

Autonomous and Safe Inspection of an Industrial Warehouse by a Multi-rotor MAV

Alexandre Eudes,
Julien Marzat,
Martial Sanfourche,
Julien Moras,
Sylvain Bertrand

Abstract This paper reports field tests of autonomous inspection in an industrial indoor facility by a Micro-Air Vehicle (MAV) with no prior knowledge on the environment. Localization, mapping and safe navigation is achieved using only the embedded sensors (stereo-vision, IMU, laser altimeter) and with the entire perception and control loop running on-board of the MAV. An overview of the algorithmic architecture and design choices is provided and the focus is put on mission and safety capabilities that have been demonstrated via several flight tests defined in association with SNCF (French Railways) in one of their train storage warehouse.

1 Introduction

Building inspection and surveillance can benefit from the use of Micro Air Vehicles (MAVs) to provide additional or complementary data (in terms of nature or point of views) that can be used for diagnosis and maintenance. If outdoor inspection of buildings involves taking measurements at high altitudes, for which the use of MAVs can be helpful, this is also the case for indoor inspection of industrial warehouses. In this case, MAVs can be used to perform autonomous and repeatable inspections inside the warehouse to assess the state of the building (walls, pillars, high altitude structures, ceiling, etc.) or provide information on objects of interest stored or located inside (number and location of goods, for example).

This paper considers autonomous inspection by a multi-rotor MAV inside a warehouse owned by SNCF (national French railway company) where maintenance of trains and storage of spare parts are performed. Several major issues arise regarding such a task. First the MAV should be able to localize itself in an indoor environment (no GPS) while taking into account possible changes in this environment (stored items moved between two flights). The flight and inspection should be fully auto-

All authors are with ONERA – The French Aerospace Lab, F-91123 Palaiseau, France
e-mail: firstname.lastname@onera.fr

matic, from take-off to landing, to lessen the dependence on a human telepilot and increase repeatability. In addition, robustness with respect to signal loss must be ensured as the vehicles evolve over long distances in a potentially electro-magnetically perturbed environment (motor coils, etc.). Finally, safety must be guaranteed with respect to possible workers entering the area of inspection.

Related work

In [13], Shen et al. present a MAV system able to navigate indoor, inside a multi-floor building. They use a setup composed of one lidar, one camera and an IMU to demonstrate self-localization, mapping, path planning and autonomous control functionalities. In [3], Fraundorfer et al. propose to use a MAV for autonomous exploration and mapping in GPS-denied environment. The MAV is equipped with a stereo camera, an optical flow camera looking downward and an ultrasonic range sensor. The MAV achieves visual odometry using the stereo camera which is fused with the attitude estimated from IMU, the optical flow of the third camera and the ultrasonic range measurement. The system computes a 3D occupancy grid that is used to make a frontier-base exploration of the area. In [11] Omari et al. propose to use a remotely operated MAV for industrial inspection applications. Their platform is equipped with a stereo camera sensor that is used to compute visual odometry and state estimation tightly coupled with an IMU. The MAV also computes an Octomap representation of the environment which helps the pilot by preventing collisions. In [1], Beul et al. use a hexarotor MAV controlled by a 3DR Pixhawk Autopilot. The perception stack uses an omni-directional and heterogeneous sensor system composed of three stereo cameras and two Hokuyo lidar sensors mounted on a moving plate. They achieve localization by combining visual odometry and lidar, as well as environment modeling in order to avoid obstacles.

In [2], Fang et al. address the problem of robust autonomous navigation in difficult environments like inside a ship. Their work propose a robust pose estimation system based on the fusion of RGB-D sensor, downward optical flow camera, downward lidar punctual sensor and an IMU using a Monte Carlo approach. This is coupled with a path planning system based on Receding Horizon Control to navigate between way-points.

In line with these research efforts, the proposed solution is based on the development of a fully embedded software architecture exploiting a stereo-vision system (in association with an IMU and a laser telemeter) which performs on-board all the functionalities required for safety and mission fulfillment: localization and mapping, planning, control, obstacle detection and avoidance, safety supervision. This paper presents this on-board architecture and its validation through field flight experiments in a large-scale indoor facility with operational interest for industrial End-Users.

The paper is organized as follows. In the next Section, a description of the industrial warehouse, of the MAV platform and of the scenarios of inspection are proposed. The overall on-board architecture and the associated algorithms are described in Section 3. Section 4 addresses the supervision of the flight and the safety functionalities. Finally, before concluding remarks, results of flight validation experiments are proposed in Section 5 for the considered scenarios of inspection.

2 Environment, MAV platform and scenarios descriptions

2.1 Industrial warehouse

The industrial warehouse in which inspections have to be performed belongs to the French national railway company (SNCF) and is located in Sotteville-lès-Rouen. It is used as a maintenance workshop for trains and as a storage place for spare parts (see Figure 1). The part of the warehouse dedicated to storage has been used for the flight experiments presented in this paper. The corresponding flight volume is defined by a ground area of 25 m by 50 m with an altitude of 10 m under ceiling.



Fig. 1: Industrial environment: SNCF infrastructure warehouse in Sotteville, France

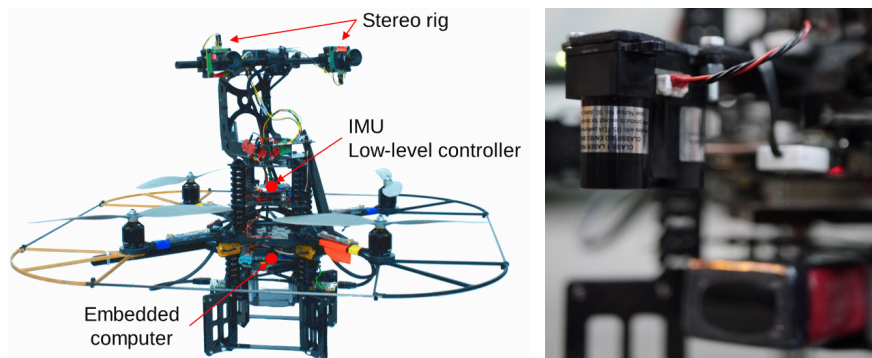


Fig. 2: ONERA MAV (based on an Asctec platform) with stereo-vision sensors, lidar altimeter (right) and embedded processing capabilities

2.2 *MAV platform*

The flight experiments described here are conducted on a Pelican quadrotor base from Asctec (Figure 2). The proprietary Flight Control Unit (FCU) includes IMU, 3D-magnetometer and processors dedicated to low-level control (attitude stabilization). The FCU is linked to an embedded computer containing an Intel i7 quad-core (2.1GHz) which handles the complete perception and control chain presented in Figure 3. All processing is achieved on-board so the MAV is fully autonomous and robust to communication loss during the execution of the mission. In addition to the IMU, the sensor setup includes a 22cm-baseline stereo-rig composed of two USB2 cameras, electronically synchronized, offering a 100° field of view and a laser telemeter pointing downward for altitude measurement (for take-off and landing).

2.3 *Inspection scenarios*

Typical scenarios of inspection in the warehouse consist in recording pictures or measurements provided by additional on-board sensors (e.g. FLIR camera) of different objects of interest: walls, high altitude structures, spare parts stored in the warehouse. This requires to automatically take-off, fly over long distances to reach areas of inspection, perform the inspection and fly back for automatic landing. During the flight, obstacle avoidance can be performed in case of trajectories conflicting with detected static or dynamic obstacles. The trajectories of inspection are defined by way-points or trajectory coordinates given either in the global frame associated to the warehouse or in the local frame associated to a specific point whose global location is known or detected during flight. In addition, sensor characteristics (range, field of view, orientation) must be taken into account for the inspection as the mission performance is directly impacted by the MAV trajectory (measurement overlap between two parts of the trajectory, relative distance and/or line-of-sight constraints to the object to be inspected, etc.).

Two scenarios representative of realistic inspection tasks have been defined. The first one consists in inspecting objects of interest defined by global coordinates in the warehouse. More precisely, we chose two elements to be inspected, a metallic structure located at an altitude of 7m that the MAV must follow and inspect and a load located over a platform around which the MAV must turn in order to take pictures. The second scenario consists in performing the inspection of a wall section whose location is not a priori precisely known, but which can be detected by some specific characteristic (e.g. visual tag). To also validate safety requirements, avoidance situations with respect to static and mobile obstacles are considered in a third scenario.

3 On-board navigation system

The embedded navigation system uses ROS to link together the numerous software components of the global architecture from Figure 3. Besides hardware related components (interface with sensors or other computers), we distinguish 3 main functional blocks described in the following subsections: multi-sensor state estimation, environment modeling and control.

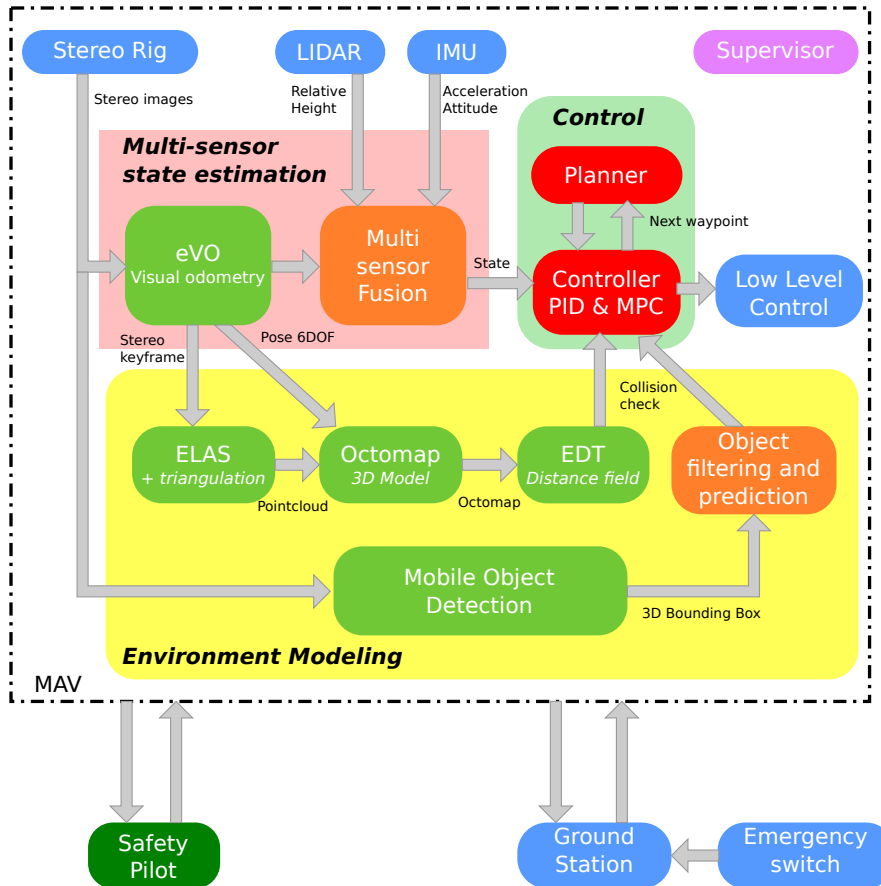


Fig. 3: Perception-control loop for autonomous navigation with obstacle avoidance

3.1 Multi-sensor state estimation

The low-level controller provides a reliable estimate of the orientation. Therefore, the vehicle state vector contains only the position and linear velocity in the reference frame aligned with gravity and centered on the IMU-frame origin at the beginning of the mission. The state is estimated by a linear Kalman filter as in [8]. Given the initial pose, the state is predicted by integration of accelerometer and attitude mea-

surements at 100 Hz. The correction is performed at video-rate (20 Hz) thanks to the pose computed from stereo-images by eVO [12], a keyframe-based visual odometry algorithm. eVO builds a map of isolated landmarks automatically initialized from images (no prior map) as in Visual-SLAM. Compared to state-of-the-art SLAM and other odometry algorithms, eVO is oriented towards low computational cost and includes some simplifications concerning the map updates: the landmarks are pruned when they leave the sensor field of view and the landmarks localization is not refined in a multi-view optimization scheme. Nevertheless, the drift on the localization is of the class 1% of the traveled distance. A detection of outliers is performed on the innovation of the Kalman filter which compares the state predicted using only the IMU with the position computed by visual odometry. If this difference is larger than the expected MAV velocity, an alarm is raised and the emergency mode activated (see Section 4). A second Kalman filter estimates independently the ground height and vertical velocity from the laser telemeter measurements and IMU attitude. It is only used during automatic take-off and landing or emergency stabilization.

3.2 *Environment modeling*

In order to be able to avoid collision in such a cluttered environment, mapping is performed on board. To handle both static and mobile objects, we consider a dual model representation. An occupancy grid approach is used to model the static environment while a feature-based (bounding box) approach is used for mobile object detection.

The static mapping module is composed of 3 subtasks as shown in Figure 3. First, a stereo matching algorithm is used to compute a dense disparity map from a rectified pair of stereo images. Here, we used the ELAS (Efficient Large-scale Stereo Matching) algorithm [4]. This disparity map is used to triangulate a dense point-cloud that is integrated in a well known Octree-based occupancy grid representation named Octomap [6]. This approach models the space with a set of voxels organized into an Octree structure where each voxel can be either occupied or free. A Bayesian estimation is performed, so each voxel stores its probability to be occupied and the cell then a MAP decision rule is taken when needed. Note that this probabilistic framework ensures that mobile obstacles are not included in this map, since several successive observations of the same point are mandatory. Finally, with the resolution of the map used (0.1m), an Octomap representation for checking collisions over each potential trajectory would be inefficient, because of the large number of voxels to be tested. A solution to this problem is to additionally compute a distance-field to the closest obstacle during the map update. For this purpose, we apply an Euclidean Distance Transform (EDT) [7] to the Octomap. This mapping updating process is performed at each keyframe.

The detection of mobile objects is based on the analysis of sparse optical flow computed on Harris corners, which are located in 3D by stereo. We assume here that mobile objects are clusters of features which violate two-views and three-views

geometrical constraints between two successive stereo-images and spatially consistent with the dimensions of a pedestrian. The first step consists in classifying into moving and static features. The classification is done in a two-tests cascade, firstly by the OpenCV robust RANSAC-based fundamental matrix estimator then, for the inliers, by the robust pose estimator implemented in eVO [12]. In a second step, the candidate moving image features are clustered. A Delaunay triangulation gives access to the rough 3D structure localized in inertial frame thanks to the pose estimated by the multi-sensor state estimator (cf. subsection 3.1). Triangles are pruned according to their size and orientation (only the most vertical ones are retained). The remaining connected triangles form candidate objects. We finally compute 3D bounding cylinders and select those whose width is inside a specified range. The mobile object raw detection is fed into a Kalman filter tracker in order to remove false alarms and to make prediction of the obstacle position in the next steps. The state vector is composed of the object position, its height and radius. A constant velocity model is used to predict the future positions of the object for avoidance purpose. The entire processing takes less than 100 milliseconds on the MAV embedded CPU.

3.3 Safe way-point navigation and trajectory tracking

The control algorithm is designed to track a reference trajectory or to reach a way-point defined by the user (in the way-point case, a straight-line trajectory at constant velocity is generated). Since the environment can be highly cluttered and no prior assumption is made, an avoidance module for static or mobile obstacles has been defined. It relies on a model predictive control (MPC) algorithm which exploits the perception information from the previous subsection. A MPC scheme relies on a prediction of the dynamical model on a time horizon to take into account future MAV behavior and interaction with the environment. The design of the controller relies on a simplified linear acceleration model, for which an analytical MPC linear quadratic solution has been obtained for trajectory tracking in the nominal case.

To guarantee the safety of the vehicle and the operators, the MPC algorithm is adapted to avoid any obstacle that can be found on the reference trajectory. The distance map information on the static environment and the future mobile object positions are combined to obtain a single distance to the closest obstacle at each position of the MAV predicted trajectory. If there is a risk of collision on the nominal predicted trajectory, an additional control input is then selected in a predefined discretized set so as to optimize a cost function on the prediction horizon which is a weighted sum between tracking the reference trajectory and respecting a parameterized safety distance from any obstacle [9]. The resulting acceleration control input, which is equal to the sum of the nominal and avoidance inputs, is then translated into thrust, roll angle, pitch angle and yaw rate and forwarded to the low-level Asctec controller.

4 Flight management and safety functions

We have implemented a flight manager which supervises the execution of the automatic mission. It includes a mission scheduling editor, a mission scheduler associated to a state machine, the supervision of critical functions and an emergency manager. All of these features contribute to make MAV operation easier and more robust to sensors or software failures, which is of tremendous importance when such automatic functionalities are deployed on the field (where no motion capture system is available, unlike in laboratory).

4.1 Mission planning and execution

A mission scheduler allows the operator to define the content of the mission and to configure and execute each task during the mission. The plan is built from a combination of simple tasks such as: way-point goto, viewpoint constraints, waiting steps, user confirmation and more high level tasks like: take-off, land, start wall inspection. A 3D representation of the plan allows an easy mission planning, verification and visualization of the execution (Figure 7).

The autonomous mission execution is managed by the state machine depicted in Figure 4. The automatic take-off is triggered by the safety pilot who starts the engine and pushes the thrust stick up to a predefined level, settled near the stability thrust level for easier manual recovery action. It should be noted that after these basic human tasks all the missions are carried out in full autonomy. After the take-off, an on-line calibration permits to adjust some controller parameters. The mission starts and the predefined plan is executed after the MAV reaches and keeps hovering around a target point at a given altitude above the starting point. The landing mode is activated either when the MAV reaches the landing way-point or if the ground station operator triggers an emergency landing. In the nominal case, the landing is done autonomously thanks to relative altitude measurements from laser telemeter. In the emergency case, depending of the state of the MAV, the emergency landing can be fully autonomous or semi-autonomous.

4.2 Supervision and Emergency manager

In order to handle any issues occurring during the mission, an emergency stack is running on board. This stack is composed of a supervisor monitoring different parameters (module status, sensor activity, etc.). An emergency event is triggered when some parameters are abnormal: loss of a software node, inconsistency between vision and IMU, no safe trajectory found by the MPC module. An emergency event can also be triggered when the ground station operator activates the emergency switch or when the safety pilot pushes any stick outside of a dead zone. Fig-

ure 5 presents the different states of the emergency stack. The event sets the MAV in *Emergency Stop* mode. In this mode the MAV suspends its mission and holds its position using all functional sensors and modules. In the case where all critical functions are faulty, the MAV sends a warning signal to the safety pilot and falls back in *Manual* control mode. When the MAV is in *Emergency Stop*, the ground station operator can evaluate the situation in order to check what triggered the emergency event. Finally, he can decide to go out of emergency mode and continue the mission, to activate an *emergency landing* or to ask the safety pilot to recover the MAV in *manual* mode.

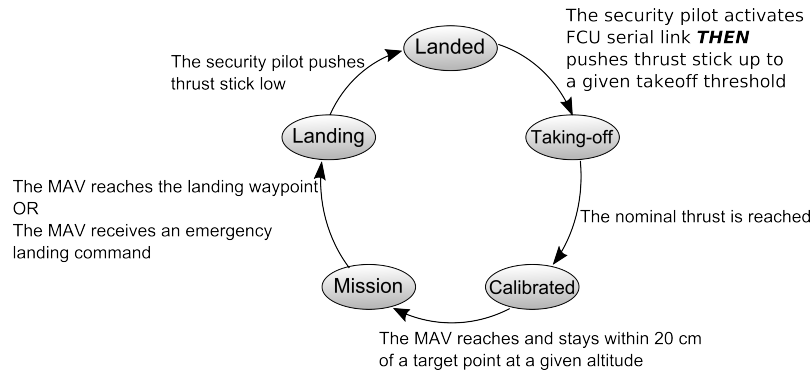


Fig. 4: State machine for flight management

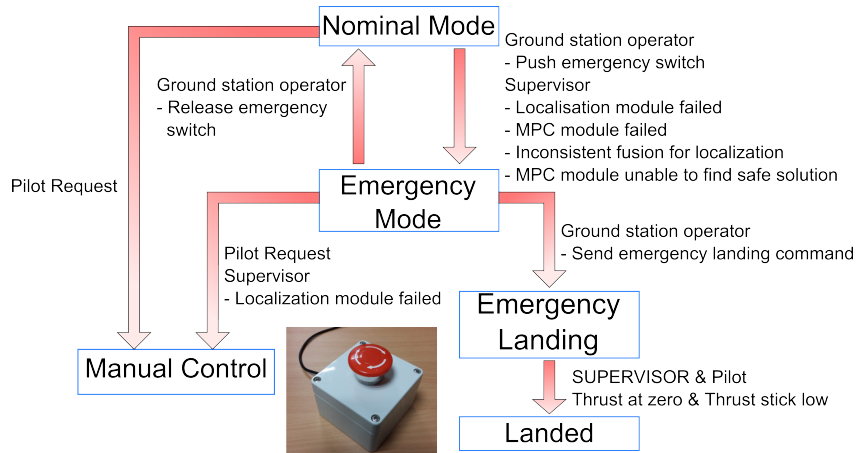


Fig. 5: Emergency management

5 Flight experiments

Scenario 1

In this first scenario, we demonstrated inspection tasks planned before the mission. This type of mission definition is especially useful for periodic inspection. Different types of trajectory are provided to allow an easy and efficient planning (example in Figure 6). Figure 7 shows the inspection plan of a bridge and two pillars which is very difficult to execute by a human pilot due to height and visibility conditions. We also added the possibility to define a viewpoint function. When this function is active, the yaw is controlled to ensure that a target point is always kept in the field of view. This allows the operator to define trajectories that automatically observe an object from multiple points of view (Figure 8).

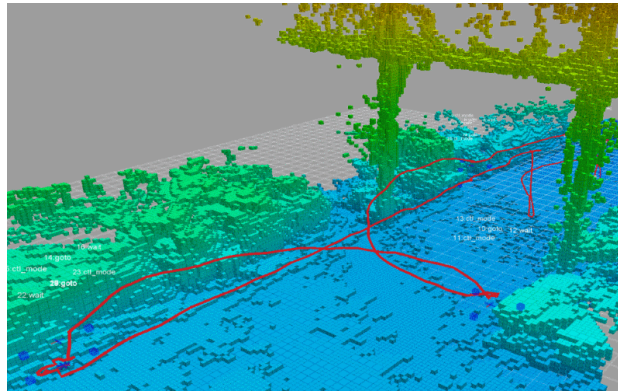


Fig. 6: 3D model autonomously built on-board by the MAV while following a long-distance reference trajectory (in red) composed of portions of ellipses and lines

Scenario 2

An automatic wall inspection task has been developed. This task consists in scanning a wall and ensuring its complete coverage by a dedicated sensor. The wall scanning zone is defined by an AprilTag [10] located at the center, predefined dimensions (height and width) and a desired scanning distance. The execution of the task is carried out as follows. When the tag is detected, the MAV is guided to a position in front of the wall to be able to correctly estimate the AprilTag orientation. The orientation of the wall plane is then refined robustly from the point-cloud computed by ELAS [5]. Assuming this model, the desired scanning distance and the field of view of the payload sensor, a trajectory that covers the wall with the given field of view is planned as a succession of linear segments that are then tracked by MPC.

Scenario 3

To tackle inaccuracies or mistakes during preflight mission planning, changes in the environment or presence of mobile obstacles, the internal MPC algorithm

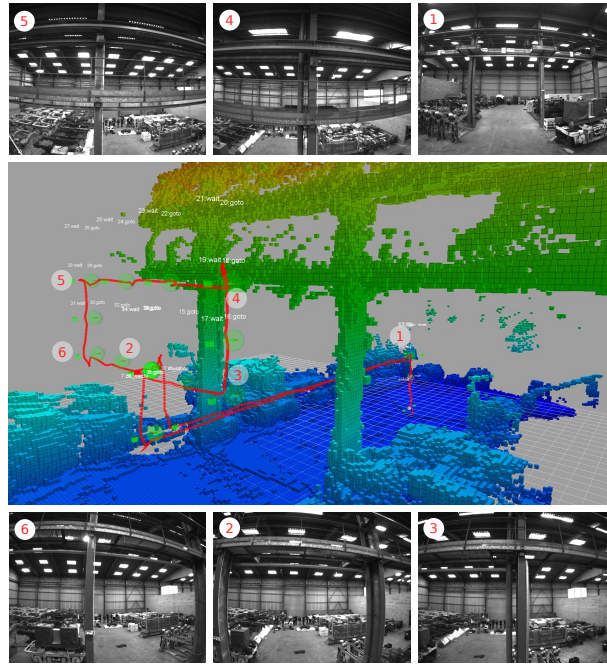
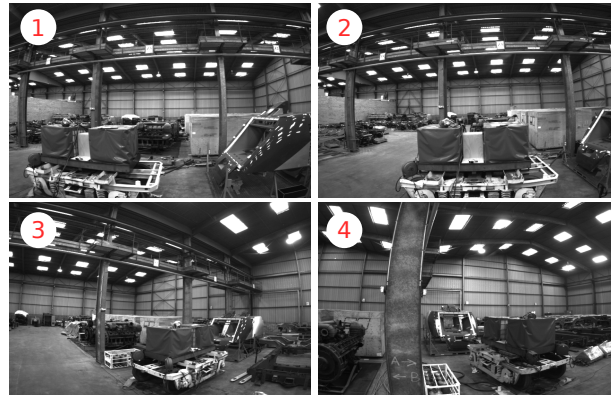


Fig. 7: Trajectory and 3D model built during the inspection scenario. The mission scheduling appears as white text indications over the map.

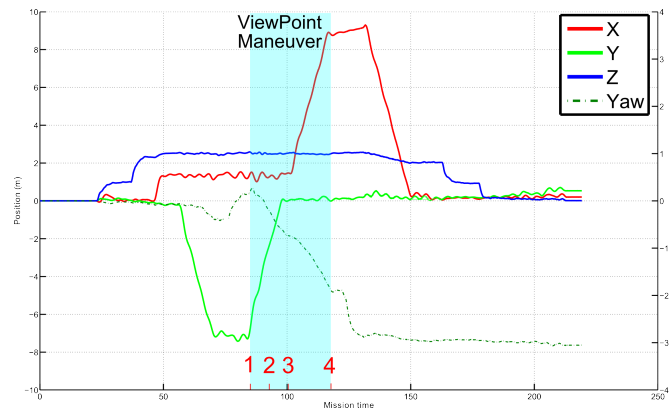
ensures collision avoidance during the plan execution. Figure 10 shows an example where the original plan (in thin red) passes through a pole. During the execution, the safety distance constraints were not satisfied and the MPC planned reactively an avoidance trajectory (final trajectory in bold red). Figure 11 illustrates the successful avoidance of a mobile obstacle with the same algorithm. These safety situations have been successfully repeated several times.

6 Conclusion

In this work, we demonstrated the capability for a MAV to navigate autonomously in an unknown GPS-denied environment in order to achieve different missions in an industrial context. The system presented relies on advanced embedded functionalities such as visual odometry, multi-sensor fusion, environment modeling and path planning and control. The complexity and diversity of the different scenarios tested on the field led us to add a flexible mission manager and several safety functionalities in order to increase the reliability of the system. Field experiments in an industrial facility have shown the soundness of the proposed approach. This is an important



(a) ViewPoint Snapshots



(b) ViewPoint Trajectory

Fig. 8: Action of the viewpoint constraint on a L-shape trajectory. The targeted point is located behind the trolley.

step towards the operational deployment of such autonomous MAV solutions for inspection, which will make it possible to relax the constraints on human operators.

Acknowledgements The authors thank SNCF for their support to this work, partially funded by the research partnership between ONERA and SNCF.

References

1. M. Beul, N. Krombach, Y. Zhong, D. Droschel, M. Nieuwenhuisen, and S. Behnke. A high-performance mav for autonomous navigation in complex 3d environments. In *2015 Interna-*

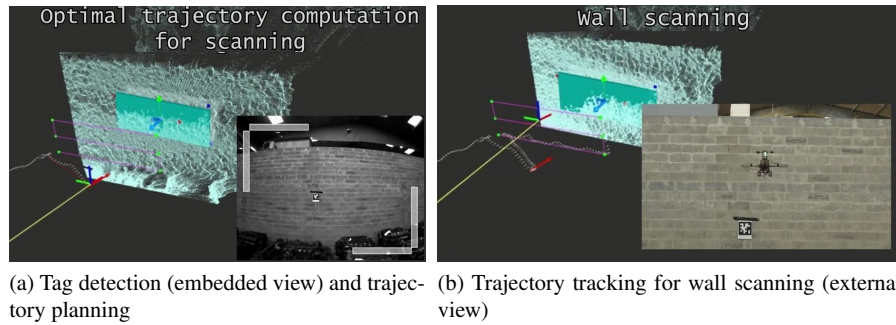


Fig. 9: Automatic wall inspection modus operandi.

- tional Conference on Unmanned Aircraft Systems (ICUAS)*, 2015.
2. Zheng Fang, Shichao Yang, Sezal Jain, Geetesh Dubey, Silvio Mano Maeta, Stephan Roth, Sebastian Scherer, Yu Zhang, and Stephen T. Nuske. Robust autonomous flight in constrained and visually degraded environments. In *Field and Service Robotics*, June 2015.
 3. F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys. Vision-based autonomous mapping and exploration using a quadrotor MAV. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal*, pages 4557–4564, 2012.
 4. A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *Proceedings of the 10th Asian Conference on Computer Vision (ACCV), Queenstown, New Zealand*, pages 25–38, 2010.
 5. Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Asian conference on computer vision*, pages 25–38. Springer, 2010.
 6. A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.
 7. B. Lau, C. Sprunk, and W. Burgard. Efficient grid-based spatial representations for robot navigation in dynamic environments. *Robotics and Autonomous Systems*, 61(10):1116–1130, 2013.
 8. S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart. A robust and modular multi-sensor fusion approach applied to MAV navigation. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan*, pages 3923–3929, 2013.
 9. J. Marzat, S. Bertrand, A. Eudes, M. Sanfourche, and J. Moras. Reactive MPC for autonomous MAV navigation in indoor cluttered environments: Flight experiments. In *Proceedings of the 20th IFAC World Congress, Toulouse, France*, 2017.
 10. Edwin Olson. AprilTag: A robust and flexible visual fiducial system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3400–3407, 2011.
 11. S. Omari, P. Gohl, M. Burri, M. Achtelik, and R. Siegwart. Visual industrial inspection using aerial robots. In *3rd International Conference on Applied Robotics for the Power Industry, Foz do Iguaçu, Brazil*, pages 1–5, 2014.
 12. M. Sanfourche, V. Vittori, and G. Le Besnerais. evo: A realtime embedded stereo odometry for mav applications. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, November 3-8*, pages 2107–2114, 2013.
 13. S. Shen, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained mav. In *2011 IEEE International Conference on Robotics and Automation*, 2011.

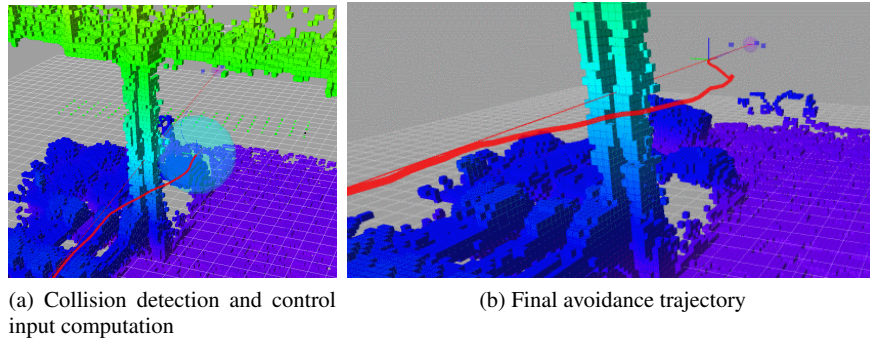
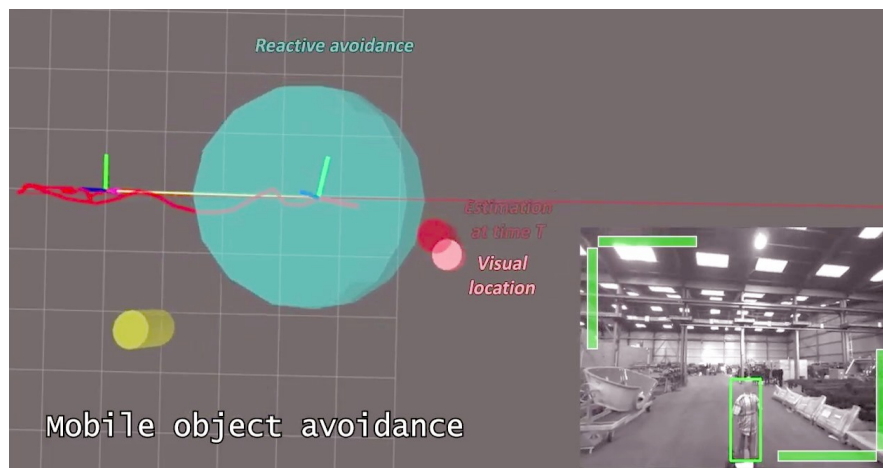


Fig. 10: MPC static obstacle avoidance



(a) Detection, estimation and motion prediction of mobile obstacle



(b) Mobile obstacle avoidance (external view)

Fig. 11: Detection, localization and avoidance of mobile obstacle