Online Safety Verification of Trajectories for Unmanned Flight with Offline Computed Robust Invariant Sets

Daniel Althoff[1] Matthias Althoff[2] Sebastian Scherer[3]

***Abstract*— We address the problem of verifying motion plans for aerial robots in uncertain and partially-known environments. Thereby, the initial state of the robot is uncertain due to errors from the state estimation and the motion is uncertain due to wind disturbances and control errors caused by sensor noise. Since the environment is perceived at runtime, the verification of partial motion plans must be performed online (i.e. during operation) to ensure safety within the planning horizon and beyond. This is achieved by efficiently generating robust control invariant sets based on so-called loiter circles, where the position of the aerial robot follows a circular pattern. Verification of aerial robots is challenging due to the nonlinearity of their dynamics, the high dimensionality of their state space, and their potentially high velocities.**

We use novel techniques from reachability analysis to overcome those challenges. In order to ensure that the robot never finds itself in a situation for which no safe maneuver exists, we provide a technique that ensures safety of aerial robots beyond the planning horizon. Our method is applicable to all kinds of robotic systems that follow reference trajectories, such as bipedal robotic walking, robotic manipulators, automated vehicles, and the like. We evaluate our method by simulations of high speed helicopter flights.

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) provide huge benefits compared to manned vehicles. Examples are increased payload capabilities due to the lack of a cabin, unique flight envelopes since there are no g-force restrictions, no loss of life in dangerous missions, the possibility of long surveillance missions, and personal aerial deliveries by drones. An example is the Boeing Unmanned Little Bird (ULB, Fig. 1) which is equipped with a LIDAR and flies autonomously in unknown environments [1]. The aforementioned benefits can only be realized if safety challenges can be overcome. Different from unmanned ground vehicles, standstill is often not a safe option since fixed-wing aircraft or rotorcraft with a high payload have to move to stay airborne. Furthermore, transitioning to a standstill of a rotorcraft is impractical since so-called loiter patterns can handle shorter sensor ranges, which in turn allows one to operate the rotorcraft at higher speeds [1]. Another challenge is that aerial vehicles face stronger disturbances than ground vehicles since winds are stronger at higher altitudes. To enable UAVs to operate in the airspace and to develop trust it is necessary to guarantee safety of these systems. One critical component is to ensure that a safe alternative trajectory always exists that can be executed in case the intended trajectory is infeasible. We ensure this safety through emergency maneuvers that are safe for an infinite time horizon. These emergency maneuvers, however, are affected by sensor noise and disturbances (e.g. wind) and need to respect obstacles, consider the limited sensor capabilities, and still allow fast flight with the limited acceleration of a typical UAV.

Fig. 1: Automatic landing of the Unmanned Little Bird.

Previous work addresses the challenges to ensure safety by limiting the acceleration and increasing clearance to obstacles, leading to potentially unsafe behavior. Here we explicitly model the influence of disturbances on the vehicle and can therefore compute the set of possible behaviors during the intended trajectory and for a potential subsequent loiter circle. Only when all possible behaviors are collision-free, we execute the plan to guarantee safety. The set of possible behaviors of the closed-loop system is computed via reachability analysis, which returns the set of states that are reachable from a set of initial states subject to a set of uncertain inputs (disturbances and sensor noise). Since the aerial robot can move along a loiter circle forever, we are able to verify the maneuver for an infinite time horizon. Due to a constant replanning of the intended maneuver and the subsequent loiter circle, the loiter circle is only executed if no new and safe intended maneuver can be found. Once a new safe maneuver exists, the aerial robot exits the loiter circle and continues on its intended trajectory. Since loiter circles are typically longer trajectories than the intended trajectories, especially at high speeds, they take substantial time to verify. We verify those maneuvers offline and use them to online verify intended trajectories. To the best knowledge of the authors, this is the first work that formally guarantees safety for UAVs in the presence of disturbances and sensor noise for a nonlinear model with a challenging number of dimensions for formal verification.

II. RELATED WORK

The idea of computing invariant sets [2] (i.e. sets in which a system stays forever) to ensure safety or stability of

[1,3] are with the Robotics Institute, Carnegie Mellon Univ., Pittsburgh, USA, {althoff[1],basti[3]}@andrew.cmu.edu
[2] is with the Faculty of Computer Science, Technische Universität München, 85748 Garching, Germany, althoff@in.tum.de

finite-horizon plans is not new. In control theory, invariant sets are used to prove stability of model predictive control (also called receding horizon control), as shown in [3]. In [4] a *nonlinear model predictive control* framework is presented that guarantees nominal feasibility using invariant sets by providing necessary and sufficient conditions on the control horizon, the prediction horizon and the constraint set. Receding horizon control for UAVs based on mixed integer linear programming is presented in [5]. Loiter circles are added to the mixed integer linear programming formulation to ensure safety for an infinite time horizon. A library of precomputed emergency loiter circles for a full-scale autonomous helicopter is suggested in [1]. The emergency maneuvers are optimized regarding their path diversity, i.e. the difference of paths according to some measure.

In order to ensure safe movements of the aerial robot under bounded disturbances, one can apply robust model predictive control approaches. In [6], feasibility conditions for model predictive control subject to disturbances are presented. The idea of that work and later work (e.g. [7]) is to formulate input constraints that are tighter than the physical limits to reserve stronger control actions in the presence of disturbances. In [8] a robust model predictive controller using mixed-integer linear programming (MILP) is presented that provides a very general framework for problems involving both discrete decisions and continuous variables. An example for a discrete decision is to whether pass an obstacle to the left or the right. Tube-based model predictive control introduces tightened constraint sets for the input and state space of the robot system resulting in the so-called nominal system. The optimized trajectories based on the nominal system ensure feasibility for the actual (uncertain) system [9]–[11]. To guarantee safe movement for an infinite time horizon under bounded disturbances, an appropriate method is to steer the system into a robust positive control invariant set (RPCIS), e.g. [12]. A RPCIS ensures that there exists a controller such that the system never leaves the invariant set subject to arbitrary disturbances within a bounded set.

An alternative is to directly find regions in which a collision is inevitable as opposed to determining RPCIS. The goal of the controller is to avoid those *regions of inevitable collision* [13], which are defined as the union of *inevitable collision states* [14], within its finite planning horizon. Most of the work on regions of inevitable collision has focused on static environments, however, the concept has already been applied in dynamic environments for car-like vehicles [15], [16]. Additionally, in [17], [18] an extension for stochastic environments is presented taking into account the uncertain motion prediction of obstacles in the workspace, which calculates the probability that a state is an inevitable collision state. Other related work deals with the problem of the existence of an infinite long collision-free trajectory. This problem is discussed in [19] by the means of the ergodic forest scenario by finding criteria such as the size of obstacles, the distribution of obstacles and the maximum velocity of the robot under which almost surely an infinite time trajectory exists.

Another line of work provides formal methods to synthesize trajectories based on temporal logic specifications that are provably correct. In [20], temporal logic specifications are used to specify requirements on missions for UAVs. Trajectories for automated vehicles in static environments are synthesized in [21] within a discretized environment. Discrete environments are also used in [22], where a special focus is on dealing with an unknown workspace of the robot. A discrete environment is also used in [23] to synthesize plans for teams of robots. Motion primitives instead of grid-based movement is used in [24] for multi-robot systems planning from linear temporal logic specifications. Another work synthesizes robotic motion for a point mass (double integrator) by bounding the error to an abstract kinematic model and using the abstraction for the planning task [25]. In [26] so-called barrier certificates are used for validating nonlinear models with uncertain parameters. This method is applicable to polynomial vector fields with polynomial equalities and inequalities by the sum of squares (SOS) decomposition [27]. Recent advances of this approach are also used to perform stochastic verification [28]. SOS programming is also used in robust control to construct robust funnels which rely on time-varying Lyapunov functions [29].

None of the previously mentioned works can formally guarantee safe motion of complicated nonlinear dynamics for an infinite time horizon subject to bounded disturbances. In our approach we precompute a set of RPCIS using novel methods for reachability analysis that scales well with the dimension of the problem [30]. In addition, we provide guarantees for the intended plan using reachability analysis.

III. PROBLEM FORMULATION

The main challenges for verifying the safety of UAVs arise from their limited perception capabilities as well as disturbances and sensor noise acting on them. First we describe the model of the robot and the environment leading to the problem statement of this work.

A. Robot Model and Environment Description

We use a standard state space formulation to model our UAV over time t using the state $\mathbf{x}(t)$, the control input $\mathbf{u}(t)$, the bounded sensor noise $\boldsymbol{\nu}(t) \in \mathcal{N} \subset \mathbb{R}^r$, the system output $\mathbf{y}(t) \in \mathbb{R}^p$, and the bounded external input (disturbance) $\mathbf{w}(t) \in \mathcal{W} \subset \mathbb{R}^q$:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t)),\ \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{y}(t) &= g(\mathbf{x}(t), \boldsymbol{\nu}(t)) \end{aligned} \tag{1}$$

Dynamic and kinematic constraints of the robot are considered by the restricted state space $\mathbf{x}(t) \in \mathcal{X} \subset \mathbb{R}^n$ and the constrained input space $\mathbf{u}(t) \in \mathcal{U} \subset \mathbb{R}^m$. The motion model has a unique solution $\mathbf{x}(t)$ for all $\mathbf{x}(0), \mathbf{w}, \boldsymbol{\nu}$ and $\mathbf{u}$ if f: $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^q \to \mathbb{R}^n$ and g: $\mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^p$ are Lipschitz and $\mathbf{u}(t), \mathbf{w}(t)$ are piecewise continuous functions [2].

The workspace of the robot is denoted by $\mathbb{W}$ and the subset of the workspace occupied by the robot is referred to $\mathcal{A} \subset \mathbb{W}$ and as $\mathcal{A}(\mathbf{x}(t))$ at the state $\mathbf{x}(t)$. The occupancy of the ith obstacle in the workspace is denoted by $\mathcal{B}_i \subset \mathbb{W}$, the union

of all obstacles is denoted as $\mathcal{B} = \bigcup_i \mathcal{B}_i$, where all obstacles are considered as static in this work.

The limitations of the perception system is modeled by a field of view region $\mathrm{FOV} \subset \mathbb{W}$. Any obstacle which is inside the FOV, $\mathcal{B}_i \cap \mathrm{FOV}(\mathbf{x}(t)) \neq \emptyset$, at state $\mathbf{x}(t)$ is assumed to be detected by the perception system. Thus, the known workspace $\mathbb{W}^{\mathrm{k}}$ is the union of FOVs starting at time t_0 and is a subset of the workspace: $\mathbb{W}^{\mathrm{k}} = \bigcup_t \mathrm{FOV} \subset \mathbb{W}$.

Due to sensor noise, the initial state of the robot is uncertain and is represented by the bounded set $\mathcal{R}_0 \subset \mathcal{X}$. The solution $\chi(t; \mathbf{x}_0, \zeta(t))$ of the closed-loop system consisting of (1) and the controller $\mathbf{u}(t) = \Phi(\mathbf{y}(t), \zeta(t))$ is affected by the disturbance and sensor noise while tracking a certain state reference trajectory ζ. Thus, the reachable set for a reference trajectory $\zeta^*(t)$ during the time span $[t_0, t_f]$ and uncertain sets $\mathcal{R}_0$, $\mathcal{W}$, $\mathcal{N}$ describes the future set of states

$$\mathcal{R}^e([t_0,t_f];\mathcal{R}_0,\zeta,\mathcal{W},\mathcal{N}) := \big\{\chi(t;\mathbf{x}_0,\zeta(t),\boldsymbol{\nu}(t),\mathbf{w}(t)) \mid t\in[t_0,t_f], \mathbf{x}_0\in\mathcal{R}_0, \zeta(t)=\zeta^*(t), \boldsymbol{\nu}(t)\in\mathcal{N}, \mathbf{w}(t)\in\mathcal{W}\big\}.$$

We employ the semicolon to distinguish between the argument (time interval) of the reachable set and the parameters such as reference trajectory, initial set, bounded disturbance and sensor noise. In the following we will omit the explicit notation of the bounded disturbance and sensor noise for the sake of a more compact notation. Since, one cannot compute the set of reachable states $\mathcal{R}^e$ exactly [31], we use overapproximations $\mathcal{R} \supseteq \mathcal{R}^e$ as described in Sec. V.

B. Problem Statement

Since we assume partially-known environments, the reference trajectory constitutes only a partial motion plan, which is referred to partial motion planning [32]. We verify whether the behavior of an aerial vehicle can lead to a collision during the planning horizon and beyond. As mentioned earlier, we can guarantee collision avoidance if the aerial vehicle enters a Robust Positive Control Invariant Set (RPCIS) and this set does not contain any obstacles.

Definition 1 (Robust Positive Control Invariant Set (RPCIS) (based on [2])): *The set Ω is a robust positive control invariant set if $\forall \mathbf{x}(0) \in \Omega$, $\mathbf{w}(t) \in \mathcal{W}$, and $\boldsymbol{\nu}(t) \in \mathcal{N}$ there exists a reference trajectory $\zeta(t)$ and a continuous feedback control law $\mathbf{u}(t) = \Phi(\mathbf{y}(t), \zeta(t))$ for the system* (1)*, which assures $\mathbf{x}(t) \in \Omega$ for $t > 0$.*

It is noted, that the definition from [2] refers to the existence of a state feedback controller, while we later fix the controller, but refer to the existence of a reference trajectory. As it becomes clear from the definition, there exists a control law and a reference trajectory that the robot can follow for an infinite time horizon once the state is within a RPCIS. A given initial set $\mathcal{R}_0$ is feasible for the system (1) for an infinite time horizon in the presence of bounded disturbance $\mathcal{W}$ and sensor noise $\mathcal{N}$ if there exists a RPCIS Ω with $\mathcal{R}_0 \subset \Omega$ that lies within $\mathbb{W}^{\mathrm{k}}$, is free of obstacles and respects the state and input constraints. This is formalized as follows.

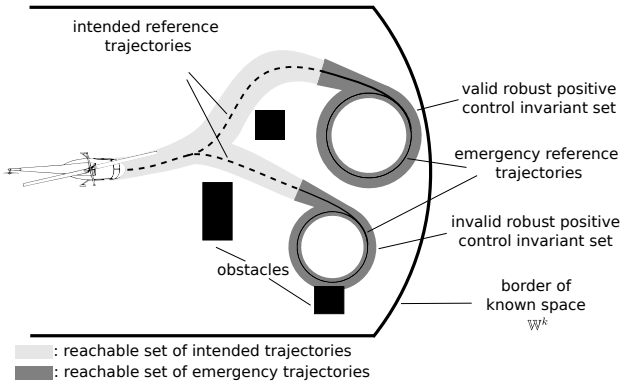


Fig. 2: Decoupling of trajectory verification into ensuring safety during the planning horizon (intended trajectory) and ensuring feasibility beyond the planning horizon. The reference trajectories of the intended plans are the dashed lines and the solid lines illustrate the reference trajectories for the emergency maneuvers. The associated reachable sets are depicted as shaded gray areas around the reference trajectories. Both emergency reference trajectories are not colliding, but the reachable set of the right emergency maneuver is overlapping with an obstacle and thus it is not safe.

Problem 1 (Verifying Safety for an Intended Trajectory in Uncertain Environments): *Given the initial set of states of the robot $\mathcal{R}_0$, the bounded disturbances $\mathcal{W}$, the bounded sensor noise $\mathcal{N}$, the intended reference trajectory ζ, and the known workspace $\mathbb{W}^{\mathrm{k}}$, the problem is to construct a* RPCIS *Ω (Def. 1) for the system* (1)*, such that $\forall t$ $\mathbf{u}(t) \in \mathcal{U}$, $\mathcal{R}_0 \subset \Omega$, $\forall \mathbf{x} \in \Omega : \mathbf{x} \in \mathcal{X} \wedge \mathcal{A}(\mathbf{x}) \cap \mathcal{B} = \emptyset \wedge \mathcal{A}(\mathbf{x}) \subset \mathbb{W}^{\mathrm{k}}$.*

IV. General Idea

The general idea to efficiently construct a RPCIS as described in Prob. 1 is to split its construction into two parts. First, we compute the reachable set of the intended trajectory online, followed by attaching the RPCIS of a precomputed loiter circle as depicted in Fig. 2. The RPCIS is constructed by computing reachable sets until the reachable set of the current time interval is fully enclosed by a subset of previously reached states:

Proposition 1. *The reachable set $\mathcal{R}([0,t];\mathcal{R}_0,\zeta_e) = \bigcup_{\tilde{t}\in[0,t]} \mathcal{R}(\tilde{t};\mathcal{R}_0,\zeta_e)$ along the reference trajectory ζ_e constitute a* RPCIS *if*

$$\exists t \in [t_0,\infty), \exists t_i \in (t_0,t) : \quad \mathcal{R}(t;\mathcal{R}_0,\zeta_e) \subseteq \mathcal{R}([t_0,t_i];\mathcal{R}_0,\zeta_e).$$

Proof. According to the semi-group property of reachable sets [33, p. 39] and with $\mathcal{R}_0^{[0,t]} := \mathcal{R}([0,t];\mathcal{R}_0,\zeta_e)$ one can iteratively compute reachable sets as

$$\begin{aligned}\mathcal{R}([0,t+\Delta t];\mathcal{R}_0,\zeta_e) &= \mathcal{R}_0^{[0,t]} \cup \mathcal{R}([t,t+\Delta t];\mathcal{R}_0^{[0,t]},\zeta_e) \\ &= \mathcal{R}_0^{[0,t]} \cup \big(\mathcal{R}([t,t+\Delta t];\mathcal{R}_0^{[0,t]},\zeta_e) \setminus \mathcal{R}_0^{[0,t]}\big).\end{aligned} \tag{2}$$

Since RPCISs are defined for an infinite time we have to repeat (2) forever, unless a fixed point is reached, i.e. $\mathcal{R}([0,t+\Delta t];\mathcal{R}_0,\zeta_e) = \mathcal{R}([0,t];\mathcal{R}_0,\zeta_e)$. From (2) one can directly see that a fixed point is reached for $\mathcal{R}([t,t+\Delta t];\mathcal{R}_0^{[0,t]},\zeta_e) \setminus \mathcal{R}_0^{[0,t]} = \emptyset$. Since

$$\mathcal{R}(t;\mathcal{R}_0,\zeta_e) \subseteq \mathcal{R}([t_0,t_i];\mathcal{R}_0,\zeta_e) \subseteq \mathcal{R}([0,t];\mathcal{R}_0,\zeta_e)$$

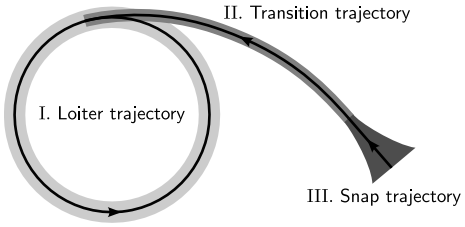


Fig. 3: The three components of the loiter maneuver. The reference trajectories are the solid black lines and the associated reachable sets are depicted as shaded gray areas around the reference trajectories.

it directly follows that $\mathcal{R}([t, t+\Delta t]; \mathcal{R}_0^{[0,t]}, \zeta_e) \setminus \mathcal{R}_0^{[0,t]} = \emptyset$ so that we reached a fixed point. □

With the result from Prop. 1 we can reformulate Prob. 1 into the problem of verifying the intended trajectory and finding an emergency reference trajectory ζ_e:

Problem 2 (Robust Emergency Trajectory): *Given is an initial set $\mathcal{R}_0$, the motion model* (1)*, the constrained sets $\mathcal{X}$, $\mathcal{U}$, the known workspace $\mathbb{W}^{\mathrm{k}}$, the bounded sets $\mathcal{W}$, $\mathcal{N}$ and the intended reference trajectory ζ_n for the time horizon $[t_0, t_e]$. The robust emergency trajectory problem is to find an emergency reference trajectory ζ_e for the time horizon $[t_e, \infty)$ with the initial set $\mathcal{R}_0^{t_e}$ such that*

1) $\mathcal{R}(t_e; \mathcal{R}_0, \zeta_n) \subseteq \mathcal{R}_0^{t_e}$
   (initial set of ζ_e is superset)
2) $\exists t \in [t_e, \infty), \exists t_i \in (t_e, t)$:
   $\mathcal{R}(t; \mathcal{R}_0^{t_e}, \zeta_e) \subseteq \mathcal{R}([t_e, t_i]; \mathcal{R}_0^{t_e}, \zeta_e)$
   (reachable set of ζ_e is a RPCIS*, Prop. 1)*
3) $\forall \mathbf{x} \in \mathcal{R}([t_e, t_i]; \mathcal{R}_0^{t_e}, \zeta_e)$:
   $\mathbf{x} \in \mathcal{X} \wedge \mathcal{A}(\mathbf{x}) \subset \mathbb{W}^{\mathrm{k}} \wedge \mathcal{A}(\mathbf{x}) \cap \mathcal{B} = \emptyset$
   (reachable set of ζ_e is feasible and collision-free)

The online verification of the intended trajectory and the concatenation of a valid robust emergency trajectory result in a RPCIS according to Prob. 1 for both trajectories. The emergency trajectory is split into a snap trajectory, a transition trajectory, and a loiter trajectory (see Fig. 3). The loiter trajectory ensures a cyclic (infinite time) emergency maneuver and the transition trajectory describes the transition from the intended trajectory to the loiter trajectory. In order to ensure that we can attract a larger set of final states of the intended trajectory onto a single emergency maneuver, we construct a snap trajectory. In the next section, we discuss the problem to verify reference trajectories during the planning horizon leading to RPCISs.

V. Verification of Reference Trajectories

In this section we briefly describe how reference trajectories are verified. As previously described and shown in Fig. 2, our reference trajectories consist of an intended trajectory, and an emergency trajectory. We compute the reachable set of the intended trajectory online since it is also computed online by the trajectory planner. The reachable set of the emergency trajectories are computed offline since they are more time consuming due to the length of the maneuver and the sufficiency to choose from a finite set of precomputed emergency maneuvers.

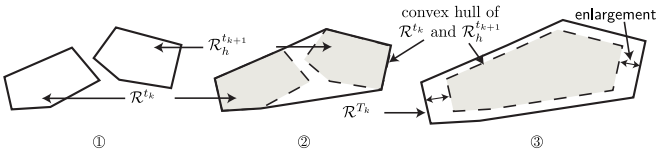


Fig. 4: Steps for the computation of an overapproximation of the reachable set for a linear system.

A. Reachability Analysis

We adopt the method from [30] for online and offline computation of reachable sets for consecutive time intervals $T_k = [t_k, t_{k+1}]$. To efficiently compute the reachable set of the nonlinear system (1), we continuously linearize the dynamics for each time interval T_k as described in [30]. After introducing the Minkowski addition $\mathcal{C} \oplus \mathcal{D} := \{c + d | c \in \mathcal{C}, d \in \mathcal{D}\}$, we obtain a linear differential inclusion, which represents all possible solutions for the time interval T_k of the original system (1) by $\dot{\tilde{\mathbf{x}}} \in \tilde{A}\tilde{\mathbf{x}}(t) \oplus \tilde{\mathcal{U}}(t)$, where $\tilde{\mathbf{x}}(t) := \mathbf{x}(t) - \mathbf{x}^*$ and $\mathbf{x}^*$ is the linearization point for $t \in T_k$. $\tilde{\mathcal{U}}(t)$ is a set of uncertain inputs that contains the sensor noise, disturbance, and the linearization errors. For simplicity, we restrict ourselves to the time interval T_k and later repeat the presented procedure for all times. For the reachabilty analysis, we split the effect of $\hat{\mathcal{U}}(t)$ into its center $\hat{\mathbf{u}}_c$ and the translated set $\hat{\mathcal{U}}_\Delta = \hat{\mathcal{U}}(t) \oplus (-\hat{\mathbf{u}}_c)$. The following algorithm (see Fig. 4) takes advantage of the superposition principle for linear dynamics:

1) Starting from $\mathcal{R}^{t_k}$, compute the set of all solutions $\mathcal{R}_h^{t_{k+1}}$ for the affine dynamics $\dot{\mathbf{x}} = A\mathbf{x}(t) + \hat{\mathbf{u}}_c$ at time t_{k+1}.
2) Obtain the convex hull of $\mathcal{R}^{t_k}$ and $\mathcal{R}_h^{t_{k+1}}$. This encloses all solutions for the current time interval assuming that trajectories from $\mathcal{R}^{t_k}$ to $\mathcal{R}_h^{t_{k+1}}$ are straight lines and that the input is certain ($\hat{\mathcal{U}}_\Delta = 0$).
3) Compute $\mathcal{R}^{T_k}$ by enlarging the convex hull of 2) to account for the error made by the assumption that trajectories are straight lines and account for the set of uncertain inputs $\hat{\mathcal{U}}_\Delta \neq 0$.

Details of this algorithm can be found in [30]. As stated in that work, we use zonotopes as an efficient representation of reachable sets since they are closed under linear maps and Minkowski addition, the two main operations of the presented reachability analysis technique. They are also a very compact representations of sets in many dimensions as opposed to e.g. convex sets spanned by vertices.

B. Connectivity of Reachable Sets

In order to guarantee that the reachable set for the complete trajectory (intended plus emergency trajectory) is overapproximative, we have to ensure that the final reachable set of the intended trajectory $\mathcal{R}(t_e; \mathcal{R}_0, \zeta_n)$ is a subset of the initial set of the emergency trajectory $\mathcal{R}_0^{t_e}$:

$$\mathcal{R}(t_e; \mathcal{R}_0, \zeta_n) \subseteq \mathcal{R}_0^{t_e}. \tag{3}$$

The precomputed (offline) reachable sets of the emergency trajectory are invariant with respect to position and orientation so that we can arbitrarily translate and rotate the

emergency maneuver such that the connectivity condition (3) is fulfilled. Additionally, it must be checked that the offline RPCIS is obstacle-free by intersecting the workspace occupancy of the RPCIS with the workspace obstacles. It can be guaranteed that the combination of intended and robust emergency trajectory is a RPCIS if (3) is fulfilled, since the robust emergency trajectory is already constructed as proposed in Prop. 1 and thus is a RPCIS. Thus, it is possible to generate a library of offline generated RPCIS in order to online verify intended trajectories.

VI. IMPLEMENTATION

In this section we present a possible implementation of our novel approach for generating RPCISs for aerial robots. We use a fixed-wing model, which is applicable for helicopter flights at high speed, and we generate circular-shaped loiter trajectories with constant speed. It is noted that the presented algorithm is independent of the used model, as long as the model can be described in the form of (1).

A. Fixed-Wing Model for Constant Wind

The state of the robotic system is defined as

$$\mathbf{x} = [x, y, z, \psi, v_{xy}, v_z, \theta]^\intercal,$$

where x, y, z describe the position in $\mathbb{W}$, v_{xy} is the absolute velocity of the UAV in the xy-plane, v_z is the velocity in the z-direction, ψ is the heading and θ is the roll angle. The differential equations of the motion model are

$$\begin{aligned} &\dot{x} = v_{xy}\cos(\psi) + w_x,\ \dot{y} = v_{xy}\sin(\psi) + w_y \\ &\dot{z} = v_z,\ \dot{\psi} = \frac{g}{v_{xy}}\tan(\theta),\ \dot{v}_{xy} = a_{xy},\ \dot{v}_z = a_z, \end{aligned} \tag{4}$$

where w_x, w_y are the constant wind speeds in x and y-direction and g is the standard gravity. The control input is

$$\mathbf{u} = [a_{xy}, a_z, \dot{\theta}]^\intercal,$$

where a_{xy} is the acceleration in the xy-plane, a_z is the acceleration in the z-direction and $\dot{\theta}$ is the roll rate. The lateral, longitudinal and altitude error term ϵ_x, ϵ_y and ϵ_z, respectively, are given as

$$\begin{aligned} \epsilon_x &= \ \ \cos(\psi_d)(x_d - x) + \sin(\psi_d)(y_d - y) \\ \epsilon_y &= -\sin(\psi_d)(x_d - x) + \cos(\psi_d)(y_d - y) \\ \epsilon_z &= z_d - z, \end{aligned}$$

where x_d, y_d, z_d are the desired positions. The desired roll angle θ_d is given as

$$\theta_d = k_1\epsilon_y + k_2(\psi_d - \psi) + k_3(\dot{\psi}_d - \dot{\psi}),$$

where $\psi_d, \dot{\psi}_d$ are the desired heading and heading rate, respectively. The control inputs are calculated based on the desired velocities $v_{xy,d}, v_{z,d}$:

$$\begin{aligned} a_{xy} &= k_5\epsilon_x + k_6(v_{xy,d} - v_{xy}), \\ a_z &= k_7\epsilon_z + k_8(v_{z,d} - v_z),\ \dot{\theta} = k_4(\theta_d - \theta). \end{aligned}$$

The parameters are chosen as $k_1 = 0.05$, $k_2 = 5.0$, $k_3 = 5.0$, $k_4 = 1.0$, $k_5 = 0.1$, $k_6 = 1.0$, $k_7 = 0.13$, and $k_8 = 1.0$ to balance control performance, sensor noise reduction, and actuator limitations. It is noted that a simple feedback controller, which is not optimized for windy conditions, is used since the focus of this work lies on the RPCIS generation.

B. Emergency Maneuver

We generate an emergency trajectory as described in Sec. IV that solves Prob. 2 for a set of uncertain wind conditions $\mathcal{W}$, sensor noise $\mathcal{N}$, and initial set $\mathcal{R}_0$. The wind uncertainty is represented by two closed intervals $\mathcal{W} = \left[[w_x^{\min}, w_x^{\max}], [w_y^{\min}, w_y^{\max}]\right]^\intercal$. The wind is defined relative to the initial heading of the robot, thus the emergency maneuver is orientation invariant. Without loss of generality, the initial heading of the emergency maneuver is set to zero for this implementation. In the following paragraphs we describe the generation of the reference trajectories to construct the RPCIS.

Algorithm 1 Iterative Robust Emergency Maneuver

Input: $\mathcal{R}_0, \mathcal{W}, \mathcal{N}, \dot{\theta}, a_{xy}, b, \mathrm{N}, \Delta t, k$

1: **Reference Trajectories**
2: $[\zeta_t, t_t] \leftarrow \mathrm{transitionTraj}(\mathcal{R}_0, \mathrm{a_{xy}}, \dot{\theta})$
3: $[\zeta_l, t_l] \leftarrow \mathrm{loiterTraj}(\zeta_\mathrm{t})$
4: **Determine Reachable Sets**
5: $\mathcal{R}_t \leftarrow \mathcal{R}(t_t; \mathcal{R}_0, \zeta_t)$
6: $\mathcal{R}_l \leftarrow \mathcal{R}(t_l; \mathcal{R}_t, \zeta_l)$
7: **Check for Steady Set**
8: $t \leftarrow t_l$
9: **while do**
10: $\mathcal{R}_{l'} \leftarrow \mathcal{R}_l$
11: $\mathcal{R}_l \leftarrow \mathcal{R}(t_l; \mathcal{R}_{l'}, \zeta_l)$
12: **if** $\mathrm{dist}(\mathcal{R}_l, \mathcal{R}_{l'}) \leq \tau_d$ **then**
13: **break**
14: $t \leftarrow t + t_l$
15: **Inclusion Check**
16: $\hat{\mathcal{R}}_b \leftarrow \mathrm{Ih}(\mathcal{R}_{l'}, b)$
17: $\boldsymbol{\mathcal{R}}_{\text{init}} \leftarrow \hat{\mathcal{R}}_b$
18: $\hat{\mathcal{R}}_0 \leftarrow \hat{\mathcal{R}}_b$
19: **for** $i \leftarrow 1, \mathrm{N}$ **do**
20: $\hat{\mathcal{R}}_i = \mathrm{Ih}\left(\mathcal{R}([(i-1)\Delta t, i\Delta t]; \hat{\mathcal{R}}_{i-1}, \zeta_l)\right)$
21: $\boldsymbol{\mathcal{R}}_{\text{init}} \leftarrow \{\boldsymbol{\mathcal{R}}_{\text{init}}, \hat{\mathcal{R}}_i\}$ (Concatenation)
22: $\mathcal{R}_c \leftarrow \mathcal{R}(t_l; \hat{\mathcal{R}}_\mathrm{N}, \zeta_l)$
23: **for** $i \leftarrow 1, \mathrm{N}$ **do**
24: $\hat{\mathcal{R}}_c \leftarrow \mathrm{Ih}(\mathcal{R}_c)$
25: **if** $\hat{\mathcal{R}}_c \setminus \boldsymbol{\mathcal{R}}_{\text{init}} == \emptyset$ **then**
26: **return** $\mathcal{R}([0, t]; \mathcal{R}_0, \{\zeta_t, \zeta_l\}) \cup \boldsymbol{\mathcal{R}}_{\text{init}} \cup \mathcal{R}([\mathrm{N}\Delta t, t_l + i\Delta t]; \hat{\mathcal{R}}_\mathrm{N}, \zeta_l)$
27: **else**
28: $\mathcal{R}_c \leftarrow \mathcal{R}(i\Delta t; \mathcal{R}_c, \zeta_l)$
29: $b \leftarrow k\,b$ **go to** 15

1) Transition and Loiter Trajectory: The transition and the loiter reference trajectory are generated together as described in Alg. 1. The transition trajectory starts at the center of $\mathcal{R}_0$ and ends in a loiter trajectory (lines 2 to 3) as

shown in Fig. 3. The time horizon of the transition trajectory and the loiter trajectory are t_t and t_l, respectively. The transition trajectory is generated by a constant deceleration in the $xy-$plane a_{xy} and a constant change of the roll angle $\dot{\theta}$ (line 2). As shown in [1], this results in compact loiter patterns and increases the likelihood that they fit inside $\mathbb{W}^{\mathrm{k}}$. Furthermore, the acceleration in $z-$direction is determined, thus that the robot reaches level flight at the end of the transition trajectory ($v_z = 0$). The following loiter trajectory is defined as a circular trajectory starting at the end state of the transition trajectory with constant roll angle and speed of the robot (line 3). Next, the reachable sets at the end of the transition and loiter reference trajectory are calculated (lines 5 to 6). The set inclusion, formalized in Prop. 1, can only be achieved after the reachable sets sufficiently converged to a limit cycle [34]. We detect this in lines 8 to 14 after computing the final reachable set of each full turn $\mathcal{R}_{l'}$ along the loiter circle ζ_l by checking whether the Euclidean distance $\mathrm{dist}()$ of centers of reachable sets from consecutive turns is below the threshold τ_d (line 12). Once the reachable sets have sufficiently converged to the limit cycle, we return to the reachable set at the beginning of the previous turn $\mathcal{R}_{l'}$ and over-approximate that reachable set by an interval hull $\hat{\mathcal{R}} = \mathrm{Ih}(\mathcal{R}) \supseteq \mathcal{R}$ for computational reasons. The ith coordinate of the interval hull $\hat{\mathcal{R}}$ is enlarged by b_i, which are stored in the vector $b \in \mathbb{R}^n$ resulting in the enlarged set $\hat{\mathcal{R}}_b$ (line 16). We perform this valid interval hull approximation (overapproximation) because set inclusion of zonotopes is infeasible in high dimensional space, while set inclusion of interval hulls is computationally cheap. Since it is unlikely that the reachable set after one turn ends in exactly that interval hull, we over-approximate the successor reachable set of a time interval Δt by another interval hull, which is repeated N times (lines 18 to 21). This results in the set of the $\mathrm{N}+1$ interval hulls $\boldsymbol{\mathcal{R}}_{\text{init}} = \{\hat{\mathcal{R}}_b, \hat{\mathcal{R}}_1, \cdots, \hat{\mathcal{R}}_{\mathrm{N}}\}$ with $\hat{\mathcal{R}}_i = \mathrm{Ih}\left(\mathcal{R}([(i-1)\Delta t, i\Delta t]; \hat{\mathcal{R}}_{i-1}, \zeta_l)\right)$. Next, the reachable set is continued from the last interval hull $\hat{\mathcal{R}}_{\mathrm{N}}$ of $\boldsymbol{\mathcal{R}}_{\text{init}}$ until the turn is completed resulting in the set $\mathcal{R}_c$ in line 22. This set is over-approximated by its interval hull $\hat{\mathcal{R}}_c$ in line 24. This is valid since if the overapproximation is included, the original set is included, too. The inclusion check of Prop. 1 (line 25) is performed by the compliment between $\hat{\mathcal{R}}_c$ and $\boldsymbol{\mathcal{R}}_{\text{init}}$. If the inclusion check fails, the set $\mathcal{R}_c$ is propagated one time step further (line 28) and the inclusion check is repeated with its overapproximating interval hull $\hat{\mathcal{R}}_c$, which is repeated for up to N times. If all inclusion checks fail, the size of the initial set $\hat{\mathcal{R}}_b$ is increased by increasing b (line 29) by a constant factor $k > 1$. This step is repeated until the inclusion check passes or until the constraints of the system (1) are violated, which aborts the algorithm. In that event, the reference trajectories must be adopted by reducing the roll rate $\dot{\theta}$ or the deceleration a_{xy} of the transition trajectory resulting in a larger loiter radius of the reference trajectory and Alg. 1 is initialized again. If the inclusion check in line 25 is true, we return the RPCIS according to Def. 1.

This algorithm is used to offline determine a valid RPCIS and thus it does not affect the online capabilities of the overall verification of intended trajectories.

2) Snap Trajectory: As described in [35] we use a parameterized reference trajectory with linear change in curvature and speed. Additionally, we extended the reference trajectory by the z-dimension. The state reference trajectory needs to provide information about the position $x(t), y(t), z(t)$ the heading $\psi(t)$ and the heading rate (roll angle) $\dot{\psi}(t)$ which are obtained by integrating the following equations

$$\dot{x} = \cos(\psi) v_{xy}(t), \dot{y} = \sin(\psi) v_{xy}(t), \dot{z} = v_z, \dot{\psi} = v(t)\kappa(t).$$

The resulting reference trajectory depends on the speed $v_{xy}(t), v_z(t)$ and curvature $\kappa(t)$, which are defined as

$$v_{xy}(t) = v_{xy}(0) + a_{xy}t,\ v_z(t) = v_z(0) + a_z t,\ v_z(0) = 0$$
$$\kappa(t) = \kappa(0) + \dot{\kappa}t,$$

where a_{xy}, a_z and $\dot{\kappa}$ are the parameters of the reference trajectory. The parameters are added to the reachability analysis by defining new state variables to the system with zero dynamics. The initial values $v_{xy}(0), v_z(0), \kappa(0)$ ensure that the final state of the snap trajectory is the initial state of the transition trajectory. In order to perform the reachability analysis, the dynamics of the system are described relative to the parameterized reference trajectory. The 13 dimensional state space consists of the longitudinal error, lateral error, error in altitude, heading error, error in v_{xy}, error in v_z, error in roll angle, and 6 error terms describing the values of the parameterized reference trajectory and their first derivative. The reader is referred to [35] for a more detailed description. Therefore, we verify the snap trajectory for a continuous (infinite) set of initial curvatures and speeds. In contrast to [35] we use the parameterized reference trajectory to compute the backward reachable set [36].

C. Validity of the Emergency Maneuver

According to Prob. 2 we still need to ensure that the resulting reachable sets of the reference trajectories are collision-free and do not violate the constraints of the robot in order to be a valid RPCIS. The state constraints are checked by projecting the reachable set of each consecutive time interval to the corresponding dimension and comparing it to the valid range of the corresponding state. The reachable sets are overapproximated by interval hulls for the x, y and $z-$direction which allows one to perform an efficient collision checking with the obstacles in the workspace. In order to account for the robot shape $\mathcal{A}$, the obstacles are enlarged, which is often referred to as the free space (configuration space) of the robot. The collision checking is performed by sequentially checking the intersection between the interval hulls and the enlarged obstacles. The same method is used to ensure that all reachable sets are inside the known workspace $\mathbb{W}^{\mathrm{k}}$ by treating the compliment $\mathbb{W} \setminus \mathbb{W}^{\mathrm{k}}$ as an obstacle.

VII. SIMULATION

We use simulation scenarios to evaluate the novel approach from Sec. VI. The rotorcraft is described by the fixed-wind

model from Sec. VI-A and has the following relevant constraints for generating the emergency maneuver: maximum roll angle $|\theta| \leq 0.52\,(30^\circ)$, maximum roll rate $|\dot{\theta}| \leq 0.26\,\mathrm{rad/s}$ $(15\,^\circ/\mathrm{s})$, maximum accelerations $|a_{xy}| \leq 1\,\mathrm{m/s^2}$, $|a_z| \leq 1\,\mathrm{m/s^2}$. These are the same constraints as used for the ULB platform [1]. In order to construct robust emergency maneuvers, we need to initialize Alg. 1. Therefore, the disturbance is set to $\mathcal{W} = \big[[-0.5, 0.5]\mathrm{m/s}, [-0.5, 0.5]\mathrm{m/s}\big]^\intercal$ and the initial set is set to

$$\begin{aligned}\mathcal{R}_0 = \big[&[-3.45, 3.45]\mathrm{m}, [-2.45, 2.45]\mathrm{m},\\ &[-7.49\mathrm{e}{-4}, 7.49\mathrm{e}{-4}]\mathrm{m}, [-2.89\mathrm{e}{-2}, 2.89\mathrm{e}{-2}]\mathrm{rad},\\ &[49.66, 50.31]\mathrm{m/s}, [-8.75\mathrm{e}{-5}, 8.75\mathrm{e}{-5}]\mathrm{m/s},\\ &[-0.11, 0.11]\mathrm{rad}\big]^\intercal.\end{aligned}$$

The initial set results from the reachability analysis of the robot following a straight line oriented in the $x-$direction with $50\,\mathrm{m/s}$ ending at the origin (intended trajectory). The bounded sensor noise $\mathcal{N}$ for this scenario is set to

$$\begin{aligned}\mathcal{N} = \big[&[-0.1, 0.1]\mathrm{m}, [-0.1, 0.1]\mathrm{m}, [-0.1, 0.1]\mathrm{m},\\ &[-0.01, 0.01]\mathrm{rad}, [-0.1, 0.1]\mathrm{m/s}, [-0.1, 0.1]\mathrm{m/s},\\ &[-0.01, 0.01]\mathrm{rad}\big]^\intercal,\end{aligned}$$

and the noise gets added to the motion model (4) of the robot. We generate the transition and loiter trajectory as described in Sec. VI-B.1 with the maximum allowed roll angle. The transition trajectory is generated with a roll rate of $0.22\,\mathrm{rad/s}$ and a deceleration of $a_{xy} = -0.8\,\mathrm{m/s^2}$ starting at position $[0; 0; 0]$ with heading $\psi = 0$ until the heading reaches $\psi = \pi$ while the roll angle is saturated at its maximum allowed value. The speed of the robot at the end of the transition trajectory is $28.4\,\mathrm{m/s}$ resulting in a loiter circle with radius of $174.5\,\mathrm{m}$. Alg. 1 is not able to verify the resulting emergency trajectory, since the maximum roll constraints are violated. Thus, the maximum roll angle of the transition trajectory is sequentially decreased until it reaches $0.436\,\mathrm{rad}$, resulting in a valid emergency trajectory. The corresponding loiter circle has a radius of $176.1\,\mathrm{m}$. For the inclusion check from Sec. VI-B.1, the reachable set is converted to an interval hull and increased by $b = [1.3, 1.3, 1.3, 1.05, 1.1, 1.1, 1.1]$. The resulting emergency maneuver together with example plots from the inclusion check are shown in Fig. 5. The reachable sets are computed for a time interval of $0.01\,\mathrm{s}$ which is equivalent to a $100\,\mathrm{Hz}$ tracking frequency. The snap trajectory is constructed by simulating the system for $0.9\,\mathrm{s}$ until the roll constraint of the system is violated. Therefore, the intervals $\dot{\kappa}(t) \in [-2.5\mathrm{e}{-4}, 2.5\mathrm{e}{-4}]\,1/\mathrm{ms}$, $a_{xy} \in [-0.25, 0.25]\mathrm{m/s^2}$, $a_z \in [-0.5, +0.5]\mathrm{m/s^2}$ are used for the parameterized reference trajectory. This emergency maneuver can be used to verify any intended trajectory ending with a curvature of $\kappa \in [-2.25\mathrm{e}{-4}, 2.25\mathrm{e}{-4}]\,1/\mathrm{m}$, speed of $v_{xy} \in [49.4, 50.5]\mathrm{m/s}$, $v_z \in [-1.05, 1.05]\mathrm{m/s}$ as long as the final reachable set of the intended trajectory is a subset of the initial set of the emergency maneuver.

In the following we briefly discuss the presented approach for the application to the ULB equipped with a LIDAR as described in [1]. The LIDAR has a range of $800\,\mathrm{m}$ and the maximum speed is $60\,\mathrm{m/s}$. Due to the limited sensor range and high speeds, the intended trajectories are typically verified for a horizon of 2–4 s. The verification of the intended trajectories can be performed online, since the current implementation (single threaded) takes approx. 80% of the time of the trajectory. However, since the loiter maneuvers at high speeds can take up to $70\,\mathrm{s}$, we use our precomputed emergency maneuvers. The calculation of an emergency maneuver takes about 3 to 5 times longer than the actual time span of the emergency maneuver, since for the inclusion check the reachable sets need to be calculated for multiple loiter turns as described in Alg. 1.

VIII. Conclusions and Future Work

In this paper we discuss the problem of safety for a robotic system in uncertain and partially-known environments. To the best of the authors' knowledge, this is the first time that safety for aerial robots in the presence of bounded disturbances and bounded sensor noise is verified for an infinite time horizon. Therefore, we propose a new algorithm to generate robust control invariant sets based on recent advances in reachability analysis. The example implementation for aerial vehicles uses loiter circles to ensure that the robot will stay inside the partially-known workspace. Each generated emergency maneuver can guarantee safety for a set of intended trajectories. Furthermore, simulation scenarios are presented, verifying the novel concept and the implementation.

In future work we want to use this approach to offline generate an extended library of emergency maneuvers allowing to online verify trajectories at high speeds. We also plan to verify different closed-loop models to validate our new approach.

Acknowledgment

This work is supported by ONR under contract N00014-12-C-0671 and by the German Research Foundation under grant AL 1185/3-1.

References

[1] S. Arora, S. Choudhury, D. Althoff, and S. Scherer, "A principled approach to enable safe and high performance maneuvers for autonomous rotorcraft," in *AHS 70th Annual Forum, Montreal, Quebec, Canada, May 20-22*, 2014.

[2] F. Blachini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

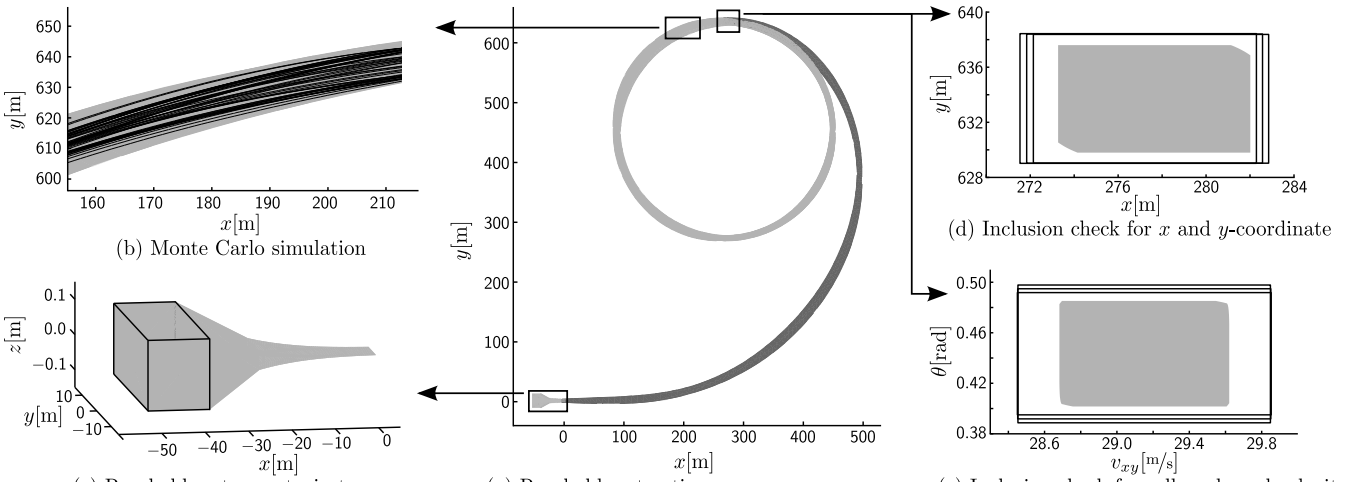
[3] H. Michalska and D. Mayne, "Robust receding horizon control of constrained systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 11, pp. 1623–1633, 1993.

[4] E. C. Kerrigan and J. M. Maciejowski, "Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model preditctive control," in *Proc. of the IEEE Conf. on Decision and Control*, 2000, pp. 4951–4956.

[5] T. Schouwenaars, "Safe trajectory planning of autonomous vehicles," Ph.D. dissertation, Massachusetts Institute of Technology, 2006.

[6] J. R. Gossner, B. Kouvaritakis, and J. A. Rossiter, "Stable generalized predictive control with constraints and bounded disturbances," *Automatica*, vol. 33, no. 4, pp. 551–568, 1997.

[7] A. G. Richards and J. P. How, "Robust stable model predictive control with constraint tightening," in *Proc. of the IEEE American Control Conference*, 2006, pp. 1557–1562.



(b) Monte Carlo simulation (d) Inclusion check for x and y-coordinate

(c) Reachable set snap trajectory (a) Reachable set entire emergency maneuver (e) Inclusion check for roll angle and velocity

Fig. 5: a) Resulting concave hull of the emergency maneuver partitioned in snap (light gray), transition (dark gray) and loiter trajectory (light gray). b) 50 Monte Carlo simulations are performed to demonstrate the performance of the reachability analysis c) The black box shows the final interval hull of the snap trajectory and the gray area illustrates the concave hull of the corresponding reachable sets d) and e) show the valid inclusion check of the robust loiter trajectory for the x, y and v_{xy}, θ dimension. The black polygons represent the array of interval hulls $\boldsymbol{\mathcal{R}}_{\text{init}}$ and the solid gray polygons represent $\hat{\mathcal{R}}_c$.

[8] B. Luders, "Robust trajectory planning for unmanned aerial vehicles in uncertain environments," Ph.D. dissertation, Massachusetts Institute of Technology, 2008.

[9] J. M. Bravo, T. Alamo, and E. F. Camacho, "Robust mpc of constrained discrete-time nonlinear systems based on approximated reachable sets," *Automatica*, vol. 42, no. 10, pp. 1745–1751, 2006.

[10] D. Limon, I. Alvarado, T. Alamo, and E. Camacho, "Robust tube-based MPC for tracking of constrained linear systems with additive disturbances," *Journal of Process Control*, vol. 20, no. 3, pp. 248 – 260, 2010.

[11] S. Raković, B. Kouvaritakis, M. Cannon, and C. Panos, "Fully parameterized tube model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 12, pp. 1330–1361, 2012.

[12] D. Q. Mayne, M. M. Seron, and S. V. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[13] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The Int. Journal of Robotics Research*, vol. 20, pp. 378–401, 2001.

[14] T. Fraichard and H. Asama, "Inevitable collision states. a step towards safer robots?" *Advanced Robotics*, vol. 18, no. 10, pp. 1001–1024, 2004.

[15] R. Parthasarathi and T. Fraichard, "An inevitable collision state-checker for a car-like vehicle," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2007, pp. 3068–3073.

[16] D. Althoff, M. Werling, N. Kaempchen, D. Wollherr, and M. Buss, "Lane-based safety assessment of road scenes using Inevitable Collision States," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2012.

[17] D. Althoff, J. J. Kuffner, D. Wollherr, and M. Buss, "Safety assessment of robot trajectories for navigation in uncertain and dynamic environments," *Springer Autonomous Robots*, vol. SI Motion Safety for Robots, 2011.

[18] A. Bautin, L. Martinez-Gomez, and T. Fraichard, "Inevitable collision states: a probabilistic perspective," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2010.

[19] S. Karaman and E. Frazzoli, "High-speed flight in an ergodic forest," in *IEEE Conference on Robotics and Automation*, 2012, pp. 2899–2906.

[20] ——, "Linear temporal logic vehicle routing with applications to multi-uav mission planning," *Journal of Robust and Nonlinear Control*, vol. 21, no. 12, pp. 1372–1395, 2011.

[21] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon temporal logic planning," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2817–2830, 2012.

[22] A. I. M. Ayala, S. B. Andersson, and C. Belta, "Temporal logic motion planning in unknown environments," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5279 – 5284.

[23] M. Kloetzer and C. Belta, "Automatic deployment of distributed teams of robots from temporal logic specifications," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 48–61, 2010.

[24] I. Saha, R. Ramaithitima, V. Kumar, G. J. Pappas, and S. A. Seshia, "Automated composition of motion primitives for multi-robot systems from safe ltl specifications," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1525 – 1532.

[25] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[26] S. Prajna, "Barrier certificates for nonlinear model validation," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 3, 2003, pp. 2884–2889.

[27] P. A. Parrilo, "Structured semidenite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, 2000.

[28] S. Prajna, A. Jadbabaie, and G. Pappas, "A framework for worst-case and stochastic safety verification using barrier certificates," *Automatic Control, IEEE Transactions on*, vol. 52, no. 8, pp. 1415–1428, 2007.

[29] A. Majumdar and R. Tedrake, "Robust online motion planning with regions of finite time invariance," in *Algorithmic Foundations of Robotics X*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2013, vol. 86, pp. 543–558.

[30] 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.

[31] G. Lafferriere, G. Pappas, and S. Yovine, "A new class of decidable hybrid systems," in *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, F. Vaandrager and J. van Schuppen, Eds. Springer Berlin Heidelberg, 1999, vol. 1569, pp. 137–151. [Online]. Available: http://dx.doi.org/10.1007/3-540-48983-5_15

[32] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 2210–2215.

[33] T. Dang, "Vérification et synthèse des systèmes hybrides," Ph.D. dissertation, Institut National Polytechnique de Grenoble, 2000.

[34] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.

[35] D. Heß, M. Althoff, and T. Sattel, "Formal verification of maneuver automata for parameterized motion primitives," in *Proc. of the Int. Conf. on Intelligent Robots and Systems*, 2014, pp. 1474–1481.

[36] I. M. Mitchell, "Comparing forward and backward reachability as tools for safety analysis," in *Hybrid Systems: Computation and Control*, ser. LNCS 4416. Springer, 2007, pp. 428–443.