To Land or Not to Land? Digital Twin Approach for Safe Landing of Multirotors on Complex Terrain

Joshua Taylor[1,2], Wei-Yun Yau[2], Guillaume Sartoretti[1], and Efe Camci[2]

***Abstract*—Simulation has become a powerful tool in robotics, enabling safe and rapid testing of robots prior to deployment. It also holds great potential to aid in real-time decision-making by emulating the surroundings on-the-fly using onboard perception modules. In this work, we introduce a simulation-based decision-making framework for the safe autonomous landing of a multirotor unmanned aerial vehicle (UAV) over difficult terrains. Our framework consists of first obtaining a terrain model using onboard sensors. It then assesses a range of possible landing configurations using a digital twin of the UAV in a high-fidelity physics-based simulation with PX4 firmware running in-the-loop. Finally, it identifies the safest simulated landing configuration and uses this to perform the landing in the real world. We validate our framework over complex, previously unseen terrains with a real multirotor. Our framework matches human-level judgment on how best to land on multiple instances of complex terrain and perform safe landings in real-world tests. This can be seen at https://youtu.be/G3nzaBFDHBY. These results show the potential of our approach for aerial robotics applications involving physical interaction.**

I. Introduction

Physics-based simulators have long been a useful tool for roboticists in testing or training robotics systems prior to deployment [1]. More recently, these simulators have also been explored as a means of enhancing robot interaction with the environment during deployment.

There are many instances of this in the literature. For robot manipulation tasks in [2]–[4], actions are tested in simulation prior to real-world execution. This provides a prediction of the physical consequences of these actions on the environment, which is used to adjust the parameters of the control sequence.

Similar techniques using real-time physics-based simulation have also been applied to mobile robotics. In [5], a simulated digital twin is run in parallel with a real quadrupedal robot to detect terrain anomalies. Furthermore, for unmanned aerial vehicles (UAVs), a digital twin is utilized for mission guidance in [6] and to improve landing accuracy in [7].

In the field of simulation-based decision-making, a method of particular interest to us is robot imagination [8]–[10]. It uses physics-based simulation as a fully explainable method for enabling robots to interact with previously unseen household items. This approach involves experimenting with a virtual model of an object, to determine whether the object's affordances [11] can be leveraged to satisfy a desired task.

Inspired by robot imagination, we introduce a framework for decision-making during contact-based UAV applications.

[1]National University of Singapore (NUS), Singapore
[2]Institute for Infocomm Research (I^{2}R), A*STAR, Singapore

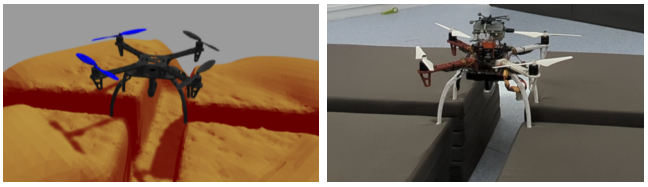


Fig. 1. Safest landing identified in simulations (left), performed in the real world (right).

We validate the framework by applying it to the safe autonomous landing of a multirotor on difficult terrain. This task is an unavoidable and often challenging form of interaction with the environment. Various approaches have been proposed to address this challenge, including mechanical [12], [13], and software-based [14] solutions. Belonging to the latter, our framework involves four steps:

1) obtain a terrain model using onboard sensors,
2) test a range of landing configurations with a digital twin in high-fidelity software-in-the-loop (SITL) simulations,
3) identify the safest landing configuration,
4) perform the safest landing in the real world (Fig. 1).

We test the framework's components in isolation, keeping the real-time implementation as future work.

In the remainder of this paper, Section II defines the safe landing problem, Section III describes the proposed framework and Section IV discusses the testing of its components. Finally, key conclusions are drawn in Section V.

II. Problem Statement

Position control of multirotor UAVs is typically achieved by commanding desired position set-points in three axes: x, y, z, and a heading (yaw) set-point, ψ. These are relative to a local reference frame and are denoted as:

$$p = [x, y, z, \psi]^T. \tag{1}$$

To hover above a landing point (x_{lp}, y_{lp}), a UAV is first commanded with $[x_{lp}, y_{lp}, z_{lp} + \Delta z]^T$, incorporating a safety offset in z-axis. Then, one needs to find a safe orientation (ψ_{safe}) to land based on terrain characteristics. Once a favorable ψ value is determined, the UAV is commanded with:

$$p_{land} = [x_{lp}, y_{lp}, z_{lp} + \Delta z, \psi_{safe}]^T, \tag{2}$$

followed by removing the safety offset, Δz, in the z-axis to perform the safe landing. Our specific problem in this work is to determine ψ_{safe} based on terrain characteristics observed through onboard perception. We assume that the values of x_{lp}, y_{lp}, z_{lp} have already been set.

III. Method

The general robot imagination [8]–[10] framework begins with *model acquisition* of relevant objects in the environment. Next, candidate actions are performed on the object in *simulation*. Based on the interactions observed, one can perform *classification* to determine how suitable the object and the candidate actions are for the given task. Finally, the robot *acts* according to the classification results. Figure 2 depicts these components in the context of UAV landing.

In this work, we apply the framework to a DJI F450 quadrotor with a modified rectangular landing gear. The UAV has a Pixhawk 4 flight controller running open-source PX4 firmware [15]. We use OptiTrack motion capture system for localization during flight. In the following subsections, we describe our implementation of the framework's components.

A. Model Acquisition

To obtain the terrain model, we mount an Intel Realsense D435 depth camera to the UAV, which produces point cloud data (PCD). We record the PCD with an onboard computer and convert it to a stereolithography (STL) file on an external computer. This conversion accounts for the pose of the camera relative to the UAV's center of mass and filters out any nearby data points unrelated to the terrain. The STL file is used by a spatial data file (SDF) to define the terrain's appearance and collision properties in simulation.

B. Simulation

We assess the interaction during landing using physics-based simulation of the obtained terrain model and a digital twin of the UAV. The twin has near-identical dimensions and physical properties to the real UAV. We also use SITL to emulate the UAV's flight control.

We implemented the simulation in Gazebo. The Robot Operating System (ROS) is used to enable communication between the PX4 firmware and Gazebo (via the MAVROS package [16]) and coordinate the simulation procedure.

C. Classification

We select ψ_{safe} from a set of headings evenly distributed across the possible range. Note that the range need not always be 0-360° for symmetrical airframes. For every simulated landing, once the UAV has settled on the ground, we measure the nutation angle, N (i.e., the angle between the local z-axis and the UAV body z-axis). Zero nutation angle corresponds to a perfectly level landing. To account for stochasticity in high-fidelity simulations, we run m simulations for every yaw angle candidate and record the nutation angle as:

$$N(\psi_{act}(j), i) = cos^{-1}(cos\phi_i cos\theta_i), \tag{3}$$

where

$$\begin{aligned} I &= \{i : i \in \mathbb{Z},\ \ 0 < i \leq m\}, \\ J &= \{j : j \in \mathbb{Z},\ \ 0 < j \leq n\}. \end{aligned} \tag{4}$$

The terms ϕ_i and θ_i refer to the roll and pitch angles at each simulation once the UAV has settled on the ground. The

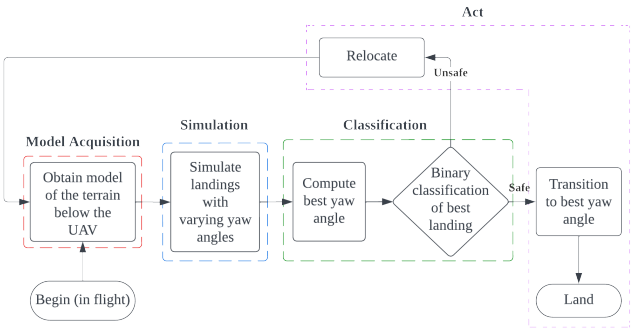


Fig. 2. The decision-making process for landing a UAV on complex, previously unseen terrain. The process is categorized into the stages defined by the robot imagination framework [8]–[10].

heading angle candidate, $\psi_{act}(j)$ is selected from n evenly spaced yaw angle candidates between 0 and the largest test angle, ψ_{max}, at each simulation batch as:

$$\psi_{act}(j) = \frac{j\psi_{max}}{n}. \tag{5}$$

We then calculate a final safety score (S) for every yaw angle as:

$$S(\psi_{act}(j)) = \max(N(\psi_{act}(j), i)). \tag{6}$$

Finally, we determine the most favorable landing orientation among the candidates, $\psi^* = \psi_{act}(j^*)$, using:

$$j^* = \arg\min_j\ S(\psi_{act}(j)). \tag{7}$$

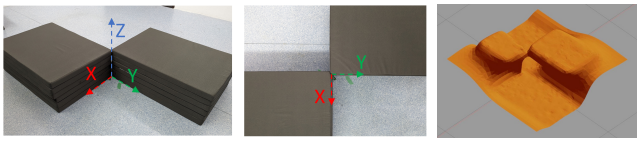
$S(\psi_{act}(j*))$ is compared to a threshold to determine if the chosen landing is safe. If it is, the corresponding yaw angle qualifies as the safe heading, ψ_{safe}. Here, we define the threshold by taking the smaller value from two tests. Firstly, we determine the maximum nutation angle that would tip the UAV back on its landing gear on a level surface. Secondly, we take the nutation angle that results in collision between the propeller and level ground when pivoting about a leg.

IV. Experiments

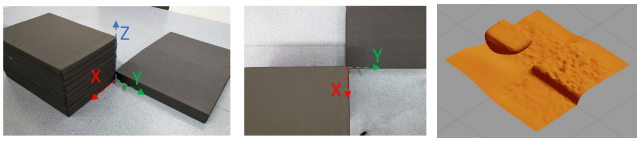
We test the framework's components on multiple real-world landing scenarios, as described in the following subsections.

A. Dataset

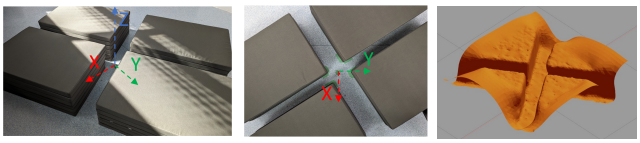
We assess the framework over three terrains, as shown in Fig. 3. The landing point (x_{lp}, y_{lp}) is set at the origin of the x- and y-axes shown in the figure, z_{lp} is set at the top-most surface of the terrain, and a safety offset (Δz) of 0.8 m is used. For a point of comparison, one of our lab's drone pilots exercised their judgment on how they would intuitively land the UAV on such terrain. In the case of Terrain 2, the pilot would not attempt to land. However, for Terrain 1 and Terrain 3, the pilot could visually identify landing orientations that would allow for contact of all four legs on the top layer of the terrain, forming a level base. This occurs at approximately 135° and 125°, for Terrain 1 and Terrain 3, respectively.



(a) Terrain 1.



(b) Terrain 2.



(c) Terrain 3.

Fig. 3. Real-world images of the terrains (two columns on the left) and their simulated versions (right column) obtained through the real point cloud. The UAV is commanded to hover at 0.8m above the top-most surface, in line with the z-axis indicated in the images.

Figure 3 shows the models in Gazebo following the conversion to STL. For the most part, the accuracy of the models is satisfactory for our purposes, as the topmost surfaces with which the initial interaction occurs are accurately recreated.

B. Running the Simulations

The simulation hyper-parameters for all terrains are shown in Table I. Given the UAV body's symmetry, we consider half the possible range of yaw angles (i.e., $\psi_{max} = 180°$).

The results of each set of simulations are shown in Fig. 4. For Terrain 1, the safest heading was 135°, which yielded a safety score of 0.035 rad. This simulated landing is shown in Fig. 5a. All other headings resulted in tipping of the UAV, with almost all cases resembling that shown in Fig. 5b. This result perfectly matches the pilot's ideal landing orientation.

As expected, for Terrain 2, no safe landing orientations were identified. The best safety score was 1.37 rad at a heading of 63° which is above the threshold. All cases resulted in the UAV tipping, often to the extent shown in Fig. 5c.

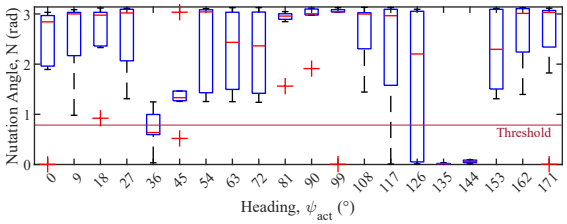
Simulations on Terrain 3 yielded the safest landing orientation at a heading of 126°, which is only 0.8% larger than our estimate. The error is likely due to the discretization of the full range of possible angles. The simulated landing is shown in Fig. 5d, which yielded a safety score of 0.033 rad.

C. Real-World UAV Landing

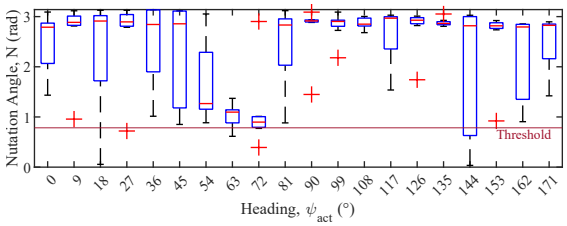
To test the framework's *act* component, we perform real-world landings of the UAV on Terrain 1 and Terrain 3, for which simulations yielded safe landing orientations. The UAV is commanded to autonomously land using the obtained value of ψ_{safe}. These landings are shown in Fig. 6. In both cases, the UAV lands safely at an almost level orientation.

TABLE I
SIMULATION HYPER-PARAMETERS. THESE ARE KEPT CONSTANT FOR EXPERIMENTS WITH ALL THREE TERRAIN MODELS.

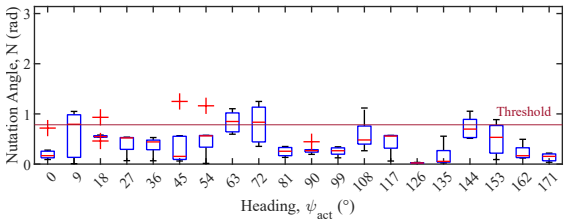
Hyper-parameter	**Value**
Number of yaw angles, n	20
Maximum yaw angle, ψ_{max}	180°
Number of iterations, m	8
Safe landing threshold	$\frac{\pi}{4} rad$



(a) Final nutation angles on Terrain 1. The lowest maximum nutation angle occurred at 135°. Therefore, this is the safest ψ_{act}.



(b) Final nutation angles on Terrain 2. The lowest maximum nutation angle occurred at 63°. Therefore, this is the safest ψ_{act}.



(c) Final nutation angles on Terrain 3. The lowest maximum nutation angle occurred at 126°. Therefore, this is the safest ψ_{act}.

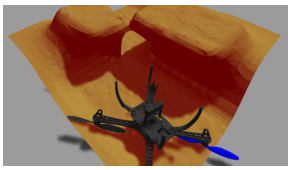
Fig. 4. Final nutation angles for landings on the terrain models with various heading choices.

D. Limitations

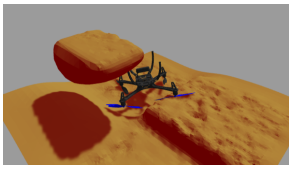
Despite safe outcomes in all test cases, our framework still faces limitations. Firstly, the depth camera does not provide a complete terrain model. Physical characteristics such as compliance and friction are not included, and the terrain's geometry can be incomplete. The latter is evident from the models of Terrain 1 and Terrain 3 in Fig. 3 but could be improved by stitching together PCD from multiple angles.

The next limitations concern the landing classification from Section III-C. For testing on Terrain 2, we noted outliers where the UAV tipped over during landing but rolled into a level final position. This was incorrectly classified as safe as we only consider the UAV's final state. This presents a need for an improved safety metric using factors like the offset from $[x_{lp}, y_{lp}, z_{lp}]^T$ in the final resting position or changes in UAV pitch and roll over time. Furthermore, our landing definition limits us to the exploration of a single variable,

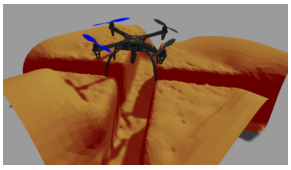
(a) Safest landing on Terrain 1 ($\psi_{act} = 135°$, i = 3).



(b) Unsafe landing on Terrain 1 ($\psi_{act} = 9°$, i = 6).



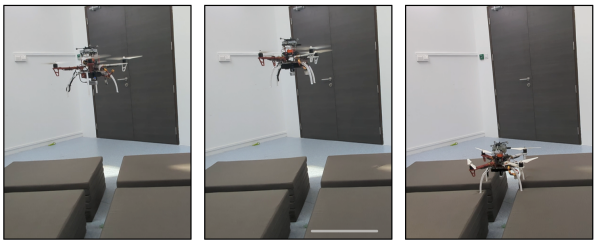
(c) Unsafe landing on Terrain 2 ($\psi_{act} = 0°$, i = 0).



(d) Safest landing on Terrain 3 ($\psi_{act} = 126°$, i = 0).

Fig. 5. Key snapshots from the simulations.

(a) Successful landing on Terrain 1 with $\psi_{act} = 135°$.



(b) Successful landing on Terrain 3 with $\psi_{act} = 126°$.

Fig. 6. Successful real-world UAV landings.

ψ_{act}. To increase the chances of finding a safe landing, we can explore various values for x_{lp}, y_{lp} or for speed of descent.

Finally, the set of simulations for each terrain took approximately one hour, which is not suitable for real-time implementation. We will optimize the simulation procedure by running Gazebo simulations headless with a real-time factor above 1.0 or using simulation software like Nvidia's ISAAC that can run on GPU.

V. Conclusion

In this work, we applied a robot imagination [8]–[10] inspired framework to UAV landings on complex, previously unseen terrain. This involved obtaining a terrain model using onboard sensors, simulating repeated landings to determine the best landing orientation, and using this information to land the UAV in the real world. These components were tested independently and yielded results that matched human pilot judgment. This shows the potential for applying the framework to more complex aerial physical interaction scenarios.

Future work will be on improving our safety metric and simulation run-time. Furthermore, we will apply the framework to more complex UAV systems involving interaction, such as UAVs equipped with grippers.

Acknowledgment

Joshua Taylor is an A*STAR scholar under the Singapore International Graduate Award (SINGA) framework. This research is partially supported by A*STAR C211518001 and C221518005.

References

[1] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. Von Stryk, "Comprehensive simulation of quadrotor uavs using ros and gazebo," in *Simulation, Modeling, and Programming for Autonomous Robots: Third International Conference, SIMPAR 2012, Tsukuba, Japan, November 5-8, 2012. Proceedings 3*. Springer, 2012, pp. 400–411.

[2] L. Kunze and M. Beetz, "Envisioning the qualitative effects of robot manipulation actions using simulation-based projections," *Artificial Intelligence*, vol. 247, pp. 352–380, 2017.

[3] M. Beetz, D. Beßler, A. Haidu, M. Pomarlan, A. K. Bozcuoğlu, and G. Bartels, "Know rob 2.0—a 2nd generation knowledge processing framework for cognition-enabled robotic agents," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 512–519.

[4] P. Mania, F. K. Kenfack, M. Neumann, and M. Beetz, "Imagination-enabled robot perception," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 936–943.

[5] G. Haddeler, H. P. Palanivelu, F. Colonnier, Y. C. Ng, A. H. Adiwahono, Z. Li, C.-M. Chew, and M. Y. M. Chuah, "Real-time terrain anomaly perception for safe robot locomotion using a digital double framework," *Robotics and Autonomous Systems*, vol. 169, p. 104512, 2023.

[6] Y. Yang and W. Meng, "Auxiliary decision-making strategy for physical quadrotor by actively interacting with virtual system," in *2023 8th International Conference on Automation, Control and Robotics Engineering (CACRE)*. IEEE, 2023, pp. 275–280.

[7] J. Uddin, M. F. Wadud, R. Ashrafi, M. G. R. Alam, and M. K. Rhaman, "Landing with confidence: The role of digital twin in uav precision landing," in *2023 10th International Conference on Recent Advances in Air and Space Technologies (RAST)*. IEEE, 2023, pp. 1–6.

[8] H. Wu and G. S. Chirikjian, "Can i pour into it? robot imagining open containability affordance of previously unseen objects via physical simulations," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 271–278, 2020.

[9] H. Wu, D. Misra, and G. S. Chirikjian, "Is that a chair? imagining affordances using simulations of an articulated human body," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7240–7246.

[10] H. Wu, X. Meng, S. Ruan, and G. S. Chirikjian, "Put the bear on the chair! intelligent robot interaction with previously unseen chairs via robot imagination," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6276–6282.

[11] J. J. Gibson, *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin, 1979.

[12] M. Ikura, L. Miyashita, and M. Ishikawa, "Stabilization system for uav landing on rough ground by adaptive 3d sensing and high-speed landing gear adjustment," *Journal of Robotics and Mechatronics*, vol. 33, no. 1, pp. 108–118, 2021.

[13] Y. S. Sarkisov, G. A. Yashin, E. V. Tsykunov, and D. Tsetserukou, "Dronegear: A novel robotic landing gear with embedded optical torque sensors for safe multicopter landing on an uneven surface," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1912–1917, 2018.

[14] M. S. Alam and J. Oluoch, "A survey of safe landing zone detection techniques for autonomous unmanned aerial vehicles (uavs)," *Expert Systems with Applications*, vol. 179, p. 115091, 2021.

[15] L. Meier, D. Honegger, and M. Pollefeys, "Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 6235–6240.

[16] "mavros - ROS Wiki." [Online]. Available: http://wiki.ros.org/mavros