Quantifying the Design-Space Tradeoffs in Autonomous Drones

Extended Abstract

Ramyad Hadidi, Bahar Asgari, Sam Jijina, Adriana Amyette, Nima Shoghi, Hyesoon Kim
Georgia Institute of Technology

1. Motivation

Over the last decade, significant progress has been made in developing autonomous drones, with countless applications such as aerial mapping, natural disaster recovery, search and rescue, ecology, and entertainment. Thus, many control, planning, and perception methods have been assimilated for drones. Nevertheless, drones must operate under quite different conditions than any other compute-based agent. First, weight and power are restrictive parameters in drones. Second, drones must arbitrate between their limited compute, energy, and electromechanical resources not only based on the current tasks and local conditions (*e.g.*, wind), but also according to the flight plan. Despite huge technological advances, these problems have been approached in isolation, and the end-to-end system design-space tradeoffs are largely unknown.

As a result of such isolated problem solving, architecting end-to-end drone systems and their computation landscape still remains an open question. For example (Figure 1), if we are making a special chip for drones, is improving processor performance useful and, if yes, is it because of energy savings or better control? How useful is improving processor power efficiency given that the majority of power consumption is coming from resources other than computing power? Should we focus on optimizing the flight-related tasks, or should we focus on autonomy tasks? These questions relate to creating cost-effective solutions with low integration cost, reasonable development time, and effectiveness on drone metrics.

To answer such questions and solve worthy research problems, we need to understand fundamental drone subsystems, classify drone computations and their requirements, extract design-space tradeoffs, and have access to a reproducible experimental platform.

2. Limitations of Prior Work

Prior studies [3, 8] have proposed a closed-loop simulator and benchmark suite for autonomous tasks in drones. The discussions only pertain to *high-speed* drones, which does not completely answer the previous questions for the following reasons. In contrast to the assumptions made in these studies, we argue that, first, mission planning computation does not increase hovering time since they have a relaxed deadline [14]. Even in high-speed, indoor, and cluttered environments, new algorithms have been proposed to enable fast planning [9, 10]. Second, collision detection does not necessarily require heavy computations (*e.g.*, using laser-range, infrared, or RGBD sensors, or even microcontrollers) [4, 6, 11, 5]. Third, localization is a highly active research area and does not necessarily limit current drone speeds (*e.g.*, real-time odometry and NASA JPL's autonomous racing) [13, 12, 7]. Finally, described conclusions in [3, 8] is based on maximum drone acceleration, the value of which is not readily known from the specifications of a drone. *In summary, current studies have the following missing components: (i) The design-space tradeoff for the drone's computational profile and the effect of computation power on the flight time; (ii) Required computing for real-time control (inner-loop) and autonomous features (outer-loop) that is not biased toward high-speed drones, but covers the full picture and technologies; (iii) An open-source, reasonably-priced, and reproducible experimental platform with a customizable hardware-software stack is not yet available.*

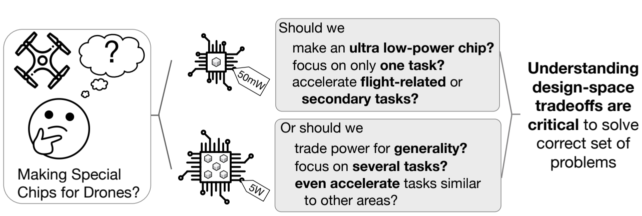

Figure 1: Impactful contributions in drones are only realized by quantifying the design-space tradeoffs.

3. Key Insights and Results

The key insight of this paper is to carry out a systematic formulization to quantify the design-space tradeoffs of autonomous drone systems by using the empirical measurements and physics to (i) explore major trade-offs across the entire *hardware-software stack*, (ii) study the *computational* profile and landscape of such systems, and (iii) connect three essential drone metrics including flight time, control response time, and autonomous features, as well as several other design-space metrics. Two examples of the metrics studied are shown in Figure 2. Figure 2a exhibits the total power consumption of 450mm drones formulized by our study and verified using data from commercial drones shown as additional data points.

Figure 2b illustrates the computation footprint for 3 W and 20 W chips, which for instance, is utilized to translate compute power efficiency to flight time by untangling the multifaceted relationships in drones. For instance, Figure 3 shows heavy computation power contribution and flight time for commer-

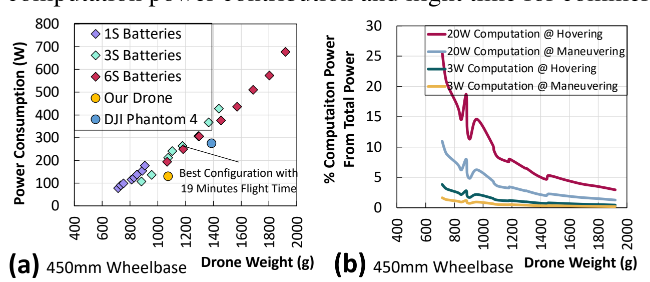

Figure 2: The total power consumption (a) and the computation footprint (b) for drones with a 450mm wheelbase.

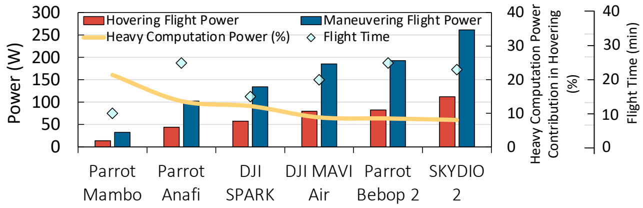

Figure 3: Heavy computation power contribution and flight time of commercial small-size drones.

cial small-sized drones.

Our studies present a few important findings. For instance, we discover that the update frequency of the inner-loop control is 50–500 Hz, which is not limited by the high-end computation power, but by the physical response time and inertia of the electromechanical components in drones.

Our findings also quantify the percentage of computation power from total power; it widely ranges from 2–30%, which potentially enables gaining +5 minutes of flight time in small drones. Finally, an example evaluation that optimizes SLAM implementation on various hardware platforms (detailed results in Table 1) shows that moving from GPU/CPU to FPGA provides 20x power savings hence extending flight time by 15–20% (+2–3 minutes) in small drones.

4. Contributions

This is the *first paper* to contribute the following:

- Formalizes the fundamental drone subsystems and quantifies the design-space tradeoffs for the computational profile of drones to discover how computation power consumption affects drone flight time, accomplished by incorporating physics and empirical measurements from 300 commercial components and 150 manufacturers.
- Clearly separates the required computing for inner-loop controls (real-time requirements) *vs.* outer-loop controls (autonomous features) in drones and outlines the required computation amount and benefits gained.
- Showcases the optimization landscape for a widely used SLAM algorithm and the effects on flight time.
- Develops an open-source and reproducible platform with a customizable hardware-software stack to address the lack of publicly available drone platforms.

5. Main Artifacts

The main artifacts of our paper are as following:

(I) We quantify design-space tradeoffs that pertain to computation power and efficiency in drones by extracting crucial metrics from over 300 commercial components and 150 manufacturers, by performing the procedure shown in Figure 5. As this procedure shows, based on the data and relationships extracted in the paper, one can accurately quantify the benefits of architecture and system optimizations to obtain gained flight time for a wide range of drones.

(II) We developed and fly tested a fully open-source experimental drone that is fully customizable across its hardware-software stack to address the lack of publicly available end-to-end experimental and reproducible drone frameworks, shown in Figure 4. Unlike popular platforms such as the CrazyFlie [2] or the PlutoX [1], which limit access to flight code and cannot carry additional payloads; our drone can be configured for a variety of research purposes, can carry additional 200g for additional embedding platforms such as FPGA boards, and grants complete access to all control systems.

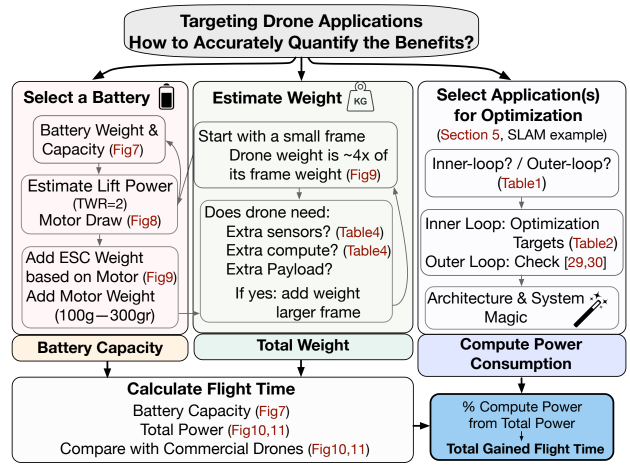

Figure 5: Quantifying compute power consumption in drones.

(III) We showcase the presented tradeoffs by evaluating design optimizations on performance and power consumption on flight time. We explore offloading SLAM, a widely-used algorithm in autonomous drones, into various hardware platforms. Table 1 lists the results of this showcase study.

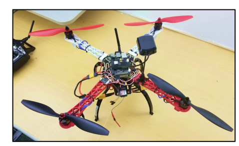

Figure 4: Our Drone.

Validation: We verify our findings by studying released flight times and battery configurations of commercial drones, shown as additional data points in graphs (similar to Figure 2). Moreover, no data skewing or pre-selection is used for extracting tradeoffs. All data points are from actual commercial components available in market (all listed in our repository). For the SLAM study, we include all benchmarks in EuRoC dataset.

6. Why ASPLOS

Our multidisciplinary paper emphasizes the synergy of architecture and system areas by the quantifying the system-level tradeoffs in autonomous drones and integrating them in architecting efficient computation platforms. Therefore, our paper will be best utilized by the broad ASPLOS community with both architecture and system backgrounds.

Table 1: Comparing costs of various platforms for SLAM.

Platform		RPi	TX2	FPGA	ASIC
SLAM Speedup		1x	2.16x	30.70x	23.53x
Power Overhead (W)		2	10	0.417	0.024
Weight Overhead (g)		≈50	≈85	≈75	≈20
Integration Cost		Low	Low	Medium	High
Fabrication Cost		Low	Low	Medium	High
Gained Flight Time (min)	Small Drones	0	≈-4	≈2–3	≈2.2–3.2
	Large Drones	0	≈-1.5	≈1	≈1

References

[1] Drona Aviation. dronaaviation.com/plutox.

[2] Bitcraze. Crazyflie 2.1. bitcraze.io/crazyflie-2-1, 2019. [Online; accessed 04/10/20].

[3] Behzad Boroujerdian, Hasan Genc, Srivatsan Krishnan, Wenzhi Cui, Aleksandra Faust, and Vijay Reddi. Mavbench: Micro aerial vehicle benchmarking. In *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 894–907. IEEE, 2018.

[4] Adam Bry, Abraham Bachrach, and Nicholas Roy. State estimation for aggressive flight in gps-denied environments using onboard sensing. In *2012 IEEE International Conference on Robotics and Automation*, pages 1–8. IEEE, 2012.

[5] Bardienus P Duisterhof, Srivatsan Krishnan, Jonathan J Cruz, Colby R Banbury, William Fu, Aleksandra Faust, Guido CHE de Croon, and Vijay Janapa Reddi. Learning to seek: Autonomous source seeking with deep reinforcement learning onboard a nano drone microcontroller. *arXiv preprint arXiv:1909.11236*, 2019.

[6] Nils Gageik, Paul Benz, and Sergio Montenegro. Obstacle detection and collision avoidance for a uav with complementary low-cost sensors. *IEEE Access*, 3:599–609, 2015.

[7] NASA JPL. Drone race: Human versus artificial intelligence. nasa.gov/jpl/drone-race-human-vs-ai.

[8] Srivatsan Krishnan, Zishen Wan, Kshitij Bhardwaj, Paul Whatmough, Aleksandra Faust, Gu-Yeon Wei, David Brooks, and Vijay Janapa Reddi. The sky is not the limit: A visual performance model for cyber-physical co-design in autonomous machines. *IEEE Computer Architecture Letters*, 19(1):38–42, 2020.

[9] Sikang Liu, Michael Watterson, Sarah Tang, and Vijay Kumar. High speed navigation for quadrotors with limited onboard sensing. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1484–1491. IEEE, 2016.

[10] Antonio Loquercio, Ana I Maqueda, Carlos R Del-Blanco, and Davide Scaramuzza. Dronet: Learning to fly by driving. *IEEE Robotics and Automation Letters*, 3(2):1088–1095, 2018.

[11] Kimberly McGuire, Guido De Croon, Christophe De Wagter, Karl Tuyls, and Hilbert Kappen. Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone. *IEEE Robotics and Automation Letters*, 2(2):1070–1076, 2017.

[12] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Showcasing real-time visual-inertial odometry. youtube.com/UZH-Robotics.

[13] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. 2017.

[14] Philemon Sakamoto. *UAV mission planning under uncertainty*. PhD thesis, Massachusetts Institute of Technology, 2006.