provide the probability of satisfying safety properties of the form $\mathbb{P}(\mathbf{x}(t_0 + Nh) \in X_u)$ for some set of states X_u of interest. The set X_u typically models an obstacle.

Our work will specifically consider vehicle models such as the Dubins vehicle (Ex. 2.1) and the coordinated turn model (Ex. 2.2). The table below summarizes the state measurements obtained at each time step and the inputs that are not measured.

Model	Measurements	Unknowns
Dubins	(x, y, v, ψ)	u_v, u_ψ
CT Model	(x, y, v_x, v_y)	Ω

A. Overall Approach

Fig 1 provides a schematic diagram of the various components used in our approach. We perform a Bayesian inference starting from a prior distribution over the unknown inputs $\mathbf{u}(t)$ and conditioning the resulting state using the measurements to infer a posterior distribution over the unknown inputs. We will then use the posterior at time t to build a prior at time $t + h$.

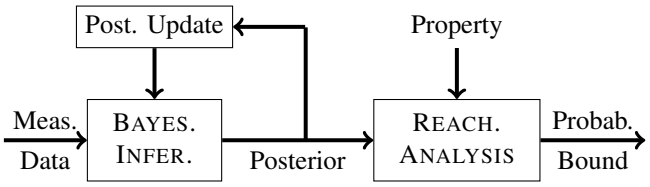


Fig. 1: Overall approach at a glance.

Next, we combine the Bayesian filtering approach with a set-valued reachability analysis that computes the possible reachable states at time $t + Nh$ using the initial states given by a combination of the measured state variables (with uncertainty) and the unknown inputs that are distributed according to the posterior. Reachability analysis propagates this information forward over the next Nh steps of the model in order to estimate the reachable set of states at time $t + Nh$. In our approach, we show how set-valued reachability approaches using Taylor polynomial expansions can be used to rigorously bound the possible future states [4], [5]. However, reachability analysis approaches are computationally expensive in terms of computation time and memory, as well as resources such as power. Therefore, our approach *precomputes* the reachable set estimates offline and uses a simple lookup of precomputed results on the fly. The translation and rotational invariance properties of the model allow us to limit the number of such precomputations. During runtime monitoring, the conditioning of the current prior distribution over unknown inputs using the measurement data is first performed, and then reachability analysis is performed using table lookups from the pre-computed results.

IV. Bayesian Inference

Bayesian inference is used to solve the problem of estimating the unknown input $\mathbf{u}(t)$ in a dynamical model given observations of the state $\mathbf{x}(t)$ at discrete time intervals of $h > 0$. The overall steps of the Bayesian inference at any time t are three-fold:

1) At any time instant t, our approach maintains a prior distribution $\mathbb{P}(\mathbf{u}(t-h))$ over the inputs $\mathbf{u}(t-h)$.
2) When a (noisy) measurement of $\mathbf{x}(t)$ is obtained, we update the prior distribution $\mathbb{P}(\mathbf{u}(t-h))$ for $\mathbf{u}(t-h)$ into a posterior distribution $\mathbb{P}(\mathbf{u}(t-h)|\mathbf{x}(t))$.
3) The prior $\mathbb{P}(\mathbf{u}(t))$ for the next step is computed using the posterior $\mathbb{P}(\mathbf{u}(t-h)|\mathbf{x}(t))$ from the previous step.

The posterior $\mathbb{P}(\mathbf{u}(t-h)|\mathbf{x}(t))$ is defined using Bayes rule:

$$\mathbb{P}(\mathbf{u}(t-h)|\mathbf{x}(t)) = \frac{\mathbb{P}(\mathbf{x}(t)|\mathbf{u}(t-h))\ \mathbb{P}(\mathbf{u}(t-h))}{\mathbb{P}(\mathbf{x}(t-h))}. \quad (1)$$

Exact inference attempts to compute a closed form for the posterior probability. This is possible under some restricted cases: the well-known Kalman filter uses a Gaussian prior,

Gaussian noise sampling distribution, and a linear model over $\mathbf{x}, \mathbf{u}$ [24], [25]. However, vehicle models we consider are often non-linear, and consequently, conjugacy relations do not hold in these non-linear models. Approximations such as Monte-Carlo methods are commonly used to approximate the posterior. They give rise to a variety of filtering approaches including *particle filters* [26], [27], [28]. These approaches provide samples $\mathbf{u}_1, \ldots, \mathbf{u}_m$ from the posterior. As $m \to \infty$, these samples converge in distribution to the actual posterior.

In our paper, we approximate the *set of support* U for the posterior by selecting finitely many *grid points* $\mathbf{u}_1, \ldots, \mathbf{u}_K$ for some number K. The posterior update in Eq. 1 is carried out exactly using Bayes rule on the discretized set of inputs. In particular, each posterior $\mathbb{P}(\mathbf{u}(t-h)|\mathbf{x}(t))$ will be a *categorial* distribution over the K chosen grid points. We first compute an "unnormalized" posterior *likelihood*:

$$L\left(\mathbf{u}_j|\mathbf{x}(t)\right) = \mathbb{P}\left(\mathbf{x}(t)|\mathbf{u}_j\right)\ \mathbb{P}\left(\mathbf{u}_j\right)\ , j = 1, \ldots, K\,.$$

The first term $\mathbb{P}(\mathbf{x}(t)|\mathbf{u}_j)$ is computed from the model: (a) apply one step of the model from $\mathbf{x}(t-h)$ using control inputs $\mathbf{u}_j$, (b) use the measurement $\mathbf{x}(t)$ along with the measurement noise to compute $\mathbb{P}\left(\mathbf{x}(t)|\mathbf{u}_j\right)$. Next, we normalize the likelihood to obtain the probability $\mathbb{P}(\mathbf{u}_j|\mathbf{x}(t))$.

Posterior to Prior Update: Finally, we use the posterior at time step $t-h$ computed above to obtain a prior for the control inputs at the next time instant t. To do so, we make a key enabling assumption that the unknown inputs change by a bounded amount at each step, and furthermore, we can assume that these bounds are known. This allows us to model the update to the posterior in many ways. We use the ϵ-transition approach that works as follows:

1) Fix a probability distribution $p_0(\mathbf{u})$. This is typically the uniform distribution over the chosen grid points.
2) Update the posterior using p_0 as follows:

$$\mathbb{P}(\mathbf{u}(t)) \ = \ (1-\epsilon)\mathbb{P}(\mathbf{u}(t-h)|\mathbf{x}(t)) + \epsilon p_0(\mathbf{u})\,.$$

Here, the parameter $\epsilon \in [0,1]$ and the distribution p_0 are chosen at the beginning of the process.

V. Reachability Analysis

In this section, we will present the problem of analyzing reachability. We will use ideas from *set-valued* reachability analysis to compute a set of states that are reachable at time $t_0 + Nh$ given the current state $\mathbf{x}(t_0)$ along with a posterior distribution of the current control input $\mathbf{u}(t_0)$. This will be computed over the *forward model* of the system that models the future states $\mathbf{x}(t_0 + kh)$ and controls $\mathbf{u}(t_0 + kh)$.

Forward Model: Formally, we express the model dynamics as $\mathbf{x}(t+h) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t))$, wherein $\mathbf{u}(t)$ is the input at time t and $\mathbf{w}(t)$ represents the disturbance (noise). The main differences between the "forward model" and that used in the Bayesian inference are as follows:

1) First, we will assume that $\mathbf{w}(t)$ is bounded by an interval $[\mathbf{w}_{\min}, \mathbf{w}_{\max}]$. This interval is obtained by truncating the distribution $\mathcal{W}$ used in the stochastic model in order to include a θ confidence interval for a fixed θ such as 0.99.
2) Second, we assume that the control inputs "evolve" over time in an uncertain manner:

$$\mathbf{u}(t+h) = \mathbf{u}(t) + \epsilon[\mathbf{u}_{\min}, \mathbf{u}_{\max}]\,,$$

In other words, we assume that at each step, the control inputs deviate compared to the control inputs at the previous time step. This is meant to capture the same effect as the ϵ transition used to update from the posterior at time t to a prior at time $t+h$.
3) Finally, the initial state of the model is given by $\mathbf{x}(0) \in \mathbf{x}_0 + [\mathbf{e}_{\min}, \mathbf{e}_{\max}]$ and $\mathbf{u}(0) = \mathbf{u}_0$, wherein $\mathbf{x}_0$ comes from the measurement data, $\mathbf{u}_0$ is a *sample* from the posterior and $[\mathbf{e}_{\min}, \mathbf{e}_{\max}]$ are interval bounds on the measurement error, accounting for at least a θ confidence interval.

Example 5.1 (Forward Coordinated Turn Model): The "forward model" uses the same dynamics for $\mathbf{x}(t)$ as provided in Ex. 2.2. Additionally, we incorporate the updates to model changes in the velocity and the turn rate.

$$\begin{aligned} v_x(t+h) &= v_x(t) + [-0.3, 0.3] \\ v_y(t+h) &= v_y(t) + [-0.3, 0.3] \\ \Omega(t+h) &= \Omega(t) + [-0.045, 0.045] \end{aligned}$$

A. Set-Valued Reachability Analysis

In this section, we will briefly survey work on reachability analysis techniques using ideas from interval analysis. A more in-depth survey of this area is available elsewhere [3]. Given a discrete-time dynamical model $\mathbf{x}(t+h) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t))$, wherein $\mathbf{u}(t)$ and $\mathbf{w}(t)$ are provided as intervals, and a set of initial states $\mathbf{x}(0) \in X_0$, the approach provides conservative bounds on the reachable states X_t over time $t \geq 0$ such that all the reachable states of the system lie inside X_t at time t. Reachability analysis techniques attempt to provide over-approximations of the reachable sets in contrast to sampling-based approaches that cannot provide guaranteed bounds but focus instead on finding counterexamples [29], [30], [31].

Many reachability analysis techniques have been proposed using interval analysis [32]. Therein, for each state variable $\mathbf{x}(t)$, we compute an interval bound. The bounds are updated by computing an interval $\mathbf{x}(t+h)$ given the interval for $\mathbf{x}(t)$. However, interval analysis approaches are well known to suffer from high overapproximation error due to the *wrapping effect* [3]. Instead, other representations of states ranging from ellipsoids [33], zonotopes [34] and "higher order" interval analysis approaches have been proposed to mitigate the wrapping effect. Higher order intervals express the state $\mathbf{x}(t)$ as an affine or polynomial function over *noise symbols* wherein each noise symbol stands for an interval [35], [4]. Since different components of $\mathbf{x}(t)$ share the same noise symbols, it is possible to avoid the wrapping effect using this approach [36].

Our approach uses the idea of Taylor polynomial arithmetic first proposed by Berz and Makino for computing reachable sets of nonlinear dynamical systems in complex particle physics and astronomical calculations [4], [36]. These ideas have been incorporated in tools such

as COSY-INFINITY, VNODE, and Flow* that can compute over-approximate reachable set estimates for ODE and switched/hybrid ODE models [5], [37], [38]. In this work, we specialize the Taylor polynomials implemented in the tool Flow* for discrete-time systems. Our approach can handle discrete systems wherein the function $f(\mathbf{x}, \mathbf{u}, \mathbf{w})$ can involve polynomials and trigonometric functions such as $\sin, \cos$ and $\tan$. Our reachability analysis uses Taylor polynomials of degree at most 3 to approximate these trigonometric functions and varies the polynomial degree at each step to tradeoff the computation cost against the precision of the bounds. These tradeoffs will be discussed in detail in an extended version of this paper. The resulting reachable set estimates are represented as a *zonotope* [34].

B. Computing Posterior Probabilities

In this section, we integrate the posterior probability for $\mathbb{P}(\mathbf{u}(t_0)|\mathbf{x}(t_0))$ obtained from our Bayesian inference with the reachability analysis. First, we note that the posterior distribution is represented as

$$(\mathbf{u}_0, p_0), \ldots, (\mathbf{u}_K, p_K)\,,$$

wherein $\mathbf{u}_0, \ldots, \mathbf{u}_K$ are the chosen grid points and $p_0, \ldots, p_K$ are the probabilities associated with each grid point such that $p_i \geq 0$ for $i = 0, \ldots, K$ and $\sum_{i=0}^{K} p_i = 1$. As a result, the "particles" used by our inference approach do not change at each step. What changes is simply the posterior probability p_i associated with $\mathbf{u}_i$.

Let $\mathbf{x}_0$ denote the measurement at time t_0 and $\mathbf{e} \in [\mathbf{e}_{\min}, \mathbf{e}_{\max}]$ bound the measurement error. As a result, our goal is to perform reachability analysis $K+1$ times over the forward model with each run using the initial state $\mathbf{x}(0) \in \mathbf{x}_0 + [\mathbf{e}_{\min}, \mathbf{e}_{\max}]$ and the initial control input $\mathbf{u}(0) = \mathbf{u}_i$ for $i = 0, \ldots, K$. Each reachability analysis pass provides us a set of states X_i for $i = 0, \ldots, K$ represented as an interval, zonotope, ellipsoid or a convex polyhedron, depending on the approach used, and bounds the possible states $\mathbf{x}(t_0 + Nh)$ for starting control value $\mathbf{u}(0) = \mathbf{u}_i$.

We can now use these reachable state estimates to compute the posterior probability that $\mathbf{x}(t_0 + Nh) \in X_u$ for some target set X_u of interest as follows:

1) Collect the set of indices $J \subseteq \{0, \ldots, K\}$ of reachable sets X_i that have a nonempty intersection with X_u:

$$J = \{j \in \{0, \ldots, K\} \mid X_j \cap X_u \neq \emptyset\}\,.$$

2) Add the posterior probabilities corresponding to J:

$$\mathbb{P}(\mathbf{x}(t_0 + Nh) \in X_u | \mathbf{x}(t_0), \ldots, \mathbf{x}(0)) \;:=\; \sum_{i \in J} p_i\,.$$

Example 5.2 (Probabilities of Collision with a Obstacle): Consider Ex. 2.1, we observe two successive state measurements $\mathbf{x}(0) = [0, 0, 2, 1.57]$ and $\mathbf{x}(0.5) = [-0.01, 0.81, 1.85, 1.58]$. The prior is assumed to be $u_v(0) \sim \mathcal{U}(-0.5, 0.5)$ and $u_\psi(0) \sim \mathcal{U}(-0.5, 0.5)$. Note that here the time step $h = 0.5$. We query the probability of collision with an obstacle O at $t = 4h$ as shown in Fig. 2. We approximate the set of possible control inputs (u_v, u_ψ) using 400 uniformly grid points over $[-0.5, 0.5] \times [-0.5, 0.5]$.

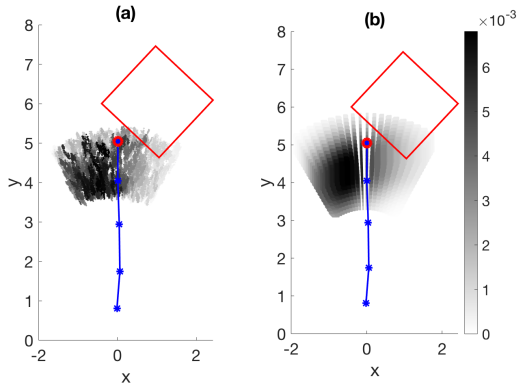


Fig. 2: Trajectory predictions for the Dubins model: (a) predicted reachable states using simulations (b) Reachability Analysis. The blue lines represents the measurements obtained at each step, the red circle is the position after 4 steps, and the rectangle represents the target set X_u (obstacle). The gray dots on the left are the samples from simulation whereas the polygons on the right are the reachable set estimates. Darker shades represent higher probabilities.

Fig 2(a) shows the probability distribution of position (x, y) at $t = 4h$ by running forward simulations from each grid point $\mathbf{u}_i$ to check of any of the simulated trajectories intersect the set X_u at time $t = 4h$. Next, we add up the probabilities of all the grid points that yield an intersecting trajectory with the target X_u. This yields an estimated probability of collision of 0.047. Fig 2(b) shows some of the reachable sets obtained by our Taylor model based reachability analyzer, corresponding to various choices of the grid points, shaded according to the posterior probability. Numerous grid points are simply too improbable in the posterior to be shown. These results lead to the upper bound on the probability of collision of 0.0862.

Precomputed Reachability Bounds: Whereas reachability analysis can be combined with Bayesian inference, existing reachability analysis methods are expensive and quite inefficient for real-time applications. We therefore propose a simple trick that works for models exhibiting invariance to translation and rotations of the coordinate frames. In particular, we precompute reachability analysis anticipating *all possible initial states* up front and store the results in a suitable data structure. During runtime monitoring, we simply lookup the results to compute the calculations.

Briefly, we assume during reachability analysis that (a) the vehicle's initial position $(x(0), y(0)) \in [-D_x, D_x] \times [-D_y, D_y]$ wherein D_x, D_y model the known errors in our position measurement method. For instance, for position measurements obtained through GPS we will assume $D_x = D_y = 0.1$ as a "reasonable" confidence interval for the error; (b) The coordinate frames are rotated so that the vehicle's current heading (yaw) is along the x axis of our coordinate frame. This allows us to simply choose $v_y \in [-V_y, V_y]$ for some uncertainty bound V_y taken to be 0.1 in our approach.

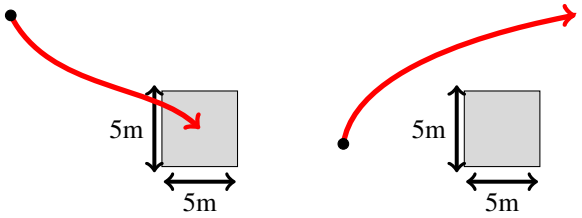


Fig. 3: Simulation setting: A UAV flies along with the red lines and an obstacles shown in gray. **(Left):** For collision detecting tests, a UAV approaches to the center of an obstacle in various directions. **(Right):** The obstacle is placed away from the known trajectory at various distances to test our monitor.

We will use a bound $v_x \in [0, 30]$ m/s in our approach and subdivide this interval into sub intervals of 0.1 m/s. Finally, the initial control inputs are already given by the possible grid points.

Putting all of this together, we precompute tens of thousands of instances of the reachability analysis algorithms each performed in a coordinate frame that allows us to preset values for many of the state variables. For instance, we precompute 5600 reachability analysis computations for the CT model (as described in Section VI) and 4000 reachability analysis calculations for the Dubins model described in Ex. 5.2. During runtime, the results are preloaded into a simple table and the reachable set estimates are obtained simply through a table lookup. After this, they are transformed into the current coordinate system for property checking.

VI. Experiments

In this section we evaluate the performance of our monitor on a dataset collected from a test flight of a Talon UAV flown over the Pawnee national grasslands in the USA in 2017 [6]. The dataset covers 800 seconds of flight time in all [1], and includes GPS positions and velocities of the UAV in the x, y, z directions recorded at $h = 0.4$ second intervals. It includes trajectories that are judged as "straight line" flights, as well as turns ranging from gentle to "tight" turns. After classifying segments of trajectories as "straight line" or "curved", we selected 200 segments at random. Each segment accounts for about 14 seconds of flying time, on the average. The sampled trajectories included 100 straight line trajectories and 100 turns from the data.

Collision Detection Tests: To test the effectiveness of our monitor, we setup collision detection tests wherein we defined obstacle regions and replayed the states from the flight data to our monitoring implementation. The obstacles were defined by 5×5 meter rectangular regions over the $x-y$ plane placed at varying distances from the trajectory segments. Obstacles placed at "0" distance represent a situation where the UAV will collide with the obstacle in the future, whereas obstacles placed at various distances represented known "misses" wherein the UAV will not collide with the obstacle in the future. These obstacle placements are illustrated in Fig 3. Note that our approach can be applied to scenarios that include dynamic obstacles although we consider only static obstacles in this paper. The position of an obstacle is used when checking whether the pre-computed reachability set is intersected with an obstacle, and does not affect the computation time.

We precomputed reachability analysis for time horizons of 2, 4 and 6 seconds, or alternatively, 5, 10 and 15 discrete steps in the model. We first measured the prediction accuracy for varying prediction time horizons and obstacle distances. The value of the Bayesian update parameter ϵ was fixed to 0.3. Based on obstacle placement, the ground truth of each scenario is classified as COLLISION or SAFE. Next, we count how many instances reported by our monitor match with ground truth. In particular, we use the probability threshold $\theta = 0.95$ for the monitor. The most desirable outcomes are when the ground truth matches with our predictions. A *false positive* occurs when our monitor reports a COLLISION whereas no collision actually happens. On the other hand, a *false negative* happens when our monitor misses an imminent collision by declaring it as SAFE. For each test, we used the previous 8 seconds of UAV trajectory data as the input to the Bayesian inference. Fig 4 plots an example scenario for various time horizons, showing the ground truth trajectory, the obstacle (marked by a green X) and the reachable sets computed by our approach shaded according to their posterior probabilities. A video of another example scenario is available at https://youtu.be/aXbqg_iHGNg.

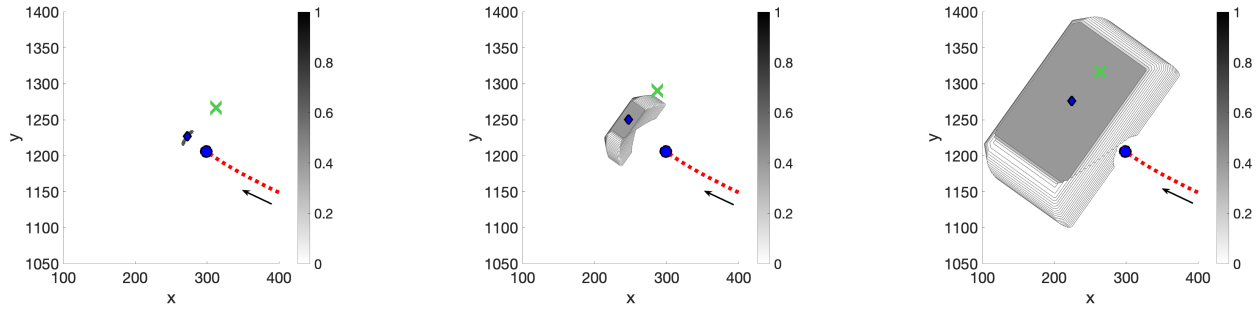
Comparison against Sampling: We compare our reachability approach against a sampling approach described previously for the Dubins model in Ex. 5.2, wherein we perform 100 simulations for each of the grid points to determine if any of the trajectories will intersect the obstacle. Using this, we add the posterior probability of the grid points that yield trajectories intersecting the obstacle. We use $\theta = 0.95$ as a threshold to declare an impending collision for this approach.

Performance Evaluation: Table I shows that the results in terms of the prediction accuracy, i.e, for what fraction of our tests did the monitor's prediction match the known ground truth outcome. For cases where the known ground truth is a COLLISION, $(1 - \text{accuracy})$ determines the false negative rate. On the other hand, for cases where the known ground truth is SAFE, $(1 - \text{accuracy})$ determines the false positive rate. The results are reported here for $\epsilon = 0.3$ since the results for $\epsilon = 0.1$ were similar.

We note that for a prediction horizon of 2 seconds (5 steps), our approach matches the ground truth data *in all the* 200 test cases. On the other hand, the sampling-based approach has a high false negative rate, wherein it misses real collisions. This is due to the "underapproximate" nature of sampling, which can cause the samples to miss obstacles and thus underestimate the true probability of collision.

The reachability analysis approach is able to flag all impending collisions, even for longer time horizons, without false negatives. This is due to the conservative nature of the

[1]Portions of the dataset corresponding to loitering maneuvers, takeoff and landing had to be discarded.



(a) Prediction time: 2 seconds (b) Prediction time: 4 seconds (c) Prediction time: 6 seconds

Fig. 4: Sample trajectory segment and reachable sets (gray) in time horizon 2, 4, and 6 seconds. Darker shades of reachable sets represent higher probabilities. A UAV is coming from the bottom right, and data points are shaded red when a UAV is updating the posterior of Ω. The current position of the UAV where prediction starts is shown as a blue circle, and future positions of the UAV in 2, 4, and 6 seconds that are unknown to the UAV are represented by blue diamonds. An obstacle is marked by a green X.

TABLE I: EXPERIMENTAL RESULTS

Time Horizon	ϵ	Trajectory Type	Ground Truth	Obstacle Distance	Accuracy	
					Reachability Analysis	Sampling
2 sec.	0.3	Straight	COLLISION	0	1.0	0.69
			SAFE	10	1.0	1.0
				20	1.0	1.0
				50	1.0	1.0
				70	1.0	1.0
				100	1.0	1.0
				200	1.0	1.0
		Curved	COLLISION	0	1.0	0.42
			SAFE	10	1.0	1.0
				20	1.0	1.0
				50	1.0	1.0
				70	1.0	1.0
				100	1.0	1.0
				200	1.0	1.0
4 sec.	0.3	Straight	COLLISION	0	1.0	0.1
			SAFE	10	0	1.0
				20	0.66	1.0
				50	1.0	1.0
				70	1.0	1.0
				100	1.0	1.0
				200	1.0	1.0
		Curved	COLLISION	0	1.0	0.11
			SAFE	10	0.01	1.0
				20	0.42	1.0
				50	1.0	1.0
				70	1.0	1.0
				100	1.0	1.0
				200	1.0	1.0
6 sec.	0.3	Straight	COLLISION	0	1.0	0.03
			SAFE	10	0	0.98
				20	0	1.0
				50	0	1.0
				70	0.5	1.0
				100	1.0	1.0
				200	1.0	1.0
		Curved	COLLISION	0	1.0	0.05
			SAFE	10	0	1.0
				20	0	1.0
				50	0.01	1.0
				70	0.5	1.0
				100	0.8	1.0
				200	1.0	1.0

TABLE II: COMPUTATION TIME IN SECONDS

Time Horizon	Highest Degree of Taylor Polynomials	Pre-computation (Reachability Analysis)		Real-time Prediction	
		# of Calculation	Avg. Time / Calculation	Reachability Analysis	Sampling
2 sec.	1	5600	1	0.1026	0.4066
	2	5600	1.1		
4 sec.	1	5600	2.3	0.1031	0.4312
	2	5600	5.4		
6 sec.	1	5600	3.5	0.1036	0.5251
	2	5600	13.7		

reachability analysis, that accounts for all possible trajectories of the model. However, the false positive rate increases as we increase the time horizon. For a time horizon of 4 seconds (10 steps), the approach is accurate for obstacles that are 50 meters or farther away from the ground truth. However, obstacles that are closer are flagged as potential collisions. This is because our approach uses conservative assumptions on how the unknown turn rate can vary in the future. At the 6 second time horizon, the approach can accurately forecast for obstacles that are 100 and 200 meters away but the approach flags all obstacles that are within 50 meters of the true flight path as potential collisions.

Computation time: We next analyze the computation times for the reachability analysis and the sampling method. Table II reports the pre-computation needed to generate all possible reachable sets for reachability analysis, and real-time computation to calculate the probability for both methods. Real-time computations were performed in Matlab(tm) running on a MacBook Pro laptop with 2.6 GHz Intel Core i7 and 16 GB RAM. As shown in Table II, real-time computation is performed fast enough for both reachability analysis and the sampling method although the latter requires more time for longer prediction time horizons. Again, pre-computation is carried out beforehand, and is parallelizable. For instance, our approach used a cluster with 25 cores to parallelize the precomputations that were performed within 4 hours in all.

VII. Conclusion

We conclude our analysis by noting that reachability analysis in combination with Bayesian inference is a promising approach for real-time monitoring. Our approach can reliably detect impending collisions 2 seconds ahead of time, while accurately predicting for obstacles that are more than 20 meters away from the ground truth within 4 seconds, and 100 meters within 6 seconds. We believe that more accurate reachability analysis techniques can be applied to our approach even though they can be more expensive, since our approach precomputes the results. This can potentially yield considerable improvements in accuracy. We also plan to apply our approach to richer classes of models and situations

involving multiple vehicles. Finally, we are also working on integrating our approach with planning to enable approaches for collision avoidance in addition to just monitoring.

ACKNOWLEDGMENTS

This work was funded in part by the US National Science Foundation (NSF) under award numbers 1815983, 1836900, the US Airforce Research Laboratory (AFRL) and the NSF-IUCRC Center for Unmanned Aerial Systems (C-UAS). All ideas and opinions expressed here are those of the authors and do not necessarily represent those of our sponsors.

REFERENCES

[1] M. Roth, G. Hendeby, and F. Gustafsson, "EKF/UKF maneuvering target tracking using coordinated turn models with polar/cartesian velocity," in *Intl. Conf. on Information Fusion (FUSION)*, July 2014, pp. 1–8.

[2] I. Hwang and C. E. Seah, "Intent-based probabilistic conflict detection for the next generation air transportation system," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 2040–2059, Dec 2008.

[3] J. Deshmukh and S. Sankaranarayanan, "Formal techniques for verification and testing of cyber-physical systems," in *Design Automation of Cyber-Physical Systems (Arquimedes Canedo and Mohammad Al Faruque)*. Springer-Verlag, 2019, pp. 69–105.

[4] M. Berz, *Modern Map Methods in Particle Beam Physics*, ser. Advances in Imaging and Electron Physics. Academic Press, 1999, vol. 108.

[5] X. Chen, E. Ábrahám, and S. Sankaranarayanan, "Flow*: An analyzer for non-linear hybrid systems," in *Proc. of CAV'13*, ser. LNCS, vol. 8044. Springer, 2013, pp. 258–263.

[6] S. Z. Watza, "Assessment of an online RF propagation hybrid architecture for communication-aware small unmanned aircraft systems," 2018, mS Thesis (University of Colorado, Boulder).

[7] L. Sha, "Using simplicity to control complexity," *IEEE Software*, vol. 18, no. 4, pp. 20–28, 2001.

[8] T. T. Johnson, S. Bak, M. Caccamo, and L. Sha, "Real-time reachability for verified simplex design," *ACM Trans. Embedd. Comput. Syst.*, vol. 15, no. 2, p. 29, May 2016.

[9] X. Chen and S. Sankaranarayanan, "Decomposed reachability analysis for nonlinear systems," in *2016 IEEE Real-Time Systems Symposium (RTSS)*. IEEE Press, Nov 2016, pp. 13–24.

[10] ——, "Model-predictive real-time monitoring of linear systems," in *IEEE Real-Time Systems Symposium (RTSS)*. IEEE Press, 2017, pp. 297–306.

[11] H. Yoon, Y. Chou, X. Chen, E. Frew, and S. Sankaranarayanan, "Predictive runtime monitoring for linear stochastic systems and applications to geofence enforcement for uavs," in *Runtime Verification*. Springer, 2019, pp. 349–367.

[12] G. Manfredi and Y. Jestin, "An introduction to acas xu and the challenges ahead," in *Digital Avionics Systems Conference (DASC)*, Sep. 2016, pp. 1–9.

[13] K. Julian, M. J. Kochenderfer, and M. P. Owen, "Deep neural network compression for aircraft collision avoidance systems," *AIAA Journal of Guidance, Control, and Dynamics*, 2018.

[14] D. Phan, N. Paoletti, T. Zhang, R. Grosu, S. A. Smolka, and S. D. Stoller, "Neural state classification for hybrid systems," in *Automated Technology for Verification and Analysis (ATVA)*, 2018, pp. 422–440.

[15] M. Prandini, J. Lygeros, A. Nilim, and S. Sastry, "Randomized algorithms for probabilistic aircraft conflict detection," pp. 2444 – 2449 vol.3, 02 1999.

[16] J. Lygeros and M. Prandini, "Aircraft and weather models for probabilistic collision avoidance in air traffic control," in *IEEE Conference on Decision and Control (CDC)*, vol. 3, Dec 2002, pp. 2427–2432 vol.3.

[17] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[18] J. Maidens and M. Arcak, "Exploiting symmetry for discrete-time reachability computations," *IEEE Control Systems Letters*, vol. 2, no. 2, pp. 213–217, 2018.

[19] H. Sibai, N. Mokhlesi, C. Fan, and S. Mitra, "Multi-agent safety verification using symmetry transformations," in *TACAS*, ser. Lecture Notes in Computer Science, vol. 12078. Springer, 2020, pp. 173–190.

[20] J. L. Yepes, I. Hwang, and M. Rotea, "New algorithms for aircraft intent inference and trajectory prediction," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 2, pp. 370–382, 2007.

[21] J. F. Fisac, A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, S. Wang, C. Tomlin, and A. D. Dragan, "Probabilistically safe robot planning with confidence-based human predictions," in *RSS'18*, 2018.

[22] R. Tedrake, "Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation," (Course Notes for MIT 6.832). Downloaded Feb. 2010 from http://underactuated.mit.edu/.

[23] S. M. LaValle, *Planning Algorithms*. USA: Cambridge University Press, 2006.

[24] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[25] R. J. Meinhold and N. D. Singpurwalla, "Understanding the kalman filter," *The American Statistician*, vol. 37, no. 2, pp. 123–127, 1983.

[26] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. A. Wan, "The unscented particle filter," in *Advances in neural information processing systems*, 2001, pp. 584–590.

[27] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2000.

[28] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 146, no. 1, pp. 2–7, 1999.

[29] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Hybrid systems: From verification to falsification," in *CAV*, ser. LNCS, vol. 4590. Springer, 2007, pp. 463–476.

[30] T. Nahhal and T. Dang, "Test coverage for continuous and hybrid systems," in *CAV*, ser. LNCS, vol. 4590. Springer, 2007, pp. 449–462.

[31] T. Nghiem, S. Sankaranarayanan, G. E. Fainekos, F. Ivancic, A. Gupta, and G. J. Pappas, "Monte-carlo techniques for falsification of temporal properties of non-linear hybrid systems," in *Hybrid Systems: Computation and Control (HSCC)*, 2010, pp. 211–220.

[32] R. Moore, R. B. Kearfott, and M. Cloud, *Introduction to Interval Analysis*. SIAM, 2009.

[33] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis." in *HSCC*, ser. LNCS, vol. 1790. Springer, 2000, pp. 202–214.

[34] A. Girard, "Reachability of uncertain linear systems using zonotopes," in *HSCC*, ser. LNCS, vol. 3414. Springer, 2005, pp. 291–305.

[35] L. H. de Figueiredo and J. Stolfi, "Self-validated numerical methods and applications," in *Brazilian Mathematics Colloquium monograph*. IMPA, Rio de Janeiro, Brazil, 1997, cf. http://www.ic.unicamp.br/~stolfi/EXPORT/papers/by-tag/fig-sto-97-iaaa.ps.gz.

[36] K. Makino and M. Berz, "Taylor models and other validated functional inclusion methods," *J. Pure and Applied Mathematics*, vol. 4, no. 4, pp. 379–456, 2003.

[37] ——, "COSY INFINITY version 9," *Nuclear Instruments and Methods A558*, pp. 346–350, 2005.

[38] N. S. Nedialkov, K. R. Jackson, and G. F. Corliss, "Validated solutions of initial value problems for ordinary differential equations," *Applied Mathematics and Computation*, vol. 105, no. 1, pp. 21–68, 1999.