
Digital Analytics and Robotics for Sustainable Forestry

CL4-2021-DIGITAL-EMERGING-01
Grant agreement no: 101070405

# DELIVERABLE 3.5

## Field Demonstration: Per-Platform Reactive Planning Software Stacks

Due date: month 37 (September 2025)
Deliverable type: R
Lead beneficiary: NTNU

Dissemination Level: PUBLIC

# Contents

# 1 Introduction

Forests pose an array of significant challenges for robotic navigation methods. While many approaches for navigation are successful in various human-made environments, many of the underlying assumptions and algorithms break down in unstructured environments such as forests. One significant challenge is the chosen representation of the present obstacles. Here, explicit, rough approximations such as voxel grids [8] or octrees [2] can be fitted to various environments and then be used for downstream tasks such as motion planning. The recovered obstacle representations are highly dependent on various hyper-parameters that need to be tuned for one type of environment. The large spectrum of obstacle dimensions encountered in forests (1cm for a thin branch, 2m for a large tree trunk) and computational restrictions onboard aerial vehicles render the construction of accurate map representations at runtime for planning infeasible. The resulting coarse map representations may not capture all obstacles present in an environment to a sufficient degree.

In this report, additional means to enforce collision avoidance *reactively*, while utilizing the sensor readings directly, are demonstrated. We introduce varying complementary modules showing different approaches with respect to (a) the use of learning-based or classical optimal control formulations, (b) the sensor modalities and (c) whether a form of memory is used. The effectiveness in avoiding collisions due to mapping failures are demonstrated for three of the four proposed methods.

In combination with an arbitrary planning module, these methods for reactive navigation allow efficient task execution of the planned trajectories, while only modifying the executed motion if the system is at risk of colliding. This enables the creation of a multi-layered approach, effectively enhancing safety of robots employed in challenging forest environments.

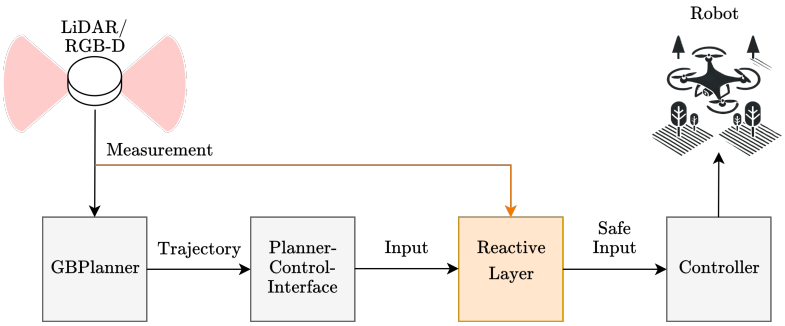


Figure 1: Overall planning architecture with the reactive layer used in this report.

# 2 Reactive Planning Software Stacks

## 2.1 RMF-Owl Robot "Kookaburra"

The RMF-Owl "Kookaburra" is a custom quadrotor platform developed within the DigiForest project as a testbed for advanced reactive planning and safety-critical control. The vehicle has a diagonal rotor-to-rotor span of approximately 0.65 m and a total mass of 2.6 kg including payload. It is equipped with a high-resolution 3D LiDAR sensor (ouster OS0) mounted on the frame for dense obstacle mapping, an onboard NVIDIA Jetson Orin NX compute module for real-time perception and control, an IMU for state estimation, and forward-facing stereo cameras for reactive navigation. All perception and control algorithms run fully onboard at high frequency, making the platform suitable for autonomous flight in GPS-denied and cluttered forest environments. The overall architecture of the reactive planning software stack is shown in Fig. 1. Here, the Baseline GBPlanner is coupled with a reactive safety layer, that only considers the locally available sensor observations to enforce collision avoidance in forest environments. While the safety layer does not strictly need to be purely reactive as in the case of the Model Predictive Controller presented shortly, the overall behavior of this layer is still a local, reactive one. We present three complementary methods that can individually form this reactive layer hereafter.

### 2.1.1 LiDAR–Composite CBF Safety Filter

The results of this subsection are a summary of the method presented in [3] To increase the safety of reactive flight in complex terrain, we implemented a LiDAR-based Composite Control Barrier Function (CCBF) safety filter as part of the Kookaburra's software stack. The safety filter is entirely independent of the planning framework used and designed to augment a navigation stack, acting as an additional safety-aware layer between the planner and the airframe. The method combines real-time 3D obstacle information from the onboard LiDAR with a control-theoretic safety filter. Individual distance constraints to each detected obstacle are formulated as control barrier functions and then combined into a single composite CBF, together with a thrust CBF. A quadratic program (QP) solves, at each control step, for the closest-possible control input to the nominal command that still satisfies the composite barrier and thrust constraints. This enables the vehicle to reject unsafe reference inputs without resorting to hard switching or conservative pre-planned margins. The conceptual design of the safety filter is shown in Fig.2.

To evaluate the effectiveness of the proposed CCBF-based safety filter, flight tests

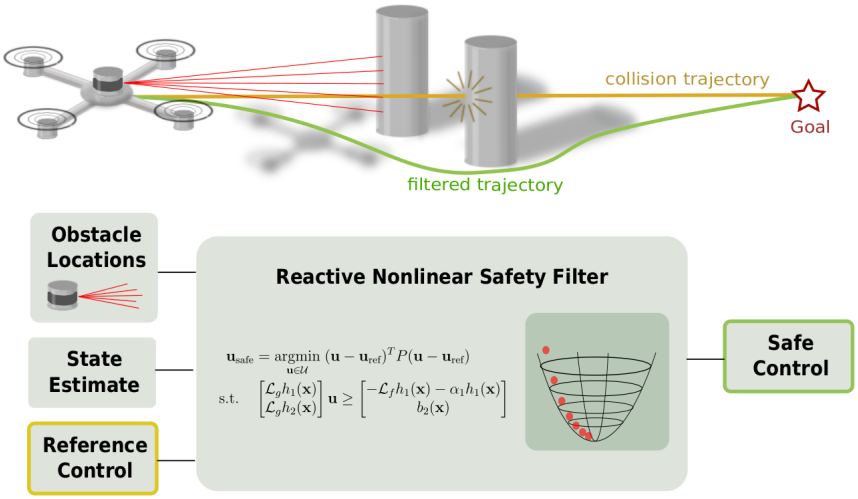


Figure 2: Architecture of the LiDAR–Composite CBF safety filter onboard the Kookaburra platform.

in a dense forest environment containing numerous tree trunks, branches and foliage were conducted. This scenario presents a highly cluttered, natural obstacle field with additional unmodeled disturbances such as light wind gusts. It thus provides a stringent benchmark for safe reactive navigation of aerial robots under unsafe reference inputs resulting from planning or localization errors.

In the experiment shown in Fig. 3, the nominal control inputs were deliberately set to be *adversarial*, commanding trajectories that would normally bring the vehicle close to or into obstacles. These adversarial reference velocities emulate a severe *planner failure*, where the onboard path planner issues unsafe commands. The CCBF safety filter ran onboard at high frequency and continuously modified these unsafe reference inputs via a quadratic program to ensure that collision and thrust constraints were satisfied.

The results demonstrate the robustness of the proposed method. Throughout the

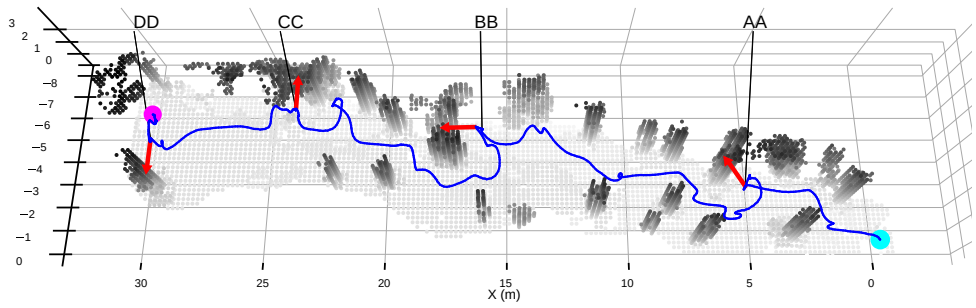


Figure 3: Aggregated map and path of the CCBF experiment. The mission starts at the cyan circle and ends at the magenta circle. The robot receives an adversarial velocity command from a human operator, actively trying to collide with tree trunks, branches and bushes. The red arrows depict this reference velocity for some selected time instances, A-D.

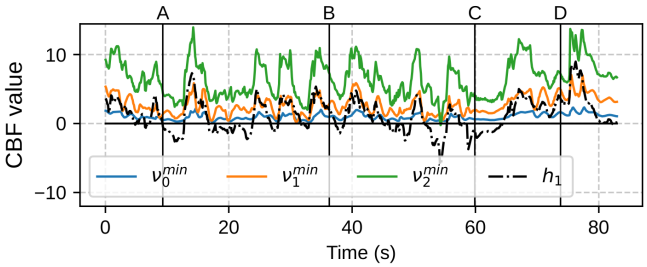


Figure 4: Value of the CCBF (h1) and it's higher order sets during the adversarial mission. Positive values indicate safety, negative values indicate unsafe behavior. The line slightly drops below the 0-line due to environmental disturbances and tracking errors but quickly recovers back to safety.

entire flight, the quadrotor maintained safe separation from all detected obstacles, successfully avoiding collisions despite the adversarial commands. Although the composite barrier function value occasionally dipped slightly below the constraint threshold (zero, shown in Fig. 4) due to environmental disturbances and sensing noise, no collision constraints were violated during the experiments. This is attributed to the fact that the CCBF is naturally robust to bounded disturbances such as wind. In each case the safety filter intervened and overrode the unsafe nominal command with only minimal deviation from the desired trajectory.

Overall, this experiment confirms that the proposed CCBF approach can enforce safety in real time even under faulty planning commands in complex, unstructured forest environments.

### 2.1.2 Neural MPC Local Planner

Another methodology developed for local collision avoidance in Digiforest is a Model Predictive Controller (MPC) utilizing a neural network representing a Signed Distance Function (SDF) from range measurements. Similar to the CBF methods described earlier, the method does not rely on any consistent map and only utilizes the current observations and odometry estimates to generate safe trajectories within the visible sensor frustum. However, this method is not purely reactive as it considers a short planning horizon to predict future states and optimal inputs, resulting in a smooth and more predictive behavior.

Furthermore, the method has been evaluated for both LiDAR and RGB-D type sensors and undergone extensive testing in forest environments with numerous adversaries such as naive, unsafe planner inputs and inaccurate (drifting) odometry estimates.

In the first field demonstration experiment, we demonstrate the ability of the reactive planning layer to ensure collision avoidance and local re-planning under faulty velocity commands generated by a human operator. In this case, a LiDAR sensor is used to obtain range measurements at a frequency of 20 Hz. A figure explaining the experiment is shown in Fig. 5. At the instance highlighted in the figure (A), the unsafe reference commands and the identified SDF level set are plotted together with the predicted, safe trajectory produced by the MPC avoidance scheme. This experiment highlights the ability of the method to enforce collision avoidance even under worst-case planner failures such as unmapped obstacles.

In the second experiment, illustrated in Fig. 5, we demonstrate that the system

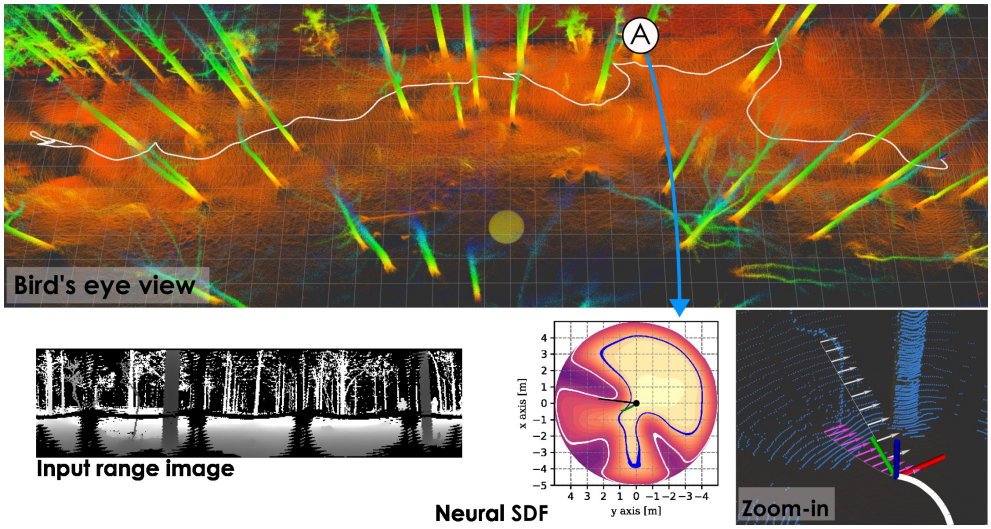


Figure 5: Visualization of the neural MPC-based planner during flight among trees. The lower part of the figure depicts three zoomed-in time instance A (adversarial reference as white arrows, actual nmpc actions as pink arrows), along with the $z_B = 0$ slice of the neural SDF. and the input LiDAR range image.

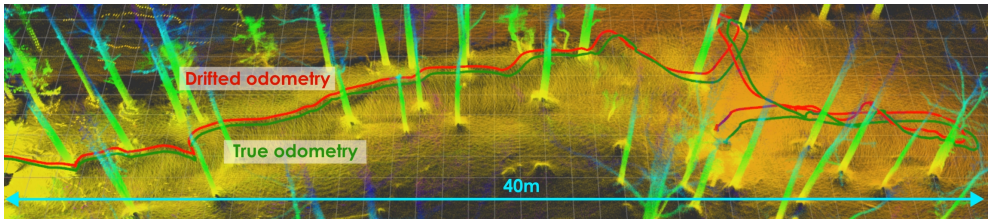


Figure 6: Bird's eye view of the trajectory with both odometries (drifted in red, ground truth in green).

remains resilient to large odometry drift. Similar to the previous case, the LiDAR range image is used for maintaining collision avoidance in the presence of an adversarial velocity reference. However, instead of relying on the accurate LiDAR-based localization, we implement an FMWC radar-based velocity estimator, which results in a drifting position and heading odometry. Indeed, the radar odometry has an RPE of 3% and a heading RPE (Relative Position Error) of 2° against the LiDAR odometry from [5], which we use as ground truth for comparison purposes. This highlights the local aspect of the method, which achieves collision avoidance while relying on local, short-term consistency of the state estimate.

## 2.2 LMF Robot "Jacamar"

### 2.2.1 Neural CBF Safety Filter

The method presented in this subchapter is inspired by the work [4]. Here a learned barrier function trained entirely in simulation is used to reactively modify unsafe control inputs inside a safety filter. While the overall architecture of the work presented here is similar to the work [4], the formalism for training the barrier certificates is

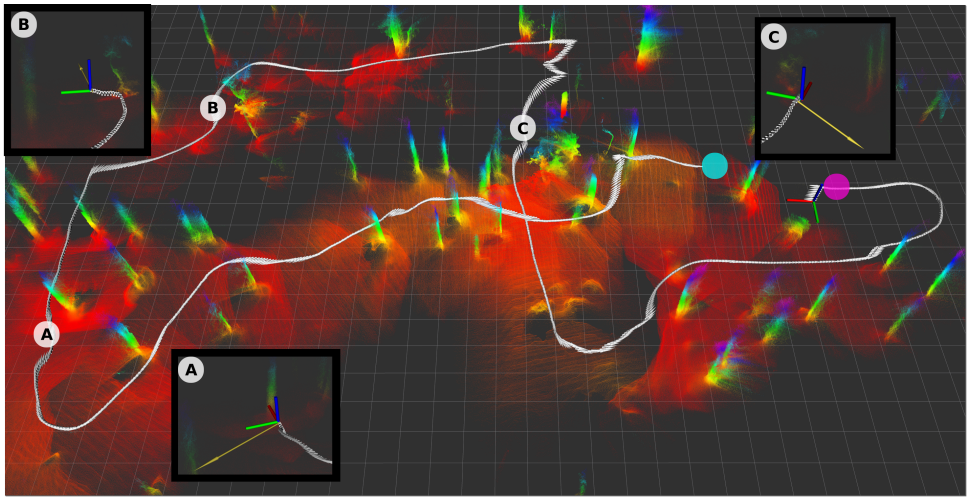


Figure 7: Overview of the field demonstration with the neural CBF safety filter. The missions starts at the cyan circle and ends at the magenta circle, with the time instances A, B, C highlighted with close up views including the modified acceleration command (dark yellow). The safety filter modifies the unsafe reference command of the human operator (simulating a planner failure) to avoid collisions.

fundamentally different. First, in this report, we assume a 3-dimensional system model to not constrain the motion of the robot to a horizontal plane. Further, the CBFs are trained using a deep-learning approximation of the lagrange-dual-constrianed formulation [1]. The resulting framework exhibits a similar behavior to the CCBF safety filter presented earlier. However, the main advantage of the neural CBF presented here is the enhanced resilience to sensor noise and the more permissive overall behavior.

## 2.3 SRL Drone

The SRL drone is a micro aerial vehicle (MAV) that operates in a purely vision-based configuration. The main advantage of relying on cameras is their low weight, which is particularly important for MAVs, since any increase in payload reduces mission duration. Furthermore, cameras provide a higher density of measurements with better ability to detect fine structures such as branches. Although alternative sensing modalities such as LiDAR or Radar enable longer detection ranges, they do so at the cost of measurement density. As a consequence, detecting thin structures becomes more difficult compared to the dense depth estimates achievable with stereo cameras.

Regardless of the sensor modality, the drone should only plan to move inside the sensor's view frustum. In this regard, the main drawback of cameras is their comparatively narrow field of view, typically up to 120° horizontally, whereas LiDAR can offer full 360° coverage. Managing this limitation is the core motivation of our reactive planning module.

Yet, the choice of a camera enables the rest of the drone to be as lightweight and cost-efficient as possible, assembled from common components for less than 3000 Euros. The onboard computing is provided by a Jetson Orin NX. For perception, a Realsense D455 is used and an OakD W is currently being implemented, which offers a wide field of view to lessen the main drawback of single camera systems.

### 2.3.1 Control Pipeline

As with the reactive planning formulations presented in previous sections, we use a global planner for high level planning through the map. An RRT* algorithm approximates the shortest path to the goal by checking for collisions in a Supereight2 [9] map. For state estimation, we use Okvis2X. To enable efficient large-scale mapping and planning, the global map is subdivided into submaps that align with the SLAM posegraph [6]. In order to deal with the map's limited resolution and potentially dynamic obstacles, the reactive planner described in the following section is used. The trajectory it generates is processed by a linear model predictive controller (MPC) computing attitude setpoints based on the reference trajectory and odometry estimates. A PX4 flight controller uses a cascaded PID control structure to compute motor RPM through the attitude setpoints.

### 2.3.2 Reactive Planning Module

The reactive planner optimizes trajectories in Euclidean space, modeling the MAVs position, velocity, and acceleration along the planning horizon. It uses a standard MPC contouring [7] formulation with energy terms that penalize deviation from the reference path, accelerations and jerk and thus indirectly large actuation outputs, and encouraging progression along the path. Hard constraints are imposed on acceleration and jerk, as well as the robot's collision with the environment. These collision constraints are evaluated by computing the shortest distance from each position on the trajectory to the closest unprojected image pixel, using parallelization with custom CUDA kernels. This method is currently in development – while a prototype has been evaluated to work in simulation, several challenges are currently being tackled. As in previous sections, the reactive planner's optimization is formulated in the sensor space directly to achieve high computational efficiency required for quick reactive behavior. However, with a camera's comparably narrow field of view, movements outside the view frustums need to be allowed to not overly restrict the MAV's mobility, which becomes especially relevant when flying along the gravity direction. However, this can lead to unsafe situations where obstacles avoided in one iteration may not be visible in future iterations as they move out of the sensor's viewing frustum (Fig. 8). This motivates the use of some form of memory, whose formulation needs to trade off computational efficiency, which both prohibits expensive map integration yet requires a complete representation of close-by observed free space. This requires not only trusting the most recent sensor measurements only, but also considering older measurements. Ongoing research investigates how such memory could model past observations and efficiently evaluate free space, to ensure safe planning.

## 2.4 ANYmal Reactive Planning Stack

We refer to the previous deliverables for a comprehensive description (**D3.4**) and a field demonstration (**D6.3**) of the reactive planning of the quadruped platform ANYmal. Due to overlap, these results are not re-introduced here again to avoid confusion.

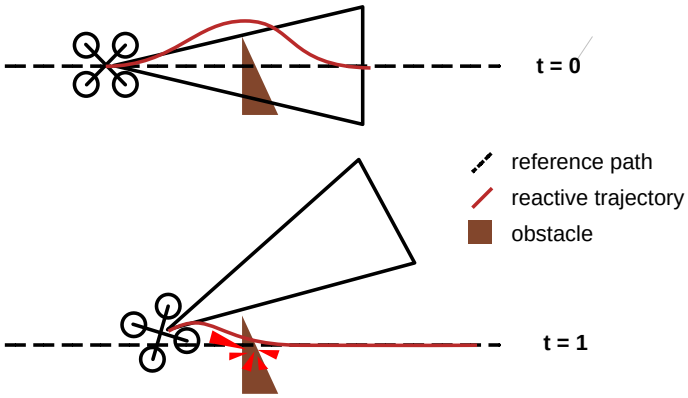


Figure 8: To not overly restrict the MAVs mobility, planning outside the view frustum must be allowed, which is unsafe, however, as obstacles might become invisible. This motivates the use of a map or a similar type of memory.

# References


[1] Ferdinando Fioretto, Pascal Van Hentenryck, Terrence WK Mak, Cuong Tran, Federico Baldo, and Michele Lombardi. "Lagrangian duality for constrained deep learning". In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2020, pp. 118–135.

[2] Nils Funk, Juan Tarrio, Sotiris Papatheodorou, Marija Popović, Pablo F. Alcantarilla, and Stefan Leutenegger. "Multi-Resolution 3D Mapping With Explicit Free Space Representation for Fast and Accurate Mobile Robot Motion Planning". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3553–3560. DOI: 10.1109/LRA.2021.3061989.

[3] Marvin Harms, Martin Jacquet, and Kostas Alexis. "Safe quadrotor navigation using composite control barrier functions". In: *arXiv preprint arXiv:2502.04101* (2025).

[4] Marvin Harms, Mihir Kulkarni, Nikhil Khedekar, Martin Jacquet, and Kostas Alexis. "Neural control barrier functions for safe navigation". In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2024, pp. 10415–10422.

[5] Shehryar Khattak, Huan Nguyen, Frank Mascarich, Tung Dang, and Kostas Alexis. "Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments". In: *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2020, pp. 1024–1029. DOI: 10.1109/ICUAS48674.2020.9213865.

[6] Sebastián Barbas Laina, Simon Boche, Sotiris Papatheodorou, Dimos Tzoumanikas, Simon Schaefer, Hanzhi Chen, and Stefan Leutenegger. *Scalable Outdoors Autonomous Drone Flight with Visual-Inertial SLAM and Dense Submaps Built without LiDAR*. 2025. arXiv: 2403.09596 [cs.RO]. URL: https://arxiv.org/abs/2403.09596.

[7] Denise Lam, Chris Manzie, and Malcolm Good. "Model predictive contouring control". In: *49th IEEE Conference on Decision and Control (CDC)*. 2010, pp. 6137–6142. DOI: 10.1109/CDC.2010.5717042.

[8] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 1366–1373.

[9] Emanuele Vespa, Nikolay Nikolov, Marius Grimm, Luigi Nardi, Paul H. J. Kelly, and Stefan Leutenegger. "Efficient Octree-Based Volumetric SLAM Supporting Signed-Distance and Occupancy Mapping". In: *IEEE Robotics and Automation Letters* 3.2 (2018), pp. 1144–1151. ISSN: 2377-3766. DOI: 10.1109/LRA.2018.2792537.