

Drivocopter: A concept Hybrid Aerial/Ground vehicle for long-endurance mobility

Arash Kalantari¹, Thomas Touma¹, Leon Kim¹, Rianna Jitosho¹, Kyle Strickland¹,
Brett T. Lopez¹, and Ali-akbar Agha-mohammadi¹
Jet Propulsion Laboratory, California Institute of Technology

Abstract—Aerial robots show promise for increased capabilities in exploring unstructured and challenging environments. However, they are limited by payload capacity, flight times, and susceptibility to damage in case of collision. On the other hand, ground robots are able to carry larger payloads and have a lower cost of transport, at the price of limited mobility over challenging terrain. This paper presents a hybrid aerial/ground vehicle that combines the capabilities of both types of vehicles to enable multi-modal mobility in diverse and challenging environments, a lower cost of transport compared to purely aerial vehicles, increased payload capacity, and a design that is more robust to collisions and physical interaction within potentially cluttered and narrow spaces. The design consists of a UAV with four independently actuated spherical wheels which, in addition to providing traction for ground mobility, protect the propellers in collision. In comparison to hybrid vehicles with passive wheels presented in other designs, actuated wheels mitigates perception degradation in dusty environments caused by downwash from thrusting close to the ground. In addition, the integration of an end-to-end autonomy stack is presented which enables the control, planning, and autonomous navigation of the hybrid vehicle in unknown environments. The controls framework employs a geometric tracking controller for aerial trajectories and a cascaded position and velocity controller for ground mobility. We leverage motion primitives to locally plan collision-free paths and a differential flatness mapping to generate kinodynamically feasible trajectories for both terrestrial and aerial modalities in a unified manner. Lastly, we utilize a grid based A* search and probabilistic 3D mapping based on octrees to plan geometric aerial/ground paths to a goal. With this framework, we hope to demonstrate the capabilities of this hybrid aerial/ground vehicle in challenging unknown environments and improved energy efficiency for hybrid mobility over purely flying.

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. SYSTEM OVERVIEW	2
3. SYSTEM AUTONOMY	5
4. CONCLUSIONS	8
REFERENCES	9

1. INTRODUCTION

Mobile autonomy in unstructured environments presents a number of unique challenges for robot platform design. A vehicle must be robust to collisions and debris, maintain a high level of maneuverability, and balance payload capacity with limited battery life. While aerial vehicles are highly maneuverable and able to navigate through cluttered and narrow spaces, ground vehicles excel in their robustness and energy efficiency.



Figure 1. Rendering of the hybrid vehicle consisting of a UAV with four independently actuated spherical wheels.

In this study, we draw from the strengths of both modes of transport by implementing a hybrid design. A rendering of our design is illustrated in Fig. 1. The vehicle is capable of bi-modal mobility which leads to higher energy efficiency since the vehicle can operate in one of two modes depending on which consumes less energy. In the case where the terrain is ideal for wheeled locomotion, the vehicle can operate as a traditional ground vehicle. If large obstacles are present or the terrain is not traversable, then the vehicle can leverage its flight capabilities to continue its mission.

Hybrid aerial-ground vehicles have previously been explored as alternative mobility method in unstructured environments. Because such environments may include cluttered or narrow spaces, these vehicles typically implement vertical take-off and landing (VTOL). Furthermore, prior work largely falls into two categories depending on whether the ground mode of transport is active or passive[1] [10] [11] [12]. A summary of prior work in hybrid aerial-ground vehicles is outlined in Fig. 2 [13] [14].

Novel Contribution

Our work differs from prior work in hybrid aerial-ground vehicles in that we draw from the strengths of both categories of hybrids. By implementing active ground mobility, the vehicle has reduced dust generation in ground mode as well as improved robustness from the decoupling of modes. Our work improves upon existing active wheels design by utilizing four spherical cages which surround the propellers. This gives a similar protection provided by a passive rotating cage design without negatively impacting the vision and perception systems, which is associated with a vehicles using

¹The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

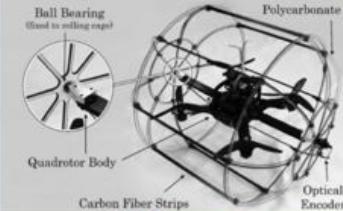
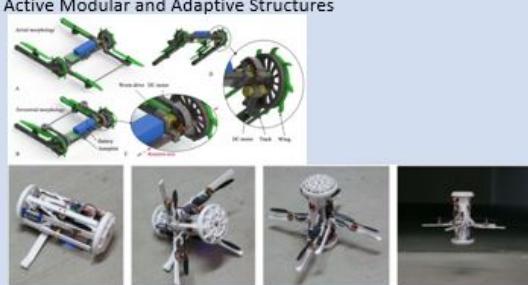
		Mode Descriptions	Mechanical Design	Control System	Reliability/Robustness
Passive Wheels		Fly over obstacles as the mechanism for aerial and ground mode is coupled.	Simplified design due to coupled modes.	Same control system structure for both modes. Quick/simple transition between modes.	Sensors not protected from debris. Dust generation in ground mode may cause drift or failure of state estimation. If failure occurs, both modes fail.
Passive Rotating Cage		Fly over obstacles as the mechanism for aerial and ground mode is coupled.	Simplified design due to coupled modes.	Same control system structure for both modes. Quick/simple transition between modes.	Whole system protected from debris. Dust generation in ground mode may cause drift or failure of state estimation. If failure occurs, both modes fail. Vision/perception difficult due to cage.
Active Wheels		Roll over uneven terrain. Fly when absolutely necessary. Reduced flight time due to weight. Increased rolling time due to efficiency of wheels ground mode.	Added weight from actuators for driven wheels.	Controls systems for each mode can be independent and decoupled for optimization. Delay during transition between modes.	Sensors not protected from debris. No additional dust generation in ground mode. Decoupled systems increase robustness to failure in a single mode.
Active Legged UAVs		Able to climb over higher obstacles. Slow motion in ground mode. Reduced flight time due to weight.	Added weight and complexity from legs.	Controls systems for each mode can be independent and decoupled for optimization. Motion plan more complex for ground mode. Delay during transition between modes.	Sensors not protected from debris. No additional dust generation in ground mode. Decoupled systems increase robustness to failure in a single mode.
Active Modular and Adaptive Structures		Additional degrees of freedom. Needs flat surfaces for transformation between modes.	Added weight and complexity from morphing mechanisms. Increased capabilities (ex. Additional modes).	Complex control system for modes and transitioning between modes. Delay during transition between modes.	Compromised reliability and robustness due to system complexity. No additional dust generation in ground mode.

Figure 2. Summary of Vertical Take Off Landing Hybrid Aerial-Ground Vehicles [1] [2] [3] [4] [5] [6] [7] [8] [9]

a single rotating cage.

In the following sections, we provide a detailed discussion of our hybrid aerial-ground vehicle. We first outline the system hardware that enables decoupled modes of transport. We then describe the on-board electronics required for autonomy, and the controllers enabling the ground and aerial modes. We discuss how these design choices enable a light-weight, robust vehicle with high efficiency and endurance. Finally, we present an end-to-end autonomy stack that handles local and global planning in unknown environments.

2. SYSTEM OVERVIEW

Mechanical Design

The main contribution of the presented mobility system is in developing independently actuated aerial and terrestrial locomotion systems, which makes design of both systems very challenging. These challenges and design choices are discussed below.

System Requirements—The presented platform was initially developed to tackle the DARPA Subterranean (SubT) Challenge where the robot should autonomously navigate, map, and locate artifacts in subterranean environments including man made tunnels, subway stations, and caves. Operation in such environments imposes crash resistance as a certain

Table 1. List of vehicle components

Subsystem	Type	Hardware
Ground Mobility	Actuator	4x Maxon DCX 16 L
	Controller	2x Pololu dual controller
	Wheel	12" dia., custom made
Aerial Mobility	Actuator	8x T-motor MN3110
	ESC	2x T-motor F45A
	Propeller	8x 9.5"x4.5"
	Flight controller	Pixhawk 2.1
Computing		Intel NUC
Sensors	3D Lidar	Velodyne Puck Lite
	1D Lidar	Garmin
	Camera	Intel Realsense D435i
Battery		10Ah, 6S

requirement for the platform. The challenge rules also requires the platform to be able to pass through a 1m x 1m window, which limits the dimensions of the platform. In order to enable the platform to autonomously navigate the subterranean environment and solve the challenge of localization and mapping, the system should be carry a Velodyne Puck Lite, an Intel Realsense D435i, and a Garmin 1D lidar, a compact processing unit (Intel NUC), and all the related cabling. This requires the platform to have a minimum sensor payload of about 1.2kg. Additionally, a minimum hover time of 10 minutes was chosen to enable the system to fly longer distance over the challenging train where driving is not possible.

Aerial Mobility—In order to design the propulsion system, a total mass of 4.5kg was assumed for the platform. This consisted of 2kg mass of airframe and ground locomotion, 1.2kg mass of sensors, and 1.3kg mass of the battery. Based on these assumptions, the 10 minutes required hover time, and the need for the system to be compact, we chose to use a coaxial quadrotor configuration for aerial mobility. Eight MN3110 T-motor brushless DC motors (KV700), equipped with 9.5"x4.5" propellers generate the thrust. Using a 6S, 10Ah lithium polymer battery pack, the platform is expected to be able to hover for at least 10 minutes.

Ground Mobility—Four driven wheels enable the ground locomotion for the platform. The platform wheels, or the Actuated Propeller Cages (APC), which also serve as propulsion system guard in case of crashes (Fig. 3), are one of the platform's primary and unique characteristic part of the design. The targeted environments contain a variation of surfaces from concrete stairs in a subway to stalagmites in a cave. The vehicle will be traversing this environment by both rolling on the surface and flying through it. Therefore the APC must retain acceptable rigidity when flying and landing to prevent the propellers from clashing with the cage in case of collision with an obstacle or the ground. The APC also has to be elastic enough to flex when absorbing shock forces so as to dissipate force transfer through the Driving Rigid Airframe (DRA) when landing. A further requirement is for the APC to retain an oblate spherical shape when ground traversing for traction hence a balance of rigidity and lightweight properties had to be achieved. The design is based on cutting a flat sheet of 2mm thick Carbon Fiber, which is then bent into a sphere and retained in shape using 3D printed nylon hubs. Maxon DCX16L (18V, 16 watts) brushed DC motors with a 243:1 gearset was chosen to drive the wheels. The motor power is transmitted to the wheels using GT3 2mm pitch timing belts with proper gear ratio to provide enough torque while making it possible for the platform to drive at a maximum speed of 1m/s.



Figure 3. The Actuated Propeller Cage containing the propulsion system and demonstrating connection to the Driving Rigid Airframe.

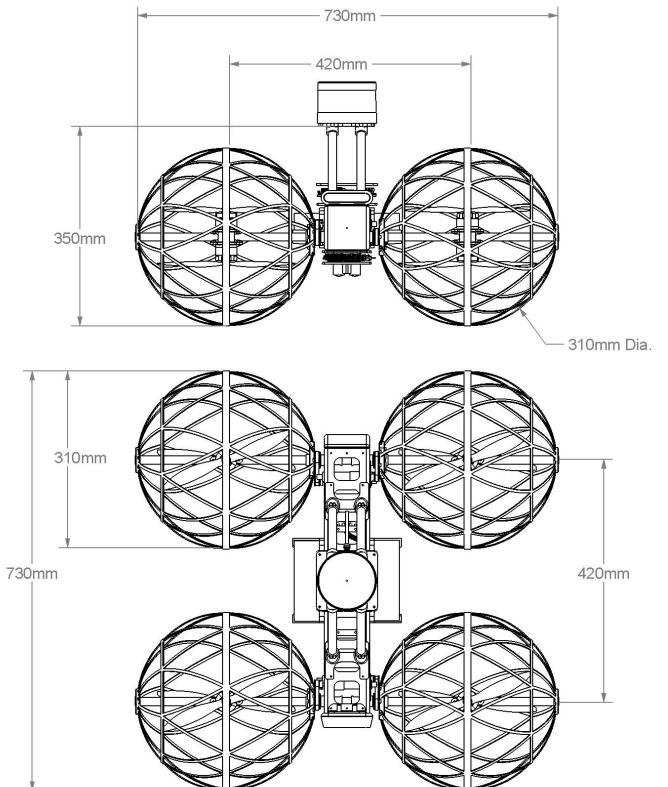


Figure 4. Overall dimensions of the platform.

Electronics and Control

Aerial Mobility—To control aerial mobility, the system utilizes a Pixhawk unit that controls the eight brush-less motors. Speed control is done via two electronic speed controllers, capable of controlling four of the motors each, which receive pulse width modulated signals from the Pixhawk. This system, used for aerial mobility only, interfaces to the NUC via a single USB connection. It is ideal for this system to only be active as a last resort, when terrain is impassable by any other means, as the power consumption is extensive.

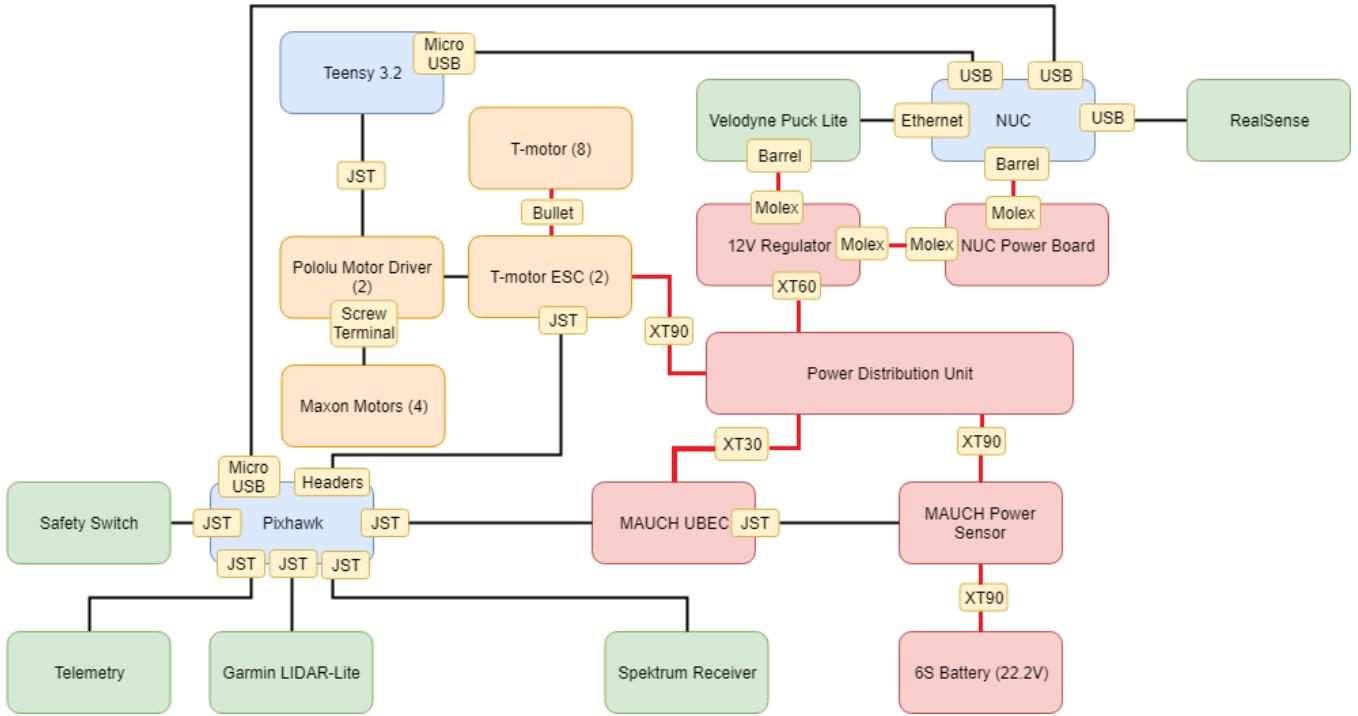


Figure 5. Overview of the electronics

Ground Mobility—For simplicity and reliability, the ground propulsion system is driven by the Teensy 3.2, a 32-bit microcontroller. This microcontroller commands the four motor drivers via high frequency pulse width modulation (PWM). Each driver receives two PWM signals and calculates the difference between the two to determine the motor direction and speed. This allows for very fine control of the motor speeds for each wheel. The four ground propulsion motors on the vehicle are tracked via rotary encoders, which the microcontroller reads and publishes via a ROS node to the intel NUC over USB. Thus, the microcontroller handles the simple and iterative operations of driving motors and tracking rates of rotation, and communicating this data to the Intel NUC to enable closed loop control of the motor velocities. The communications system between the embedded motor control system and the Intel NUC is outlined in the next section.

Embedded Communications—The vehicle features three main, separate processing systems; the Intel NUC, which executes the far more computationally expensive tasks like perception and navigation, the Pixhawk, which has control authority over the flight system, and the Teensy 3.2, which tracks and commands the ground propulsion system. All these systems interface over ROS network; The pixhawk and the NUC communicate over a USB connection via MAVROS, and the Teensy 3.2 and the NUC communicate over a USB connection via the open source Arduino ROS Serial, which has been slightly modified to support the ARM architecture Teensy 3.2. As previously mentioned, the Teensy 3.2 does not conduct any actual decision making; rather it reports a vector over the ROS Serial connection, containing a rate of rotation for each wheel. The NUC processes this data, and in response sends a vector to the Teensy containing wheel velocities for the ground propulsion system to act upon.

The Teensy 3.2 was chosen to read and control wheel velocity over other available microcontrollers, due to its

high clock speeds and correspondingly higher PWM timer frequencies, as well as its native USB capabilities, number of PWM channels, and number of interrupt capable pins. Such a long list of available resources in such a compact design was critical, as weight and power consumption were critical factors. Similar microcontroller boards that could support the number of motor control signals, quadrature inputs from encoders, and provide an equally reliable native USB port were typically much larger.

Performance Assessment

The vehicle was tested in NASA’s Jet Propulsion Laboratory Mars Yard. The system was able to drive over dynamic environmental features such as rocks and granite. The APC successfully demonstrated shock transfer and safe elastic deformation through its structure when overcoming obstacles. The APC subsequently returned to its natural shape after this deformation, maintained its structural integrity and the vehicle continued driving efficiently and in a stable manner. The system consumes about 130 watts of power for driving on a flat ground. During flight, the power consumption rate is about 8 times higher (1060 watts). It is estimated that the system can drive on ground for about 90 minutes and hover in air about 8 minutes with a fully charged 10Ah, 6S LiPo battery.

The mass of the vehicle and its various divisions can be observed in Fig. 7. The total gross weight of the vehicle is 5kg. The total takeoff weight was originally targeted at around 4.5kg. This excess weight falls within the margin that was accounted for in the propulsion system design, however iterations are being made in the next design to reduce total takeoff weight in order to increase flight performance specifically. As mentioned previously, ground locomotion proved to be highly efficient with battery power lasting for over an hour when operating only in driving mode. Due to the additional mass of the ground locomotion hardware the flight time of the vehicle is reduced from a possible 15 minutes



Figure 6. The vehicle being tested in NASA’s Jet Propulsion Laboratory Mars Yard.

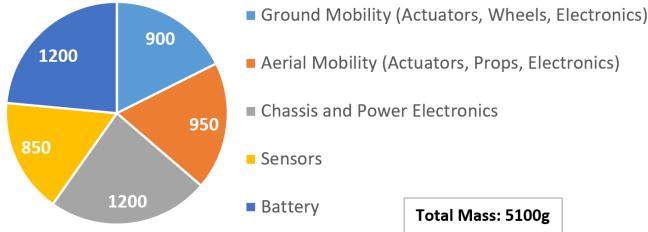


Figure 7. Vehicle mas breakdown for main subsystems.

(without this mass), however the overall performance time has increased significantly, to potentially over an hour with both flight and driving being utilized on a single charge.

3. SYSTEM AUTONOMY

The architecture of the platform’s autonomy system largely leverages work from a previous iteration of the vehicle, the “*rollcopter*”, as presented by Fan, et. al [15]. For completeness, the planning and control components will be covered at a high level here. We refer readers to [15] for a more rigorous and detailed treatment.

We use three frames common to other works employing the differential flatness properties of quadrotor dynamics [16][17]. As illustrated in Fig. 8, these are the world frame $\{W\}$, the body frame $\{B\}$, and the intermediate yaw frame $\{C\}$. We assume a flat plane as the ground with the z-axis of the world and intermediate yaw frame lying normal to this ground plane. The world and intermediate frame differ in orientation by a yawing rotation around the z-axis while the body and intermediate frame differ by rotations about the x and y-axes. The intermediate yaw frame and body frame share their origin at the center of mass of the system, assumed to be located at the center of the central airframe, whose position with respect to the world frame is given by the vector ${}^W\vec{p} \in \mathbb{R}^3$. The superscript in front of the variable denotes the frame of reference, which if absent, is by default the world frame. The intermediate yaw frame is used in order

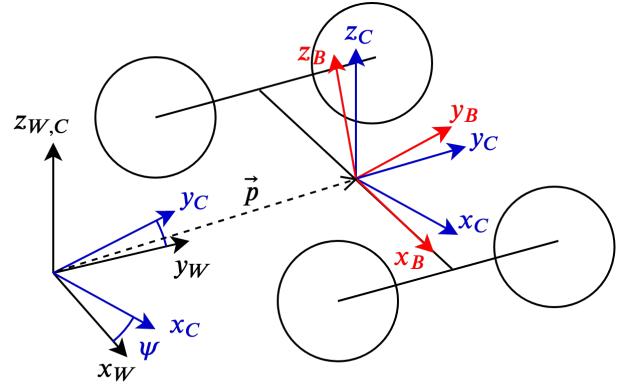


Figure 8. Reference frames of the system

to derive the rotation from the world to the body and is a part of the differential flatness transform that relates the state and actuation of the aerial dynamics in terms of the flat outputs, \vec{p} and ψ . We refer the readers to [17] for the derivation.

Fig. 9 illustrates the full autonomy architecture of our system at a high level and the flow of information between each module. The data from the camera and IMU in the hardware layer is utilized by the Visual Inertial Odometry (VIO) module to produce state estimates, in this case position \vec{p} , velocity \vec{v} , acceleration \vec{a} , orientation (represented as a rotation matrix) \mathbf{R} , and angular velocity $\vec{\omega}$, of the body with respect to the world frame.

The local mapper module in the perception layer utilizes the point cloud data from the Lidar sensor as well as pose information from the VIO module to produce a local representation of obstacles in the environment. This information is used by the global planner in the planning/control layer to plan collision-free hybrid geometric paths represented by a sequence of positions (or vertices) in the world frame along with an associated mobility state for each vertex $s \in \{a, g\}$, representing the aerial and ground modality respectively.

A vertex (and corresponding mobility state) along this global path as well as the local map is passed to the local planner module which is responsible for producing kinodynamically feasible trajectories which follow the geometrically planned path, while reactively avoiding nearby obstacles. Thanks to differential flatness mappings, the local planner is able to generate both flying and rolling trajectories in a unified manner.

Finally, a setpoint along the local planner trajectory (and a corresponding mobility state) is then tracked by the hybrid controller which switches between the discrete rolling and flying controllers depending on the commanded modality. The controller sends either desired motor thrusts \vec{F} or angular wheel velocities $\vec{\theta}$ in the aerial or ground modality respectively.

Localization and Mapping

Localization/State Estimation—To enable the system to autonomously explore and navigate through previously unknown environments, the platform must be able to perform state estimation with reasonable accuracy. We utilize the image stream of the Intel Realsense camera and ORB-SLAM2, an open source system for SLAM [18], to provide pose estimates which are then fused with IMU data via an Extended

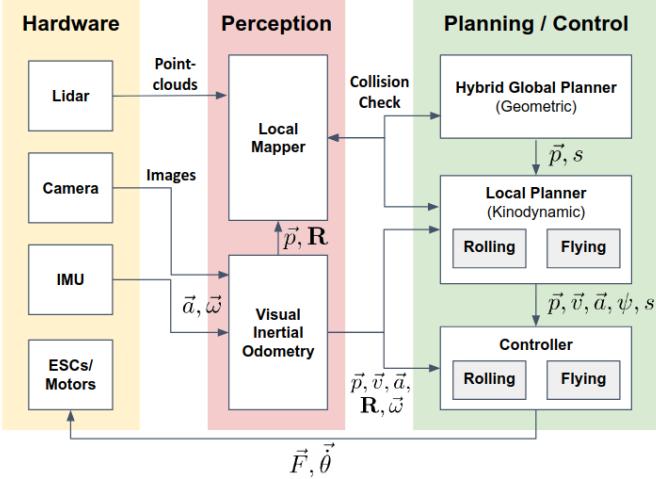


Figure 9. Overview of the Autonomy Software Architecture

Kalman Filter (EKF) on-board the Pixhawk Flight Controller.

Mapping—In order to plan in complex geometric environments, an efficient and real-time mapping solution is required. To this end, the vehicle utilizes the open source OctoMap package [19] to generate volumetric representations of the environment using pointclouds from the Velodyne Puck Lite LIDAR and localization estimates as described above. The Octomap framework utilizes the Octree data structure to efficiently represent occupied, unoccupied, and unknown portions of the environment with adaptive resolution in addition to probabilistic occupancy to account for sensor noise and dynamic environments. The package also efficiently generates an incrementally updated Euclidean Signed Distance Field (ESDF) for path planning use through a brushfire algorithm. The ESDF provides information on a location’s distance to the nearest obstacle. These features are crucial to enable sophisticated reasoning and planning in challenging perception-degraded environments such as the ones presented in the DARPA SubT Challenge.

Global Planning

The occupancy map of the mapping module is used by the global planner to plan geometric paths of shortest cost from the start position to a desired goal position. An A* search planner [20] is run on a grid based graph generated using the 3-dimensional occupancy map and ESDF as described above. This graph consists of vertices corresponding to uniformly discretized positions in the world frame as well as edges that connect each of the vertices, with each vertex having at most 26 connected edges. Vertices that are within a small radius of occupied voxels are considered in collision and are pruned from the graph to prevent infeasible paths or those that result in collision with nearby obstacles.

One important limitation of this graph is the assumption of a flat ground plane which is determined via the bottom-facing range sensor. This graph is regenerated in a local region around the vehicle at regular temporal intervals to provide increased robustness to drift in the map. Each vertex in the graph is assigned a cost based upon its distance from obstacles as well as an additional weighted cost in the z-axis for vertices above the ground plane which are considered as flying vertices.

$$C = C_g \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} + C_a z_i \quad (1)$$

Where C is the cost-to-go (or edge cost) from vertex $i - 1$ to i and C_g and C_a are tunable weights on the cost of euclidean distance in the ground plane and cost of vertices above the ground plane respectively.

This additional cost corresponds to increased power required for flight over rolling and results in more efficient planned paths that prioritize ground mobility. As it is difficult to estimate the amount of flying and rolling mobility required for a given vertex, a simple euclidean distance-to-goal heuristic is used for the A* planner.

The planned path is stored as a tuple of vertices and each vertex is labelled with a binary variable corresponding to the ground or aerial mobility state. Due to our flat ground assumption, all vertices lying above the ground plane are considered aerial and vice versa. The modality state along with the next waypoint in the planned path, is passed to the local planner, and allows the local planner to generate the appropriate trajectory for the controller to track, as well as to trigger transitions between rolling and flying modalities.

Local Planning

While the globally planned path is able to reason about costs from purely geometric information, the planned paths are similarly geometric and are not conducive to be tracked on a physical platform with dynamic and kinematic constraints. In addition, while the globally planned path should avoid regions near obstacles due to the cost incorporated from the ESDF, the frequency at which a new path is generated is not fast enough to reactively avoid obstacles at our desired velocities. Thus, a local planner is needed that plans kinodynamically feasible trajectories that track the globally planned path as closely as possible while reactively avoiding obstacles from the 3D LIDAR as well as the generated occupancy map.

We employ a motion primitive approach based upon the requirements above. Multiple motion primitives are generated given the current vehicle state and a predetermined set of end states in a 180° arc around the vehicle (in the x-y plane of the intermediate yaw frame). These primitives are dynamically generated polynomial-based trajectories produced by employing techniques described in a prior work for efficiently generating minimum-snap trajectories for quadrotors [21]. In the aerial modality, we also include sets of endpoints at certain angles above and below the plane as illustrated in Fig. 10. We show that regardless of the mobility modality (rolling or flying) we can use the same framework to plan both trajectories on the ground and in the air by leveraging the differential flatness properties of both aerial quadrotor and rolling differential drive vehicles.

Once the primitives are generated, the local planner prunes out trajectories resulting in collisions via a KD-tree search and the map given by the mapping module. The remaining trajectories are evaluated based upon a weighted sum of the distance to the desired waypoint given by the globally planned path as well as the distance to the nearest obstacle. Thus, the local planner produces a kinodynamically feasible trajectory that balances tracking of the global path and reactive avoidance of local obstacles.

Unified Hybrid Differential Flatness—As the dynamics and actuation of the drivcopter are similar to that of a conven-

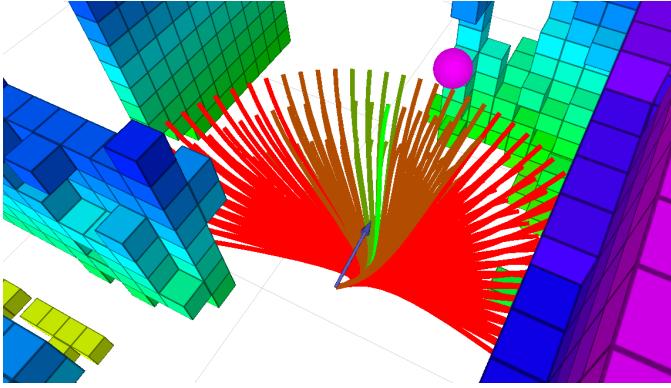


Figure 10. Polynomial Based Motion Primitives for the Aerial Modality [15]

tional quadrotor, we are able to employ approaches utilized on other quadrotor platforms. In the aerial modality, the state variables are given as $x_a = (\vec{p}, \vec{v}, R_a, \vec{\omega}_a) \in \mathbb{R}^3 \times \mathbb{R}^3 \times SO(3) \times \mathbb{R}^3$, along with the control inputs $u_a = (|F|, {}^bM_x, {}^bM_y, {}^bM_z)$ which maps to the thrust of the four motors \vec{F} . Differential flatness is a mathematical property commonly used in trajectory planning for quadrotors as it allows the multiple state variables and control inputs of the system to instead be written as algebraic functions of four flat outputs: $z_a = (\vec{p}_a, \psi)$ and its derivatives. In other words, a planned smooth trajectory given in terms of the flat outputs can be tracked with an underactuated quadrotor as long as the derivatives of the trajectory are within the system's dynamic limits [22]. We define aerial trajectories $z_a(t) : [t_0, T] \rightarrow \mathbb{R}^3 \times SO(2)$ as smooth curves parameterized by t . The smooth differential flatness mapping is given by $(x_a, u_a) = \Phi(z_a, \dot{z}_a, \ddot{z}_a, \ddot{\ddot{z}}_a)$.

Similarly, we consider the dynamics of the rolling mobility state of the drivcopter, a skid steered vehicle, to closely approximate that of a two-wheeled differential drive robot and actuate the two left side wheels at equivalent angular velocities as well as on the right side. In this ground modality, we instead consider the state variables $x_g = (p_g, v_g, R_g, \omega_g) \in \mathbb{R}^2 \times \mathbb{R}^2 \times SO(2) \times \mathbb{R}$, along with the control inputs $u_g = ({}^r\vec{v}_{gx}, \psi)$, the linear velocity and yaw rate, which can be mapped to a left and right angular wheel velocities via the kinematics model. Akin to the quadrotor approach, it has been proven that a differential flatness mapping exists for differential drive ground robots [23]. In this case, the state and control inputs can be mapped to algebraic functions of the flat outputs $z_g = (\vec{p}_g)$ and its derivatives via $(x_g, u_g) = \Psi(z_g, \dot{z}_g)$. Also, we define ground trajectories as smooth curves parametrized in t , $z_g(t) : [t_0, T] \rightarrow \mathbb{R}^2$. As the flat space for the ground modality is in the subspace of the flat space for the aerial modality, as illustrated in Fig. 11, we are able to utilize the same framework for generating trajectories for both ground and aerial mobility.

Controllers for Flying and Rolling Mobility

As there are independent sets of actuators for flying and rolling modalities, we employ two discrete controllers for either modality and switch between the two based upon the desired mobility state command from the local planner (which is itself given by the globally planned path). In addition to the desired modality, the controller receives a setpoint from the local planner module in order to track the trajectory produced by the methods described above.

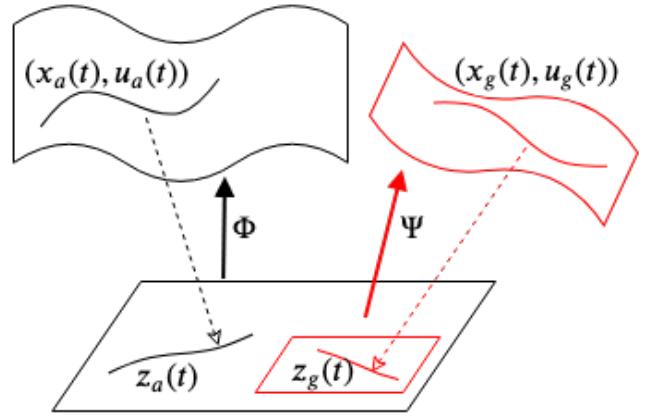


Figure 11. Differential Flatness Mappings for Both Aerial and Ground Modalities [15]

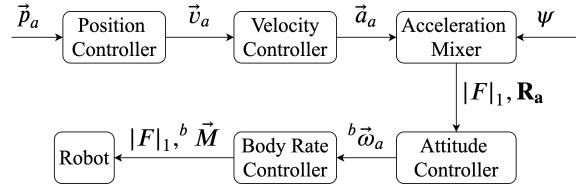


Figure 12. Architecture of the flying controller [15]

Flying Controller—The flying controller receives a desired position, velocity and acceleration in x, y, z and desired heading from the trajectory generated by the local planner and tracks the states via a cascaded control loop as shown in Fig. 12. A feedback position controller uses a proportional control law which is cascaded to the velocity controller. The velocity controller then generates a desired acceleration using a PID control law on the velocity error and a feed-forward term from the position controller. This desired acceleration along with feed-forward acceleration and desired heading is used to produce a desired total thrust $|F|_1$ and attitude \mathbf{R}_a by the acceleration mixer. Finally, The desired thrust and attitude is then tracked via the nonlinear quaternion-based attitude controller of the PX4 flight stack [24].

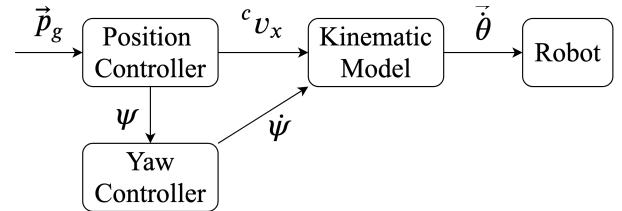


Figure 13. Architecture of the rolling controller

Rolling Controller—In the rolling modality, we treat the robot as a differential drive vehicle and employ a control architecture shown in Fig. 13 to track position setpoints from the local planner. The equations for this simple rolling controller are detailed below and follow closely with the control law from [15]. Note that estimated quantities coming from the VIO module are denoted with a hat (e.g. \hat{p}) and desired quantities are denoted with a superscript d (e.g. p^d). Also note that in this ground modality, positions p_g and velocities v_g are in 2-dimensional space with orientation given by ψ .

Position Controller: A desired linear velocity in the x-axis of the body frame is calculated using a proportional control

law on error in position as follows:

$${}^w e_g = {}^w p_g^d - {}^w \hat{p}_g \quad (2)$$

$${}^c e_g = R(\psi_{yaw}) {}^w e_g \quad (3)$$

$${}^c v_{gx} = {}^{pos} k_p ({}^c e_g) \quad (4)$$

Where ${}^{pos} k_p$ is the proportional gain term on the error in position.

Yaw Controller: In addition, a desired yaw is generated from the position setpoint and tracked via a proportional control law on error in yaw which outputs a desired yaw rate:

$${}^w \psi^d = \tan^{-1}({}^w e_{gy} / {}^w e_{gx}) \quad (5)$$

$${}^w e_\psi = d_{S(1)}({}^w \psi^d, {}^w \psi) \quad (6)$$

$${}^w \dot{\psi} = {}^w k_p ({}^w e_\psi) \quad (7)$$

Where ${}^w k_p$ is the proportional gain term on the error in orientation.

Kinematics Model: The desired linear velocity and yaw rate are then mapped to the left and right angular wheel velocities via the kinematics model:

$$\begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & b \\ 1 & -b \end{bmatrix} \begin{bmatrix} {}^c v_{gx} \\ {}^w \dot{\psi} \end{bmatrix} \quad (8)$$

Where r is the wheel radii and b is the distance of the wheel from the center of the wheel axle.

Finally, the left and right angular wheel velocities are tracked by a lower level proportional controller on the Teensy microcontroller using angular velocity estimates from the motor encoders and varying the PWM duty cycle of each motor.

Transitions between Rolling and Flying—While rolling and flying trajectories are generated and tracked discretely from the other, transitions are treated as aerial takeoff/landing commands and are passed to the hybrid controller from the local planning module based upon a change in the mobility state of the current and upcoming waypoint in the planned geometric path.

Transitioning from rolling to flying is treated as a takeoff command to a predetermined height with fixed velocity in the world frame's z-axis. We close the loop on the bottom-facing GARMIN 1-D LIDAR to terminate the transition.

Transitioning from flying to rolling is similarly treated as a landing command at a fixed velocity in the world frame's z-axis which is terminated once the bottom-facing range sensor achieves the corresponding value for being considered grounded.

Preliminary Results in Simulation

A vehicle model was developed for use in the gazebo simulation environment which enabled quick and efficient testing of the autonomy algorithms during development (Fig. 14). We utilize the ETH Zurich developed RotorS simulation add-on to simulate the physics of multirotor vehicles.

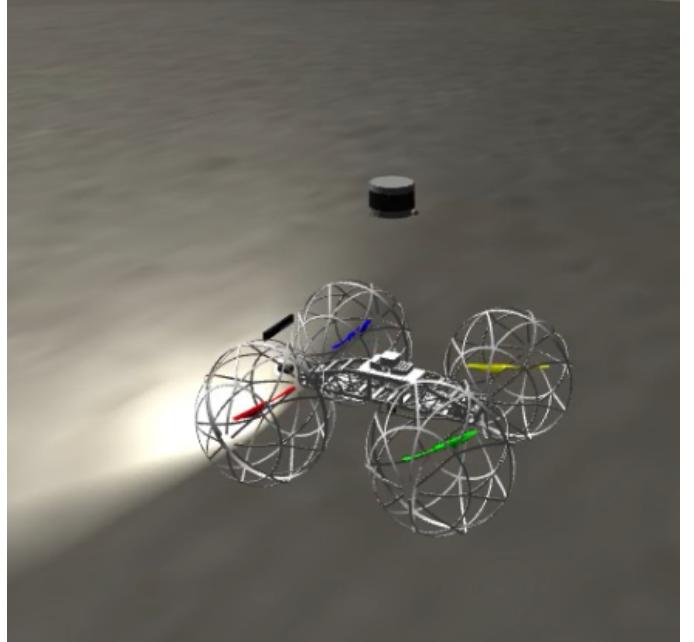


Figure 14. Vehicle Robot Model in the Gazebo Simulation Environment

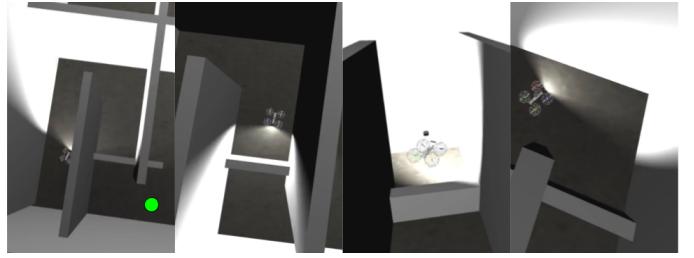


Figure 15. Vehicle autonomously traversing through the maze. The global planner goal is indicated in the first frame by the green circle.

We utilize the simulation environments provided by the DARPA SubT Challenge coordinators which feature subsurface environments with maze-like tunnels. This environment provides a good testbed for tuning and evaluating effectiveness of the reactive local planning algorithms as the tunnels feature relatively tight spaces with sharp corners.

To test the ability of the hybrid global planner as well as the planning and controls algorithms downstream of the global planner, we utilize the simple geometric primitives such as boxes to construct a small maze that can only be completely traversed with both ranges of mobility modalities. Early tests show the robot is able to successfully complete the maze environment through utilizing both rolling and flying modalities, as illustrated in Fig. 15.

4. CONCLUSIONS

The paper presented design and prototype of a hybrid aerial/ground autonomous vehicle. The vehicle consists of a coaxial quadrotor and four independently actuated wheels. The proposed concept and design for the vehicle was validated by fabricating a prototype and running experiments. However, further work is required to improve the design and optimize the performance of the platform. Additionally an autonomy system was implemented for the vehicle, the

preliminary results of which show promise for extension to novel planning and controls approaches for general hybrid aerial/ground vehicles. One capability of interest for a rolling vehicle that can also produce thrust normal to points of contact with the ground is a method of hybrid force/velocity control, a well-studied problem in robot manipulators. We hypothesize that enabling the robot to "conform" to highly non-planar and unstructured terrain via control of contact forces normal to the local plane of contacts in parallel with velocity control tangent to the local plane would increase both speed and efficiency. In addition, the robustness of the vehicle to collisions via the wheeled propellers makes this vehicle a well-suited platform for navigation through collisions. Thomas, Et al. [25] presents a novel state estimation pipeline leveraging contacts which could be extended to aid in pure navigation through collision in cases of loss of vision-based state estimation in GPS-denied, perceptually degraded environments. Furthermore, future goals for the platform in the context of the DARPA SubT Challenge envision the vehicle working in concert with other heterogeneous agents with diverse mobility capabilities to explore a wider range of areas in challenging subterranean environments. The cooperative mapping of such environments will require a framework that enables the inter-agent sharing of sensor-information and mapping data which has been presented in prior work [26].

ACKNOWLEDGMENT

The authors would like to thank the rest of the JPL SubT team for their support in the development of the hybrid aerial/ground vehicle. Particularly, we would like to thank Meriem Miled for her work in preparing a thorough survey of hybrid aerial/ground vehicles, Fernando Chavez for his support in hardware preparation, and Matthew Anderson for his support in flight safety and general hardware advice regarding multi-rotor vehicles.

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

- [1] A. Kalantari and M. Spenko, "Design and experimental validation of hytaq, a hybrid terrestrial and aerial quadrotor," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4445–4450.
- [2] J. R. Page and P. E. I. Pounds, "The quadroller: Modeling of a uav/ugv hybrid quadrotor," *International Conference on Intelligent Robots and Systems (IROS)*, 2014.
- [3] K. Tanaka, D. Zhang, S. Inoue, R. Kasai, H. Yokoyama, K. Shindo, K. Matsuhiro, S. Marumoto, H. Ishii, and A. Takanishi, "A design of a small mobile robot with a hybrid locomotion mechanism of wheels and multi-rotors," in *2017 IEEE International Conference on Mechatronics and Automation*. IEEE, 2017.
- [4] "<https://dronemajor.net/brands/inkonova/products/tilt-scout>," *Tilt Scout*, Inkanova.
- [5] "<https://www.hobbywow.com/>," *XDrone Quadcopter Car*, hobbyway.
- [6] S. Mintchev and D. Floreano, "A multi-modal hovering and terrestrial robot with adaptive morphology," in *Proceedings of the 2nd International Symposium on Aerial Robotics*, 2018.
- [7] S. Morton and N. Papanikopoulos, "A small hybrid ground-air vehicle concept," *International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [8] K. Li, B. Han, Y. Zhao, and C. Zhu, "Motion planning and simulation of combined land-air amphibious robot," in *IOP Conference Series: Materials Science and Engineering*, vol. 428. IOP Publishing, 2018, p. 012057.
- [9] G. M. M. P. M. Y. S. L. Anthony Stroffolino, Lauren Davis and M. Kofron, "Design of a hybrid exploration robot for air and land deployment (h.e.r.a.l.d) for urban search and rescue applications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014.
- [10] A. Kalantari and M. Spenko, "Hybrid aerial and terrestrial vehicle," June 23 2015, US Patent 9,061,558.
- [11] A. akbar Agha-mohammadi, "Rollocopter: Novel hybrid aerial-ground vehicle for failure-resilient, energy-efficient, and agile mobility," US Patent, Submitted.
- [12] S. Sabet, A. A. Agha-Mohammadi, A. Tagliabue, D. S. Elliott, and P. E. Nikravesh, "Rollocopter: An Energy-Aware Hybrid Aerial-Ground Mobility for Extreme Terrains," in *IEEE Aerospace Conference Proceedings*, vol. 2019-March. IEEE, 2019, pp. 1–8.
- [13] A. akbar Agha-mohammadi, "Cobots: Autonomous cooperatives robots for resilient mobility on titan and beyond," US Patent, Submitted.
- [14] A. Tagliabue, S. Schneider, M. Pavone, and A. a. Agha-mohammadi, "The shapeshifter: a multi-agent, multi-modal robotic platform for the exploration of titan," in *IEEE Aerospace Conference, Big Sky*.
- [15] D. Fan, R. Thakker, T. Bartlett, M. B. Miled, L. Kim, E. Theodorou, and A. Agha-mohammadi, "Autonomous Hybrid Ground / Aerial Mobility in Unknown Environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [16] M. L. Taeyoung Lee and N. H. McClamrock, "Geometric tracking control of a quadrotor uav on $se(3)$," *49th IEEE Conference on Decision and Control*, 2010.
- [17] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2520–2525, 2011.
- [18] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [19] 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.
- [20] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [21] C. Richter, A. Bry, and N. Roy, *Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments*. Cham: Springer International Publishing, 2016, pp. 649–666. [Online]. Available: https://doi.org/10.1007/978-3-319-28872-7_37
- [22] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadro-

- tor,” *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 20–32, 2012.
- [23] C. P. Tang, “Differential flatness-based kinematic and dynamic control of a differentially driven wheeled mobile robot,” in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2009, pp. 2267–2272.
- [24] D. Brescianini, M. Hehn, and R. D’Andrea, “Nonlinear quadrocopter attitude control: Technical report,” ETH Zurich, Tech. Rep., 2013.
- [25] T. Lew, T. Emmei, D. Fan, T. Bartlett, A. Santamaria-Navarro, R. Thakker, and A. a. Agha-mohammadi, “Contact inertial odometry: Collisions are your friend,” in *The International Symposium on Robotics Research (ISRR)*.
- [26] P. V. Eric Heiden, Daniel Pastor and A. akbar Agha-mohammadi, “Heterogeneous sensor fusion via confidence-rich 3d grid mapping: Application to physical robots,” in *International Symposium on Experimental Robotics (ISER)*.