Launching a Micro–Scout UAV from a Mobile Robotic Manipulator Arm

Prateek Arora
Robotic Workers Lab
University of Nevada, Reno
prateeka@nevada.unr.edu

Christos Papachristos
Robotic Workers Lab
University of Nevada, Reno
cpapachristos@unr.edu

***Abstract*— This paper addresses the problem of autonomously deploying an unmanned aerial vehicle in non-trivial settings, by leveraging a manipulator arm mounted on a ground robot, acting as a versatile mobile launch platform. As real-world deployment scenarios for micro aerial vehicles such as search-and-rescue operations often entail exploration and navigation of challenging environments including uneven terrain, cluttered spaces, or even constrained openings and passageways, an often arising problem is that of ensuring a safe take-off location, or safely fitting through narrow openings while in flight. By facilitating launching from the manipulator end-effector, a 6-DoF controllable take-off pose within the arm workspace can be achieved, which allows to properly position and orient the aerial vehicle to initialize the autonomous flight portion of a mission. To accomplish this, we propose a sampling-based planner that respects a) the kinematic constraints of the ground robot / manipulator / aerial robot combination, b) the geometry of the environment as autonomously mapped by the ground robots perception systems, and c) accounts for the aerial robot expected dynamic motion during takeoff. The goal of the proposed planner is to ensure autonomous collision-free initialization of an aerial robotic exploration mission, even within a cluttered constrained environment. At the same time, the ground robot with the mounted manipulator can be used to appropriately position the take-off workspace into areas of interest, effectively acting as a carrier launch platform. We experimentally demonstrate this novel robotic capability through a sequence of experiments that encompass a micro aerial vehicle platform carried and launched from a 6-DoF manipulator arm mounted on a four-wheel robot base.**

TABLE OF CONTENTS

1. INTRODUCTION

In recent years, Micro Aerial Vehicles (MAVs) have been widely deployed for various application such as underground mine inspection[1, 2], infrastructure surveying[3, 4] and most importantly for search and rescue operations [5, 6]. Due to agile nature of MAV's and their ability to be rapidly deployed they are perfectly suited for such time sensitive operations and challenging environment [7, 8]. However, launching the MAV from uneven terrain and inside cluttered space and constrained environment might not be possible given the dynamics of the system. This entails determining a safe take-off location than ensures collision free launch of the MAV by taking into account the system's take-off dynamics.

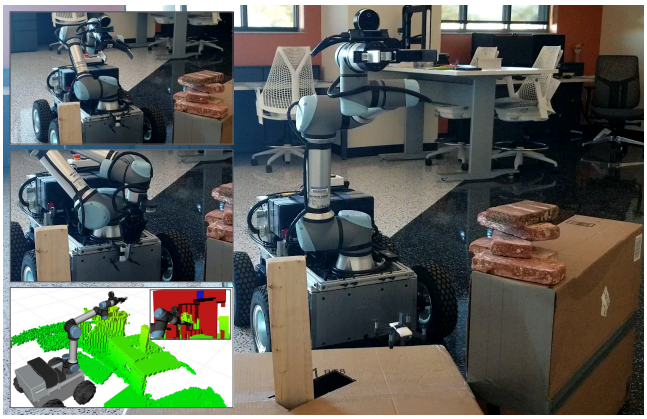

Figure 1. The MAV micro-scout after being launched by the mobile manipulation carrier. Detail: Planning and deploying the aerial robot by picking it up from the mobile base and launching it from a collision-safe takeoff point.

To achieve this, we propose the use of a heterogeneous multi-robot system comprising a manipulator arm equipped with a gripper and a MAV. A 6-Degree-of-Freedom (DoF) launching pose within the arm's workspace can be achieved, which allows to accurately position and orient the aerial vehicle to facilitate safe take-off. Such a heterogeneous multi-robot agent has the ability to collaboratively handle a large variety of tasks by leveraging systems with diversified kinematics and dynamics [9]. Furthermore, a multi-robot system offers us with more control and flexibility for the design of the overall system and individual robots in terms of the size, weight, area, power. For instance, an aerial robot equipped with bulky LiDAR, cameras and onboard computer can be replaced with a MAV with minimal sensors and limited computation power, that can fit through narrow spaces, offloading the computation power to another system such as a mobile manipulation system.

In this paper, we present an approach towards autonomous deployment of unmanned aerial vehicle in constrained, non-trivial environment by leveraging a manipulator arm to reliably position and orient the MAV for safe launch. First, to facilitate evaluation of safe takeoff region in the environment and answering multiple collision-queries for generating collision-free trajectories, the manipulator generates volumetric representation of its surroundings using the onboard sensors. Second, characterization of viewpoint orientations with respect to nominal target pose within the arm workspace of manipulator as free or occupied is performed. Third, to enable safe takeoff, a launch pose is selected by evaluating free

978-1-7281-2734-7/20/$31.00 ©2020 IEEE

2021 IEEE Aerospace Conference | 978-1-7281-7436-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/AERO50100.2021.9438326

viewpoint orientations with forward simulation of closed-loop translational dynamics model of MAV. Lastly, a collision free trajectory is generated for the determined launch pose to be executed by the manipulator. The presented approach is tested in a non-trivial scene which is discussed in section 4.

This paper is organized as follows: Section 2 presents the overview of our Heterogeneous multi-robot system. Section 3 elaborates the proposed approach, followed by Section 4 that details the experimental verification studies. Finally Section 5 concludes the article.

2. System Overview

The mobile manipulation robot alongside the Micro Aerial Vehicle that are used for the development and demonstration of the proposed MAV launching scheme, are overviewed within this Section.

Mobile Manipulation System— The in-house developed *Beaver* system [10] represents an autonomous physical interaction robot, equipped with the required perception, navigation, and planning pipelines that enable fundamental capabilities including multi-modal Simultaneous Localization And Mapping (SLAM) [11–14], volumetric mapping [15], exploration [16–19], guidance [20], and manipulation [21]. Its onboard perception incorporates a custom "eye–in–hand" multi-modal sensor module that enables autonomy in the aforementioned contexts. Figure 2 illustrates the specific embodiment including a breakdown of its main components and subsystems. The robot comprises:

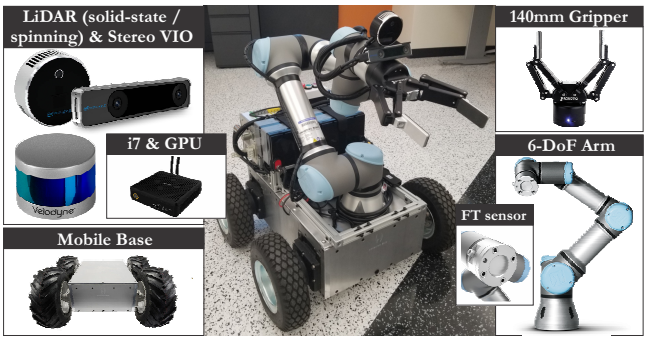

Figure 2. The *Beaver* Mobile Manipulation system.

- A differential 4-wheel drive mobile base with 0.85 $[m]$ wheelbase, $>$ 120 $[kg]$ payload capacity, and 4 wheel encoders providing direct wheel odometry.
- An industrial-grade 6–DoF manipulator arm with 0.85 $[m]$ reach and $\simeq$ 5 $[kg]$ manipulation payload capacity, rigidly mounted onto the mobile base.
- An Force/Torque sensor integrated with the arm, providing force and torque feedback at ranges of up to 50 $[N]$ and 10 $[Nm]$ respectively.
- A 2–$Finger$ gripper with 140 $[mm]$ opening and 125 $[N]$ holding force.
- An Intel $i7$–$9750H$ (6-core) and NVIDIA RTX–2070 GPU based onboard computer with $Wi-Fi$ connectivity.
- An Intel Realsense $L515$ sensor integrating a solid-state 860 $[nm]$ LiDAR sensor and a 6–DoF IMU with 400 $[Hz]$ update rate, as well as an RGB color camera (rolling shutter).
- An Intel Realsense $T265$ sensor providing on-chip Visual Inertial Odometry (VIO) and 2 fisheye-lens (173^o–FoV) monochrome camera image streams.

The software architecture relies on a fully preemptible real-time build of a state-of-the-art Linux kernel. Additionally the framework of the Robot Operating System (ROS) acts as the middleware solution for the required inter-process communication between a collection of modules including device drivers to high-level algorithmic pipelines.

Micro Aerial Vehicle System—The launchable MAV is embodied by a RYZE Tello miniature quadcopter robot measuring a mere 98x92.541 $[mm]$ at a takeoff weight of 0.080 $[kg]$. It offers the following application-relevant features:

- $802.11n$-based communication over a dual-antenna 2.4GHz $Wi-Fi$ link.
- Onboard optical flow-based stabilization control and barometer-based altitude hold.
- Video streaming at $720p$ – 30 $[Hz]$ resolution of an electronically-stabilized forward-facing onboard camera.
- An Application Programming Interface (API) offering support for UDP-based wireless command and data access.

The overall system architecture is visualized in Figure 3.

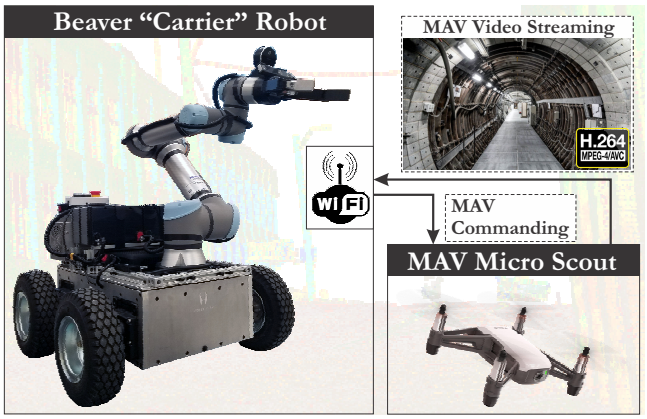

Figure 3. The MAV Mobile Manipulation Launching system architecture.

3. Proposed Method

Considering the previously elaborated system specifications, the following approach is proposed to perform safe and consistent autonomous launching of the MAV from a mobile manipulation system.

Overall Approach

This paper considers a policy that to address the deployment of a MAV in an unstructured environment, on authority of a mobile manipulation robot that acts both as its carrier, while at the same time fulfilling the primary role of autonomous exploration within the context of a relevant mission. Therefore, the corresponding problem statement is formulated as part of a comprehensive pipeline which initiates with partial information about a space of interest which may contain a number of obstructions, and is capable of performing the necessary actions that guarantee the successful deployment of the carried MAV.

To do so, the proposed pipeline leverages a) the carrier robot manipulation flexibility, onboard perception, and volumetric reconstruction capabilities, b) a planning pipeline which respects the dynamics model and relevant constraints of the

MAV, such that its launching is not expected to lead to collisions with the environment, c) a Finite State Machine (FSM) strategy to coordinate the entire process while taking into account system specifics, such as the placement of the aerial robot on the carrier mobile manipulation system, and finally d) a direct robot-to-robot command channel that facilitates the actual launch-deployment in an autonomous fashion.

Nomenclature—Figure 4 illustrates the *Beaver* and the MAV robots alongside the corresponding coordinate system frames of reference utilized within this paper's nomenclature.

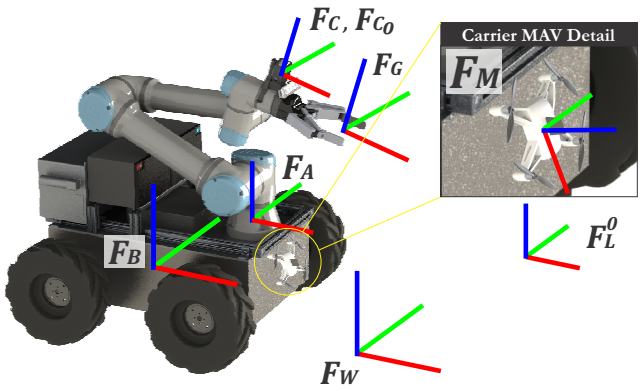

Figure 4. Nomenclature and Reference Frames.

Onwards, we will be using the following notations:

- A reference frame A is denoted as $\mathcal{F}_\mathcal{A}$.
- A translation vector between 2 reference frames $\mathcal{F}_\mathcal{A}$ and $\mathcal{F}_\mathcal{B}$ expressed in the $\mathcal{F}_A$ frame is denoted as ${}_A r_{AB}$.
- A rotation matrix between 2 reference frames $\mathcal{F}_\mathcal{A}$ and $\mathcal{F}_\mathcal{B}$ is denoted as $\mathcal{R}_{AB}$ $(\in SO_3)$.
- As a result, the $\mathcal{T}_{\mathcal{AB}}$ $(\in \mathbb{R}^{\triangle \times \triangle})$ rigid body transformation matrix between the 2 aforementioned reference frames is: $\mathcal{T}_{AB} = \begin{bmatrix} \mathcal{R}_{AB} & {}_A r_{AB} \\ \mathbf{0}^{\times \mathbf{3}} & 1 \end{bmatrix} (\in SE_3)$
- A q_{AB} quaternion denotes another possible representation of the equivalent rotation represented by $\mathcal{R}_{AB}$ (and encapsulated in $\mathcal{T}_{AB}$).

The indicated frames of interest are:

- $\mathcal{F}_W$ represents an inertial frame of reference, i.e. a "flat world" coordinate frame which is arbitrarily chosen to be gravitationally-aligned, and coincident with the virtual footprint of the mobile base at the beginning of an experiment (the centroid of all four wheels' ground contact points).
- $\mathcal{F}_C$ represents the optical frame of the L515 solid state LiDAR, which is utilized for environment volumetric reconstruction.
- $\mathcal{F}_C^0$ signifies a nominal "resting" pose for the manipulator LiDAR frame, which corresponds to the arm configuration in the beginning of an experimental sequence.
- $\mathcal{F}_G$ represents the gripper grasping virtual end-effector configuration, i.e. the centroid of the closed finger pads, and is aligned per the fingers' axes.
- $\mathcal{F}_A$ represents the arm base, i.e. the frame of reference for calculation of the joint forward kinematic solution to derive the $\mathcal{T}_{\mathcal{GA}}$ transformation.
- $\mathcal{F}_M$ represents the MAV rigid body frame. It is highlighted that this is not fixed to the end-effector, but becomes rigidly associated to it once picked up by the gripper, and left free-floating once released.

Finally, $\mathcal{F}_L^0$ signifies a "target launching" pose which is considered as the centroid of a nominal takeoff region for the MAV. It is noted that the proposed policy leverages this a root to initiate the search that will lead to the eventual MAV deployment, thus being able to bias the derived takeoff pose around a desired region of interest.

Mobile Manipulation-based Volumetric Mapping

In order for the safe takeoff of the MAV to be achieved autonomously, the mobile manipulation system has to be capable of reconstructing its surroundings to a level of detail that enables the subsequent evaluation of safe takeoff trajectories. At the same time, the employed underlying representation should facilitate the computationally efficient evaluation of multiple collision-queries such that kinematic planning for the manipulator arm yields feasible collision-free trajectories that lead to a configuration that achieves $\mathcal{F}_\mathcal{M} \simeq \mathcal{F}'_\mathcal{L}$, i.e. positions the aircraft at a viable launch pose around the nominal region.

To these purposes, Octomap [15] is used as the mapping backend, and MoveIt [21] is leveraged as the core motion planning framework. Assuming a volumetric representation of the robot's operating space $\mathcal{V}$, derived by splitting it into cube-shaped sub-regions (voxels) of a minimum edge-length of $m = 50\ [mm]$ such that it remains to scale with the MAV robot mechanical specifications (98x92.541 $[mm]$ body and $76mm$ propeller disc size), we consider that it can be distinguished into: a) a subset comprising the occupied mapped space $\mathcal{V}_{occ}$, b) the free mapped space $\mathcal{V}_{free}$, and c) the unknown space $\mathcal{V}_{unknown} \equiv \mathcal{V} \setminus \mathcal{V}_{known}$, where $\mathcal{V}_{known} \equiv \mathcal{V}_{occ} \cup \mathcal{V}_{free}$.

In order to facilitate the UAV launch planning functionality, volumetric mapping takes place through grid-based planning to derive viewpoints for the L515 LiDAR sensor which offer observations of the "target launching" pose from different orientations. More specifically, considering an azimuth-elevation orientation vector search space $\{Az,\ El\}$ w.r.t. the nominal target pose $\mathcal{F}_L^0$, the goal is to evaluate all reachable viewpoint orientations in order to comprehensively map the structure of the launch site and characterize this part of the arm workspace as either belonging to $\mathcal{V}_{occ}$ or $\mathcal{V}_{free}$.

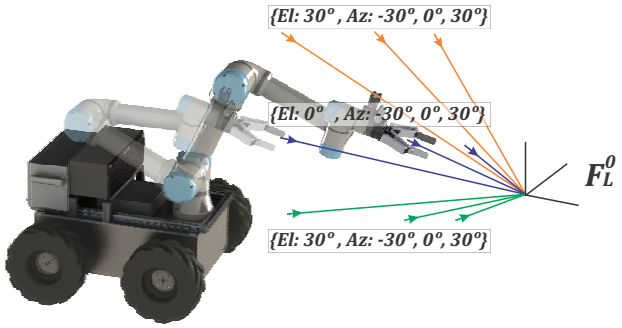

Figure 5. Planning for MAV launch site volumetric mapping.

Figure 5 intuitively illustrates the proposed approach. For each i-th viewpoint vector determined by the elements of $\{Az,\ El\}$, we first compute the minimum-distance viewpoint v_i w.r.t. the "resting" $\mathcal{F}_C^0$ pose that also lies on the vector line, and attempt to find a kinematically feasible plan to reach it. In case of failure, we iteratively evaluate a discrete set of alternative candidates on the same vector line by shifting

the viewpoint towards-and-away from $\mathcal{F}_L^0$, by a step size determined by the mapping backend voxel size m. This search terminates as soon as a valid viewpoint is recovered, or when the distance exceeds the manipulator arm workspace.

Base on the assembled list $\mathcal{V}$ of kinematically reachable viewpoints, we subsequently proceed to compute and execute the manipulator arm trajectory that results in the desired sequence of observations that drive the mapping backend characterization of the MAV launch region.

Planning for MAV Launching

The derived volumetric model is subsequently leveraged in a planning layer that aims to safely position the MAV in an initializing pose as close as possible to the nominal launch location $\mathcal{F}_L^0$, while at the same time guaranteeing that takeoff will be safe and collision-free. The particular challenge involved in this objective is the fact that an aerial vehicle's dynamics affect its feasible takeoff process and should therefore be accounted for in the collision-free planning.

The specific aerial robot embodiment determines both the dynamics model to be considered, as well as the launch procedure. For simplicity, the proposed scheme consider a common closed-loop (attitude and altitude-stabilized) dynamic response that common ground in relevant Frequency Domain-identified models found in literature [22–24] for this MAV class type:

$$\begin{aligned} \dot{T} &= -a_T\, T + \tfrac{1}{a_T}\, T^{ref} \\ \ddot{z} &= T - mg \quad , \quad \dot{z} = \int \ddot{z} + \ddot{z}_0 \quad , \quad z = \int \dot{z} + \dot{z}_0 \end{aligned} \tag{1}$$

, i.e. a first-order model for the aerial robot thrust driving double-integrator dynamics.

Algorithm 1: MAV Launch Planning

Data: i) Environment Volumetric Model sections $\mathcal{V}_{occ}$, $\mathcal{V}_{free}$, $\mathcal{V}_{unknown}$
ii) Arm Kinematic Model (DH) $\mathcal{T}_{i-1,i}\ \ i : A \rightarrow G$
iii) MAV Grip Tranform $\mathcal{T}_{\mathcal{GM}}$
iv) MAV Dynamics Model (1)
Result: i) A kinematically reachable and launch-safe pose $\mathcal{F}_M^{ref}$
ii) Corresponding joint trajectory $\xi^{\mathbf{ref}}$ for manipulator
initialize search pose $\mathcal{F}_M^{ref}$ with nominal launch $\mathcal{F}_L^0$
do
  if $\mathcal{F}_M^{ref} \in \mathcal{V}_{free}$ **then**
    $\xi^{\mathbf{MAV}} \leftarrow$ `Forward_Sim`((1), init: $\mathcal{F}_M^{ref}$)
    if $\xi^{\mathbf{MAV}}$ is collision free **then**
      $\xi^{\mathbf{ref}} \leftarrow$ `IK_Plan`(goal: $\mathcal{F}_M^{ref}$)
      if $\xi^{\mathbf{ref}}$ is collision free **then**
        break
      end
    end
  end
  `Shift`($\mathcal{F}_M^{ref}$)
while `exists`($\mathcal{F}_M^{ref} \in$ manipulator workspace)

It is noted here that the intuitively understood notion of takeoff does not necessarily refer to a vertical climb; particularly for the RYZE Tello MAV considered in this work, the robot can be launched by releasing it –letting it drop or throwing it– and therefore the "takeoff" motion will consist of a vertical dip, until the robot stabilizes in hovering flight.

As previously elaborated, the manipulator arm planning has to ensure that initial positioning of the MAV will ensure a collision-free takeoff; to this purpose, the Algorithm 1 is employed. The core enabling components of the aforementioned algorithm are followingly outlined:

- `Forward_Sim`: Perform forward simulation of the closed-loop translational dynamics model of (1) given the initial conditions of the candidate $\mathcal{F}_M^{ref}$, to derive the expected trajectory of the MAV after launching and valuate whether it is expected to lead to a collision with the environment
- `IK_Plan`: Perform kinematic motion planning to find a joint trajectory for the manipulator arm that is collision-free w.r.t. the manipulator/gripper links and the attached body of the MAV (held by the gripper) by leveraging the previously derived volumetric model of the environment.
- `Shift`: Iteratively shift the considered launch pose by a discrete step size determined by the mapping backend voxel size m, to exhaustively evaluate all possible launch poses around the nominal launch pose $\mathcal{F}_L^0$.

The algorithm terminates as soon as the first kinematically reachable and launch-collision-free pose is recovered, returning the corresponding joint trajectory $\xi^{\mathbf{ref}}$ to be executed by the manipulator. The algorithm will continue exhaustively evaluating poses around the originally provided $\mathcal{F}_L^0$ until there exists no possible shifting of $\mathcal{F}_M^{ref}$ that will yield a pose within the manipulator workspace.

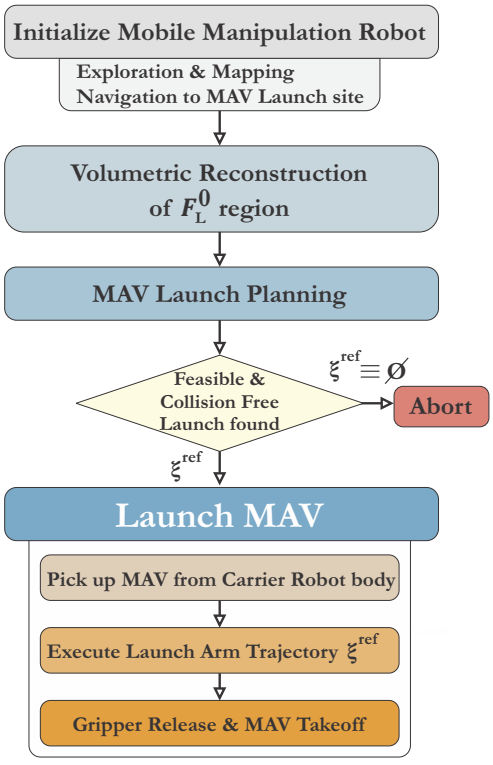

Figure 6. MAV Launching operational Flowchart.

It should be highlighted that the pruning of candidate launch poses based on the aerial robot launch dynamics takes place

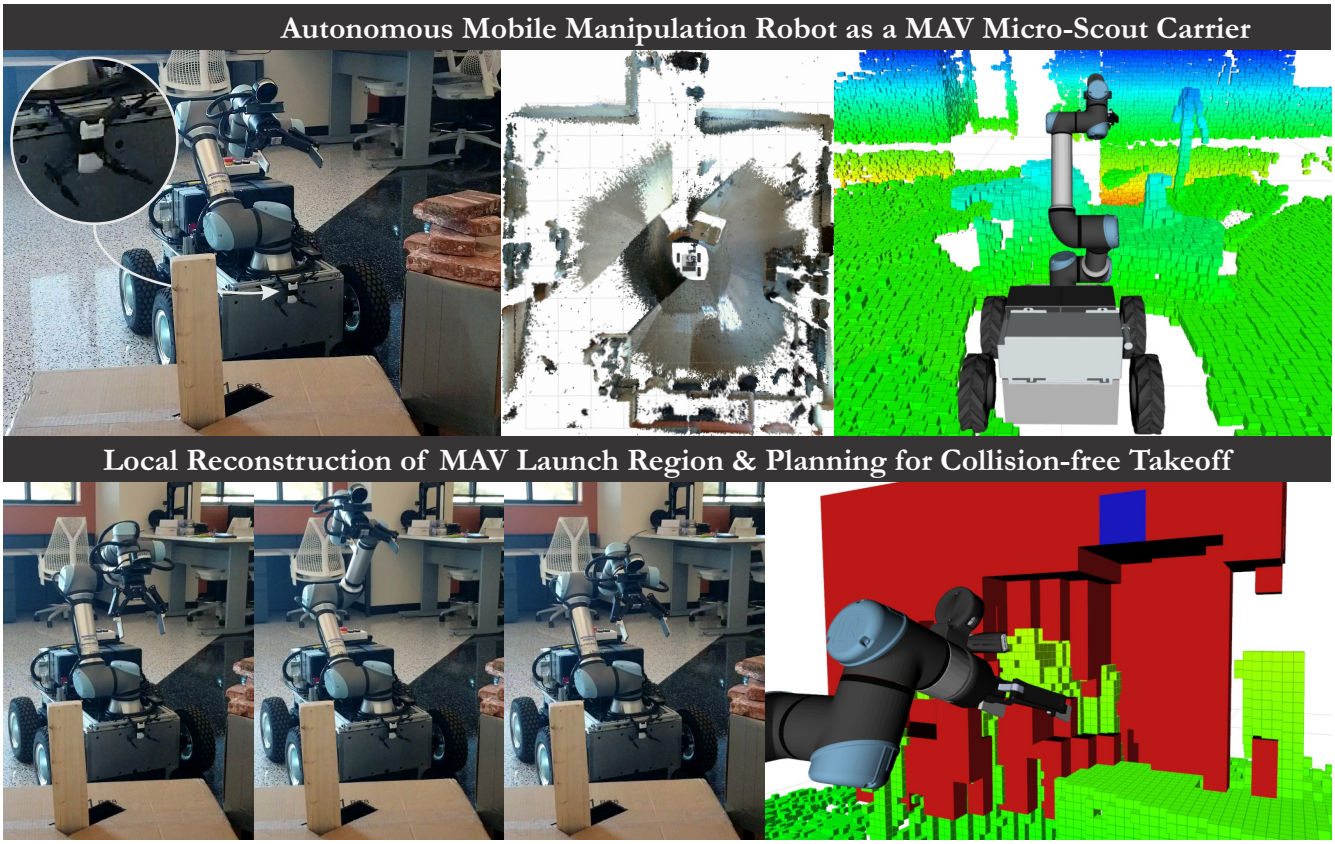

Figure 7. Top row: The mockup deployment environment of the Mobile Manipulation system and MAV micro-scout. Detail: Mounting position of aerial robot the onto the carrier mobile base. Bottom row: Local inspection for fine volumetric reconstruction of the launch region, and annotation of feasible workspace boundary (red-colored) and valid launch poses (blue-colored) that guarantee collision-free takeoff of the MAV.

before their reachability evaluation based on the manipulator motion planning pipeline, as the latter incurs significantly increased computational cost as compared to the limited number of collision checks for a small-bounding box volume that corresponds to the MAV body. As a last remark, in this process the $\mathcal{F}_C$ reference frame is not involved, which implies that there is no perception-related planning at this stage and it is entirely driven by the previously derived volumetric model of the environment.

MAV Launch Procedure

To coordinate the entire launch sequence an appropriately composed flowchart-based action sequencing is designed, encompassing the operational specifications of the robotic systems involved, and leveraging the previously elaborated algorithmic components. Figure 6 intuitively visualizes the process.

It is noted that the initial positioning of the mobile manipulator system is the product of collision-free navigation for the mobile base, which implies that some local volumetric map of the region of deployment has already been reconstructed by the *Beaver* robot. However, this is generally not applicable for the fine-grained planning required to perform MAV launching; therefore the first step of the proposed flowchart is the execution of the fine-detail volumetric reconstruction of desired launch region as described in Section 3. Subsequently, the launch planning process of Section 3 follows, resulting in either a kinematically reachable collision-free joint trajectory to execute, which will position the MAV at a safe-to-takeoff pose close to the originally desired nominal $\mathcal{F}_L^0$, or abort on failing to find one that satisfies these constraints. It is highlighted that the kinematic chain for planning accounts for the additional link of the MAV body, which will be picked up at the time of execution of the launch process.

Once the the plan is obtained, the mobile manipulation robot picks up the MAV which is positioned on the base that also acts as a carrier. It is noted that this pickup trajectory is feedforward-executed as it the positioning of the MAV is known and fixed; moreover, it lies fully within the mobile robot's own planning envelope for collision-free navigation, which means that since the *Beaver* robot arrives at a certain location, execution of the MAV pickup trajectory is known to be feasible.

After the execution of the launch plan $\xi^{\mathbf{ref}}$, a takeoff command is given to the MAV, with an appropriately timed open command to the gripper to release it. The remaining mission which relates to the deployment of the aerial robot follows subsequently.

4. Experimental Evaluation

To verify the effectiveness of the proposed aerial robotic launching from a mobile manipulation system, we conducted a sequence of related experimental studies. More specifically,

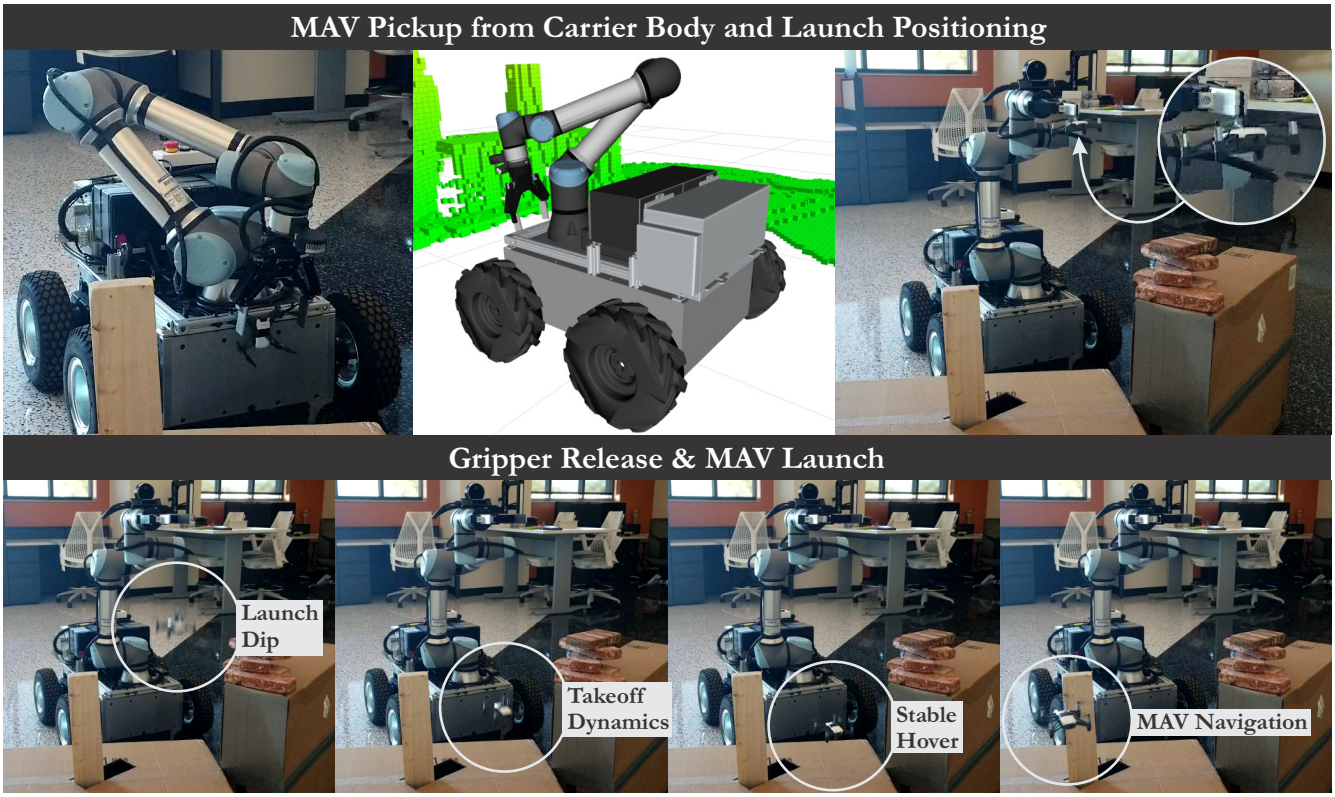

Figure 8. Top row: Picking up the MAV from the carrier mobile base body, and executing the planned motion to position it into the takeoff-safe launch configuration. Bottom row: The MAV launch sequence; the aerial robot is released and drop until its dynamic response brings it into stable hovering, safely clear of environment obstacles around and under it. After launch success, the MAV is commanded to perform scouting navigation, streaming video footage back the the *Beaver* carrier robot.

the *Beaver* Mobile Manipulation system acting as the carrier of the MAV is initialized by moving into a reference position inside the operating environment. As noted in Figure 8 the specific environment comprises a mockup demonstration within a lab space populated with an unstructured obstruction built with heavy boxes, loose bricks, and other structural entities. It corresponds to an impassable entry barrier for the ground robot, but at the same time a perfectly navigable environment for the micro-sized aerial robot.

As highlighted in the Figure, the MAV is attached onto the ground mobile base, ready to be deployed as soon as an appropriate launch location is evaluated; the first step towards this evaluation is the local volumetric reconstruction according to the process elaborated in Section 3, which is visualized in the bottom row of Figure 7. The nominal launch pose $\mathcal{F}_L^0$ drives this inspection process. It is noted that although the evaluated lab space is a controlled testing space, and therefore trivial solutions can be found for the MAV deployment, we intentionally define a nominal launch region which is within the subset of the environment that contains more challenging physical obstructions. This is an attempt to both emulate a more challenging launch requirement (such as having to "stick" the MAV into a narrow pocket), and at the same time verify the planner component that evaluates the aerial robot launch dynamics against collisions with an unstructured environment.

The last frame of Figure 7 illustrates a sample the breakdown of the region around the nominal launch pose; the red parts correspond to unreachable points for the manipulator arm due to maximum workspace, within the clear region encapsulated by this red-colored boundary, the arm may position the MAV there, but given the particular launch process of the Tello quadcopter (expects to be dropped and during free-fall spins up its motors to transition to a hover) only the blue-colored points are viable initial configurations such that the MAV does not crash during takeoff.

Subsequently, Figure 8 illustrates the actual MAV launching procedure, according to the description of Section 3. The carrier robot reaches to grasp and detach the MAV from its body, and executes the previously planned feasible kinematic trajectory ξ^{ref} that positions the MAV into a takeoff collision-free launch pose. The bottom row comprises the final launch sequence; the aerial robot is launched by releasing its hold, and based on the evolution of its takeoff dynamics stabilizes around a hover position which is in safe clearance of the underlying structures due to the presence of the aforementioned obstructions. Once the safe launch of the MAV is complete, the wireless communication protocol offered by the Tello SDK can be leveraged to command the aerial robot navigation, while at the same time streaming onboard stabilized video footage back to the carrier *Beaver* Mobile Manipulation system.

5. Conclusions

This work presented a pipeline for the mobile manipulation-based deployment of a MAV micro-scout robot within an unstructured environment. The proposed policy respects

the environment constrains, while also accounting for the expected dynamical evolution of the aerial robot takeoff motion. A concrete embodiment of the envisaged multi-robot system was also elaborated, and a conclusive experimental evaluation was presented, indicating this strategy's overall effectiveness.

ACKNOWLEDGMENTS

This material is based upon work supported by Governors Office of Economic Development of the State of Nevada under the Construction Robotics Award. The presented content and ideas are solely those of the authors.

REFERENCES

[1] C. Kanellakis, S. S. Mansouri, G. Georgoulas, and G. Nikolakopoulos, "Towards autonomous surveying of underground mine using mavs," pp. 173–180, 2018.

[2] P. Gohl, M. Burri, S. Omari, J. Rehder, J. Nikolic, M. Achtelik, and R. Siegwart, "Towards autonomous mine inspection," pp. 1–6, 2014.

[3] N. Metni and T. Hamel, "A uav for bridge inspection: Visual servoing control law with orientation limits," *Automation in construction*, vol. 17, no. 1, pp. 3–10, 2007.

[4] T. Özaslan, G. Loianno, J. Keller, C. J. Taylor, V. Kumar, J. M. Wozencraft, and T. Hood, "Autonomous navigation and mapping for inspection of penstocks and tunnels with mavs," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1740–1747, 2017.

[5] N. Michael, S. Shen, K. Mohta, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno *et al.*, "Collaborative mapping of an earthquake damaged building via ground and aerial robots," pp. 33–47, 2014.

[6] D. Thakur, G. Loianno, W. Liu, and V. Kumar, "Nuclear environments inspection with micro aerial vehicles: Algorithms and experiments," pp. 191–200, 2018.

[7] S. Waharte and N. Trigoni, "Supporting search and rescue operations with uavs," pp. 142–147, 2010.

[8] S. Waharte, A. Symington, and N. Trigoni, "Probabilistic search with agile uavs," pp. 2840–2845, 2010.

[9] P. Chand and D. A. Carnegie, "Mapping and exploration in a hierarchical heterogeneous multi-robot system using limited capability robots," *Robotics and autonomous Systems*, vol. 61, no. 6, pp. 565–579, 2013.

[10] P. Arora and C. Papachristos, "Mobile manipulator robot visual servoing and guidance for dynamic target grasping," in *International Symposium on Visual Computing*. Springer, Cham, 2020, pp. 223–235.

[11] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[12] F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "A multi-modal mapping unit for autonomous exploration and mapping of underground tunnels," in *2018 IEEE Aerospace Conference*. IEEE, 2018, pp. 1–7.

[13] S. Khattak, C. Papachristos, and K. Alexis, "Keyframe-based direct thermal-inertial odometry," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3563–3569.

[14] ——, "Keyframe-based thermal-inertial odometry," *Journal of Field Robotics*, vol. 37, no. 4, pp. 552–579, 2020.

[15] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[16] T. Dang, F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, "Graph-based path planning for autonomous robotic exploration in subterranean environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3105–3112.

[17] C. Papachristos, F. Mascarich, S. Khattak, T. Dang, and K. Alexis, "Localization uncertainty-aware autonomous exploration and mapping with aerial robots using receding horizon path-planning," *Autonomous Robots*, vol. 43, no. 8, pp. 2131–2161, 2019.

[18] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty-aware receding horizon exploration and mapping using aerial robots," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 4568–4575.

[19] C. Papachristos, M. Kamel, M. Popović, S. Khattak, A. Bircher, H. Oleynikova, T. Dang, F. Mascarich, K. Alexis, and R. Siegwart, "Autonomous exploration and inspection path planning for aerial robots using the robot operating system," in *Robot Operating System (ROS)*. Springer, 2019, pp. 67–111.

[20] C. Rsmann, F. Hoffmann, and T. Bertram, "Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control," in *2015 European Control Conference (ECC)*, 2015, pp. 3352–3357.

[21] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 18–19, 2012.

[22] C. Papachristos, T. Dang, S. Khattak, F. Mascarich, N. Khedekar, K. Alexis *et al.*, "Modeling, control, state estimation and path planning methods for autonomous multirotor aerial robots," *Foundations and Trends® in Robotics*, vol. 7, no. 3, pp. 180–250, 2018.

[23] Mark B. Tischler, Robert K. Remple, *Aircraft and Rotorcraft System Identification: Engineering methods with Flight-Test examples*. American Institute of Aeronautics and Astronautics (AIAA).

[24] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*. Wiley Interscience, 1992.

BIOGRAPHY

Prateek Arora *is a PhD student at the Robotic Workers (RoboWork) Lab at the University of Nevada, Reno (UNR). Prior to joining UNR, he graduated with M.Eng Robotics degree from the University of Maryland in association with Perception and Robotics Group. He received his B.S. degree from Indraprastha University, India. His research interests are in the areas of aerial robotics, control, and motion and path planning for autonomous systems.*

Christos Papachristos *is an Assistant Professor at the Computer Science & Engineering Department of the University of Nevada, Reno and the director of the Robotic Workers (RoboWork) Lab. Previously, he held the position of Research Assistant Professor at the University of Nevada, Reno in affiliation with the Autonomous Robots Lab, where he focused his robotics research on enabling operational resilience and autonomous exploration, consistent high-fidelity mapping, and characterization of complex GPS-denied degraded visual environments including underground mines and tunnels. Dr. Papachristos obtained his Ph.D. in aerial robotics at the University of Patras in Greece. He is the author of more than 50 publications, and has acted as co-PI or Senior Personnel in multi-million projects in the US and the EU.*