A Study of Drift Variability in Crazyflie Nano-Quadcopters by Pitting in-built Logging Parameters against Meticulous Measurement Techniques

Vihaan Bhaduri
Saratoga High School
Saratoga, California, USA
vihaan.bhaduri@gmail.com

Mark Santolucito
Barnard College, Columbia University
New York, USA
msantolu@barnard.edu

Abstract

Drift is one of the most unpredictable challenges when operating nano-quadcopters like Crazyflie. Caused by various factors including propeller orientation, battery positioning, and hardware manufacturing variations-drift significantly impacts flight predictability, particularly in tight spaces such as narrow hallways, crowded rooms, and public areas. While a permanent solution to drift may not exist—especially when it arises from inherent hardware variations such as mechanical inconsistencies and alignment inaccuracies—understanding its effects is crucial for flight precision, safety, and the mitigation of unintended flight path deviations. Moreover, drift analysis is essential in environments lacking precise positioning systems. Addressing drift directly within flight operations provides a more effective and broadly applicable solution to this mechanically induced issue. In this case study, we examined the typical drift patterns observed in Crazyflie quadcopters, assessing their scale and comparing differences across model versions. This analysis can help researchers determine whether these drones offer the precision required for their applications or if additional stabilization measures, such as a lower-level Proportional-Integral-Derivative (PID) controller, are necessary.

Keywords

Nano-quadcopters, Crazyflie, mechanical variability, drone drift

ACM Reference Format:
Vihaan Bhaduri and Mark Santolucito. 2025. A Study of Drift Variability in Crazyflie Nano-Quadcopters by Pitting in-built Logging Parameters against Meticulous Measurement Techniques. In *Proceedings of International Journal of Secondary Computing and Applications Research (IJSCAR VOL. 2, ISSUE 1).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.5281/zenodo.14988716

1 Introduction

Drift is a critical factor in all aspects of flight, as its impact on trajectory can compromise safety and precision. In our study, we focused on mitigating hardware- and mechanically-induced drift in the Crazyflie quadcopter series from Bitcraze—an issue possibly stemming from slight imperfections in the drone's construction. Though seemingly minor, such variations can lead to unintended deviations, increasing the risk of collisions with surrounding obstacles. As we advance toward autonomous drone operation, refining even the smallest flight details is essential to achieving an optimally designed drone. An ideal build could potentially eliminate drift, but it is very difficult to achieve. Since drift is undeniable, minimizing it is crucial for maintaining a drone's structural integrity and ensuring it can reliably complete tasks without sustaining damage. This would enable precise navigation through tight spaces, where accuracy is vital for the drone's operation. In summary, the key contributions of our work are:

(1) Graphing data from logger module vs. manually measured drift data to determine whether the logger module provided by Bitcraze is capable of generating accurate location data. The logger data proved to be quite erratic and often incorrect, and drift was undeniably visible in all models of the Crazyflie.
(2) Cross-evaluating model performance with Crazyflie 2.1+. This model performed better than Crazyflie 2.1 model in terms of drift but certainly showed its significant presence affecting all aspects of flight.
(3) Validating whether having a patterned floor would affect the quadcopter's optical position analysis module perform better, thereby reducing drift. The experiment did not yield a positive outcome.

2 Related Work

Although drift is essential in all forms of aerial flight, there is minimal research available on drift analysis and its implications on a quadcopters. A significant and manually created drift is explored in [5]. By cutting the propellers and analyzing the effects this had on the drone, the authors were able to better understand the significance of drift on a quadcopter. Another work, [6], related to drone drift in fact argues that the concept is often negligible. Built-in and pre-implemented flight algorithms were utilized to reduce the effect of drift in this paper. The paper mentions the use of IMUs (Inertial Measurement Units) when conducting short flight experiments where drift is insignificant. However, in situations where mechanical drift is high, the simple use of an IMU is problematic.

Nano-Unmanned Aerial Vehicles (UAVs) have the potential to be one of the most useful forms of autonomous drones. Whether they are used for traffic patrolling, hallway monitoring, or other purposes, they can play a significant role in shaping the future. However, their creation is not simple. Many challenges, such as memory problems and their short lifetimes, pose difficulties. In [7], the use of the Parallel Ultra-Low Power Shield (PULP-Shield) as well as the convolutional neural network, PULP-DroNet, is able to both

This paper is published under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 International (CC-BY-NC-ND 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.
IJSCAR VOL. 2, ISSUE 1
© 2025 The Author(s). Published by IJSCAR.

calculate the steering angle of the drone as well as the drone's crash probability, resulting in a functional automated nano-quadcopter.

This drone was created by adding a PULP-Shield on top of the Crazyflie 2.0's body. The PULP-Shield consists of a PULP-GAP8 processor, composed of a fabric controller and a cluster, as well as other parts. A fabric controller is similar to a microprocessor, and the cluster is used to accelerate parallel code running on the processor. The PULP-Shield's use is mainly related to input filtration. The image stream coming from the camera is offloaded to the Crazyflie's hardware, where the drone's microcontroller unit (MCU) then offloads much of the intensive visual navigation workload to the PULP accelerator. This filtered input is then sent to the PULP-DroNet model.

Although this automated nano-quadcopter is functional, it still lacks the necessary robustness needed for real-life application. As seen in the video provided in the paper, the drone is unable to keep a straight and steady path, moving at a slow pace of 1.5 m/s while maintaining a height of 0.5 m/s. Furthermore, the PULP-DroNet model is trained on an outdoor dataset, making it difficult to perform with environmental variations. In our work, we addressed this challenge by performing our experiments in an enclosed garage space.

When regarding all forms of aerial flight, flight precision cannot be trivialized. A high precision model could save battery life, increase efficiency, and reduce crash hazard. When regarding work similar to that conducted by ETH Zurich [7], drift analysis can be essential to further progress in this space. Positioning alternatives, although feasible as possible solutions, are both less effective and unreasonable in the context of experimental flight. Drone drift analysis when utilized effectively reduces flight risks. Moreover, in the new domain of swarm flight [2, 4, 8], drone drift could potentially create accidents between drones. If a quadcopter veers off its flight path, it could easily collide with other neighboring drones, leading to crashes.

A potential solution to drift mitigation is seen within the context of runtime monitoring, a technique often used to detect threats to a system. As demonstrated in [1] this research domain has huge potential in drone autonomy and flight enhancement. The authors used RTLola, a stream-based specification language that helped develop a dynamic monitoring framework for the DLR ARTIS (Autonomous Robotics and Technology for Intelligent Systems), a UAV. RTLola is both an expressive and reliable specification language, meaning it is able to analyze the health status of a UAV during flight. The use of the specification language was heavily present in the work demonstrated, helping UAVs on reconnaissance missions. Based on real-time data analysis, a UAV would make decisions centered on ensuring its safety. The aforementioned details of RTLola can possibly be utilized for direct drift correction. The safety validation system can be changed to prevent excessive displacement from the flight path. These technologies are essential to alert the system of any errors so it can correct itself.

3 Hardware Stack

The Crazyflie 2.1/2.1+ is built from a variety of different microprocessors and chips in order to create a lightweight and prototype-friendly nano-quadcopter. The essential components in use for our work were part of Bitcraze's Flow Deck V2. The navigation sensor is composed of two main components: the VL53L1x ToF sensor and the PMW3901 optical flow sensor. The former is used for precise measurement in the drone's upwards movement (traditional, '+z' axis) while the latter measures movement relative to the ground. The sensor calculates the change in pixels between frames. We collected data from flight trials on both the Crazyflie 2.1 and Crazyflie 2.1+ models in order to analyze drift. Some of the data we were able to collect also allowed us to help determine the performance capability of the optical flow sensor under specific conditions, analyzing whether a patterned or netural floor would improve the sensor's estimations.



Figure 1: Enclosed garage setup to minimize environmental factors

4 Methodology

To collect sufficient data to prove drift variability, we conducted a series of five different styles of flight experiments (in-place, forward, backward, rightward, and leftward) as shown in Fig. 2 using two different drone models: Crazyflie 2.1+ and Crazyflie 2.1. We used Crazyflie 2.1+, one of Bitcraze's latest releases, as our base model. The lightweight drone was assembled with the Flow Deck v2 attached to its underside. This deck helped the drone understand its location and generate coordinates. We used these coordinates to calculate the net drift of the drone during flight and used Crazyflie's motion commander ('mc') module for command and control.

Drift measurements were performed in an enclosed garage to minimize the effect of environmental factors such as wind and erratic air currents and external temperature variations, as shown in Fig. 1. In all our experiments, we measured between two points, a take-off and a landing point. Ideally, with zero drift, these two points should coincide for in-place (hover) flights and should follow perfect grid geometry for right and left lateral flights and forward and backward flights. We kept a distance of two meters between the takeoff and the ideal landing points in each of the four flight types, as depicted in Fig. 2.

To calculate the net drift caused by the flight, we measured the displacements 'x' and 'y' from the takeoff location to the landing position. While we performed manual measurements, the Bitcraze

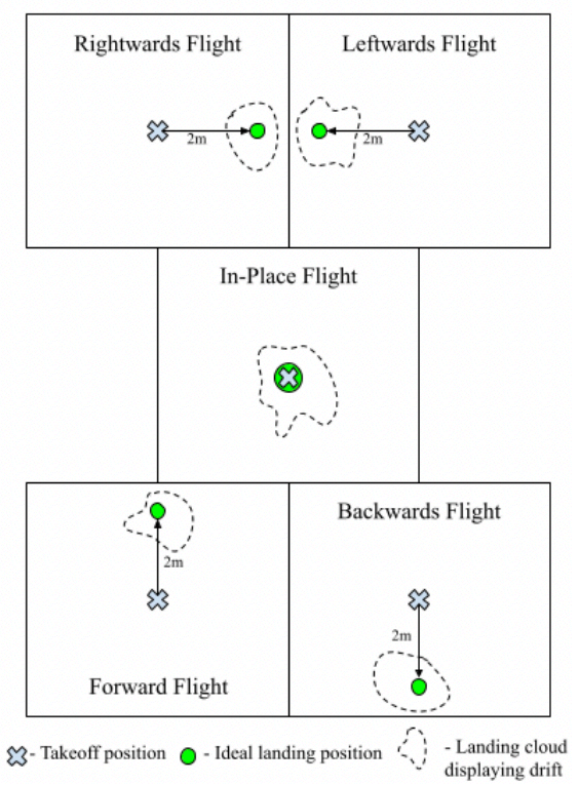


Figure 2: Five different drone flights - ideal vs actual

logger module would also record its own interpretations of drift. We used three essential log variables:

`state.position.x, state.position.y, state.position.z`

from the Bitcraze position framework to record the drone's location coordinates. This was later used to calculate drift from the logger module. We faced a challenge matching the coordinate axes between Bitcraze's logger system axes convention and the traditional coordinate axes used in our manual measurements. Our traditional coordinate system assumed the drone's front/nose aligned with the positive 'y' direction. The Bitcraze logger system differed from this convention by using unconventional axes as: right (traditional '+x' axis) → '-y' axis, forward (traditional '+y' axis) → '+x' axis, left (traditional '-x' axis) → '+y' axis, backward (traditional '-y' axis) → '-x' axis. We later adjusted the data to conform to our interpretation of a logical and traditional axes system.

We divided the flight into two categories, in-place flight and path-based flight. The in-place flight involved a simple movement:

`mc.up(1.5, velocity=1.5)`

This would elevate the drone 1.5 meters upward (traditional '+z' axis), hover for 10 seconds, and then descend. In the path-based flight, we measured drift variability while the Crazyflie was in motion across four different simple paths - lateral left, lateral right, forward, and backward. We positioned the drone to be oriented such that its direction of flight would lead it to its ideal landing spot (ex: for forward flight, we placed the drone with its nose facing forward). We followed the same flight motion as the in-place flight, raising the drone to a height of 1.5 meters, where it would hover for 3 seconds. Then we would use Bitcraze's motion commander module to fly the drone two meters in the path direction (left, right, forward, backward). The quadcopter would then finally descend.

In total, we conducted 50 drift variability tests of in-place flight using the Crazyflie 2.1+ model and 25 tests for each category of path-based flights: forward, backward, rightward, and leftward. We also performed 40 in-place flight drift variability tests on the Crazyflie 2.1 model that helped us compare the variability between models and conclude that the drift variability is undeniable and real. There was no set guidance behind choosing the 50 and 40 iteration numbers. To balance human time and effort spent conducting such meticulous experiments and measurements, we used our engineering discretion to finalize the test iteration numbers that we considered reasonable to help us empirically conclude drift variability.

Moreover, we performed an additional 25 in-place flight runs to test the impact of propeller replacement on drift. We swapped the pair of propellers, which came with the Crazyflie 2.1 bundle, with replacement propeller parts.

Lastly, we conducted tests to prove that a checkered pattern on the floor would not affect the perception of movement by the PMW3901 optical flow sensor, an integral part of the Crazyflie flow deck module. We created a nonuniform grid pattern on the floor and performed a series of 10 in-place flight tests.

5 Results

5.1 Graphical Interpretation

The results of our collected data were represented in three different ways (see Fig. 6 - Fig 13). The (a) graphs for each mini-dataset composed of a scatter plot of both the logger and manually interpreted data. Points relating to the same data are connected in the graph. The (b) and (c) graphs represent a running average plot with the blue graph displaying the x-drift while the orange graph indicating the y-drift. The (b) graphs represented the trend lines for the data collected by the logger, while the (c) graphs represented manually collected data. The inability for the trend lines to plateau emphasizes the presence of inherent variability in drift.

All outliers in our dataset were ignored while plotting the graphs; otherwise, they would make them visually unreadable. These outliers were a result of the erratic output of the crazyflie's logger module.

In-place flight is the most accurate measurement of drift, as shown in Fig. 6. It does not involve any potential confounding variables that Crazyflie's motion commander module might have. Hence, we used in-place flight as our baseline for drift comprehension and analysis.

Crazyflie 2.1+: Depicted in Fig. 6, Fig. 7, Fig. 8, Fig. 9, and Fig. 10.
Crazyflie 2.1: Depicted in Fig. 11.
Crazyflie 2.1 with new propellers: Depicted in Fig. 12.
Crazyflie 2.1+ with a patterned floor: Depicted in Fig. 13, Fig. 5.

5.2 Threats to Validity

Our experiments, despite their rigorous nature and proof, have the potential to be limiting due to small mechanical discrepancies between quadcopter builds. However, the general conclusion on

the omnipresence and potentially detrimental drift is undeniable across drones.

Internal Validity: The quadcopters that we used in our experiments could potentially have had mechanical defects. This could include motherboard variations, warped propellers, or misaligned battery installation. Despite our thorough balancing of the propellors, and precise installation of the quadcopter components, a chance of mechanical misalignment remained unavoidable that could have affected the drift slightly.

External Validity: There is no guarantee; in fact, it is nearly impossible to duplicate this study. However, a replication of this work is certainly possible. While it may not yield the exact results presented in this paper, it will likely show a similar trend and conclusion, unless it is inherently addressed in future Crazyflie models. Every quadcopter is made uniquely, and thus drift is more noticeable in some than in others. This generalization may apply to all miniature drone models, not just the Crazyflie series by Bitcraze. However, drift can potentially be mitigated, regardless of the type of quadcopter. We proposed a few options in Section 6.1 (Future Work).

5.3 Analysis

Our different flight trials gave us a varying set of data points. However, all plots indicated one key element: drift is evident in all our unique flight types - dominant in Crazyflie 2.1 and a little less dominant in Crazyflie 2.1+.

The Crazyflie logger module and the motion commander module also presented inaccurate results. The data from the logger module often deviated drastically from the ground truth values in both the 'x' and 'y' directions. The use of the logger module when attempting to control flight is problematic, as its inaccurate positional estimates would make the flight adjustment process difficult. Furthermore, the motion commander module often was unable to move the drone as expected. For instance, with the Crazyflie 2.1+, in-place (hover) flight deviated by around 10-15 centimeters in both the 'x' and 'y' directions; rightward and leftward lateral flights deviated by around 50 centimeters in the 'y' direction. Even forward and backward flights demonstrated significant drifts with an average variation of (-7.51 cm, 17.14 cm) for forward flight and (-0.34 cm, -38.40 cm) for backward flight from the ideal. Our trials for in-place flight in Crazyflie 2.1 gave us an average deviation of (42.46 cm, -53.62 cm) when taking off from (0, 0).

In addition, we witnessed an inconsistency in drift when we replaced the propellers of the Crazyflie drone as shown in Fig. 3. When the counterclockwise rotating propellers were replaced with a new pair, the average coordinates for in-place flight was (-17.30 cm, 46.93 cm) as shown in Fig. 12 graph. Thus, any form of mechanical deviation like a propeller replacement, would alter the flight trend, making the drone's flight unpredictable. This can pose a major handicap for precise navigation and flight safety. Consistency is hard to come by in the Crazyflie models. Even without a change in flight type or mechanical components, the drone would still not produce repeatable results. This can be seen by the often fluctuating running mean plots shown in Fig. 4(b).

The variability within each of the datasets is concerning and can be seen in the (b) and (c) graphs throughout Fig. 6 - Fig 13 in details. To aid visualization, a summary graph showing this variability is

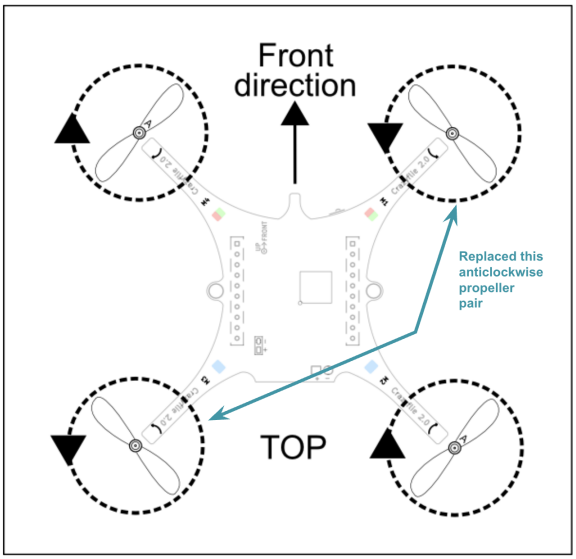


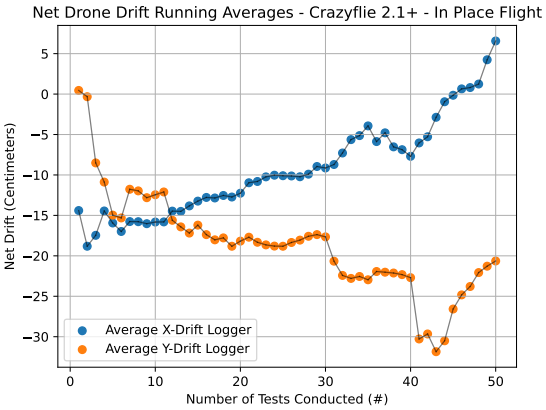
Figure 3: Replaced anticlockwise propeller pair (Adapted from Bitcraze Getting Started Guide [3])

presented in Fig. 4. The difference between flight drift indicates an inconsistency in flight. This implies that drift can affect a quadcopter in a different way each flight. For instance, one flight may cause the drone to drift by 10 cm in the rightwards direction while the next flight may cause a 10 cm drift in the leftwards direction. It is possible that a simple human touch to a propeller or to the drone's framework, altering drift's effect.

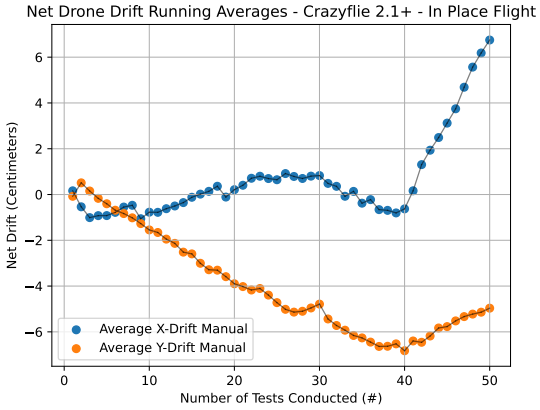
To ensure minimal to no drift and achieve precise flight behavior when the PMW3901 optical sensor of the Crazyflie's flow deck module detects a patterned or textured floor, we conducted an experiment using a custom nonuniform checkered pattern on the floor, as shown in Fig. 5. This setup allowed us to evaluate the optical sensor's effectiveness in stabilizing the Crazyflie's motion and if it could generate consistent optical flow data for better trajectory control and drift. However, the results in Fig. 13 showed no visible improvement, as drift persisted. This indicates that the checkered floor pattern did not enhance the performance of the PMW3901 optical flow sensor or effectively reduce drift in the Crazyflie's flight.

6 Conclusions

We empirically demonstrated that drift in Crazyflie quadcopters is undeniable. We also showed the inaccuracies in Crazyflie's inherent logger module and how it differed from our meticulous manual measurements. We demonstrated that by changing mechanical parts like propellors or the Crazyflie model to the latest one failed to eliminate or drastically reduce drift. Lastly, we proved that a checkered pattern on the floor yielded no noticeable reduction in drift or flight behavior. Since drift is unavoidable, it can be detrimental to flight safety and precise navigation. Hence, in our future work, we explore potential methods to combat drift and improve flight precision, safety, and reliability.



(a) Logger module: Running Average Plot



(b) Manual measurement: Running Average Plot

Figure 4: Summary plots showing variability of datasets Average Point for Logger Module Data: (6.57 cm, -20.64 cm); **Average Point for Manual Msmnt. Data:** (13.50 cm, -9.93 cm)

6.1 Future Work

This drone analysis has paved the way for critical improvements in Crazyflie drone flight trajectories. To counteract mechanically and externally induced drift, we can implement a simple program to stabilize flight. By continuously monitoring the drone's position, we can confine its movement within a predefined boundary, preventing excessive displacement. This ensures precise, stable, and controlled flight, enhancing overall performance and reliability.

Another effective way to counteract drift is by adjusting the drone's flight perspective based on a detailed drift analysis. For example, if the drone consistently drifts 'x' centimeters to the right when flying forward, a program can be designed to compensate for this deviation, ensuring a more accurate trajectory. However, this auto-correction is only viable in scenarios where the drift pattern remains consistent. For inconsistent drift, a more complex program can be designed to compensate and correct the flight path on the fly. Using any of these, the impact of drift on a quadcopter can be effectively mitigated.

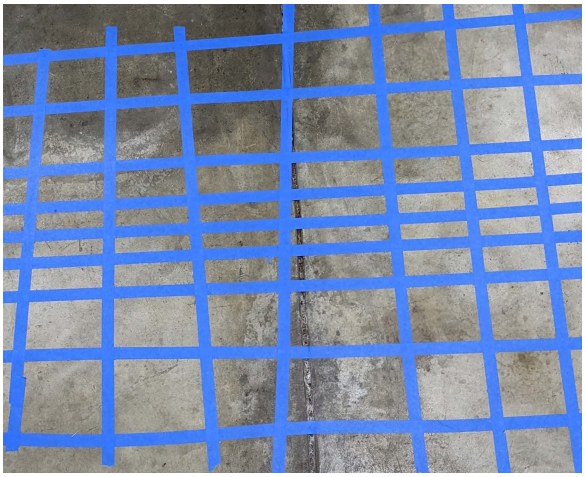


Figure 5: Nonuniform checkered pattern on floor

Furthermore, our work with Crazyflie models has opened the door to a wide range of innovative possibilities. One promising avenue is the implementation of a runtime monitoring framework like RTLola on small-scale drones, such as those used in this study. This integration could enhance safety mechanisms, allowing nano-quadcopters to operate beyond traditional boundaries while minimizing the risk of damaging expensive equipment. Additionally, the modular design of these drones makes them easily adaptable, enabling cost-effective prototyping and rapid development of new technologies.

References

[1] Baumeister, J., Finkbeiner, B., Schirmer, S., Schwenger, M., Torens, C.: Rtlola cleared for take-off: monitoring autonomous aircraft. In: Computer Aided Verification: 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21–24, 2020, Proceedings, Part II 32. pp. 28–39. Springer (2020)

[2] Biediger, D., Mahadev, A., Becker, A.T.: Investigating the survivability of drone swarms with flocking and swarming flight patterns using virtual reality. In: 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE). pp. 1718–1723. IEEE (2019)

[3] Bitcraze: Getting started with the crazyflie 2.0 or crazyflie 2.1(+), https://www.bitcraze.io/documentation/tutorials/getting-started-with-crazyflie-2-x/

[4] Lehto, M., Hutchinson, B.: Mini-drones swarms and their potential in conflict situations. In: 15th international conference on cyber warfare and security. vol. 12, pp. 326–334 (2020)

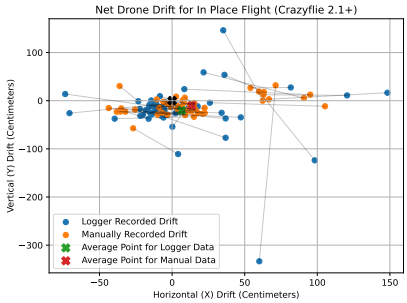
[5] Masalimov, K., Muslimov, T., Kozlov, E., Munasypov, R.: Crazypad: A dataset for assessing the impact of structural defects on nano-quadcopter performance. Data **9**(6), 79 (2024)

[6] Örnhag, M.V., Persson, P., Wadenbäck, M., Åström, K., Heyden, A.: Trust your imu: Consequences of ignoring the imu drift. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4468–4477 (2022)

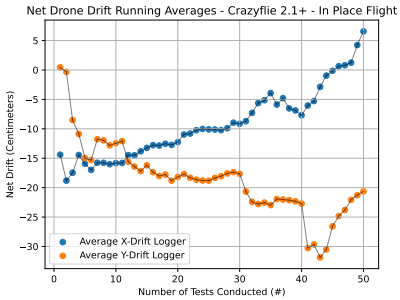
[7] Palossi, D., Conti, F., Benini, L.: An open source and open hardware deep learning-powered visual navigation engine for autonomous nano-uavs. In: 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS). pp. 604–611. IEEE (2019)

[8] Zhou, X., Wen, X., Wang, Z., Gao, Y., Li, H., Wang, Q., Yang, T., Lu, H., Cao, Y., Xu, C., et al.: Swarm of micro flying robots in the wild. Science Robotics **7**(66), eabm5954 (2022)

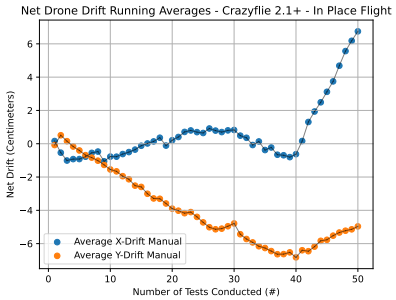
Received 06 January 2025; accepted 25 March 2025



(a) In Place Drift Plot

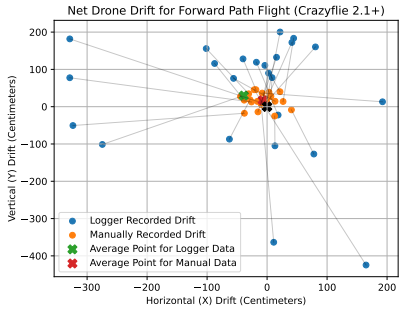


(b) Logger module: Running Avg. Plot

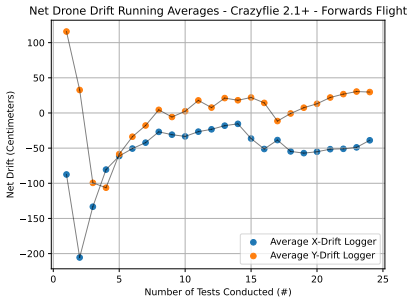


(c) Manual measurement: Running Avg. Plot

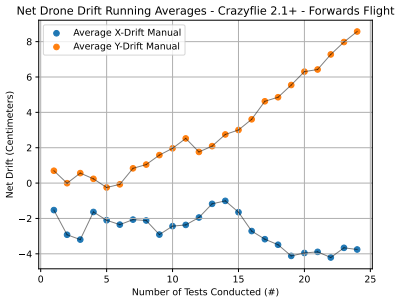
Figure 6: Crazyflie 2.1+ Plot - In-Place Flight
Outliers: None
Average Point for Logger Data: (6.57 cm, -20.64 cm); **Average Point for Manual Data:** (13.50 cm, -9.93 cm)



(a) In Place Drift Plot

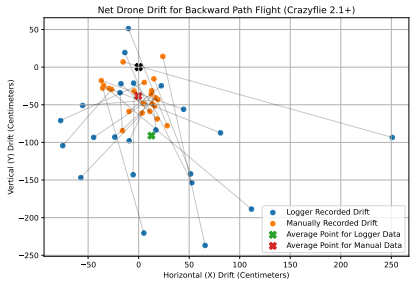


(b) Logger module: Running Avg. Plot

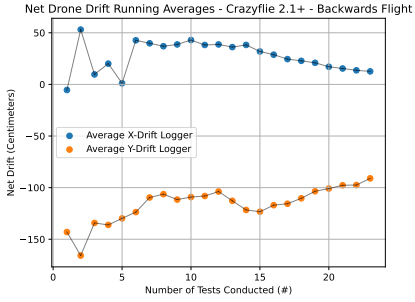


(c) Manual measurement: Running Avg. Plot

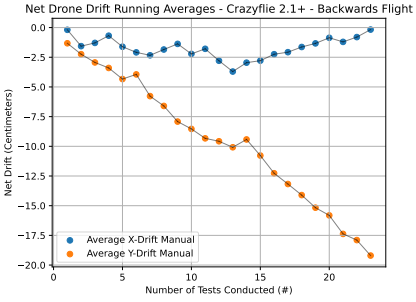
Figure 7: Crazyflie 2.1+ Plot - Forward Path Flight
Outliers: 1st Data Point Collected - (2.56 cm, 22.55 cm, 0.01 cm)
Average Point for Logger Data: (-38.86 cm, 29.74 cm); **Average Point for Manual Data:** (-7.51 cm, 17.14 cm)



(a) In Place Drift Plot

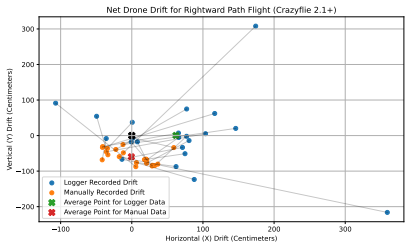


(b) Logger module: Running Avg. Plot

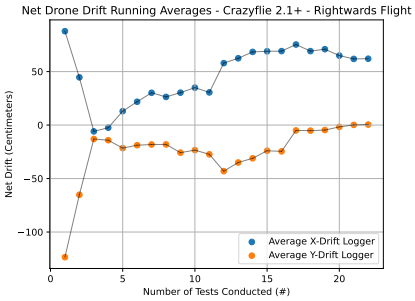


(c) Manual measurement: Running Avg. Plot

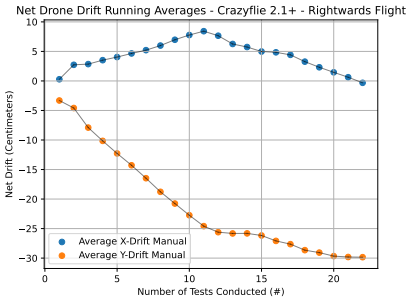
Figure 8: Crazyflie 2.1+ Plot - Backward Path Flight
Outliers: 1st Data Point: (-78.98 cm, -69.41 cm, 0.005 cm); 22nd Data Point: (-29.00 cm, 1.50 cm, 0.004 cm)
Average Point for Logger Data: (12.61 cm, -91.01 cm); **Average Point for Manual Data:** (-0.34 cm, -38.40 cm)



(a) In Place Drift Plot

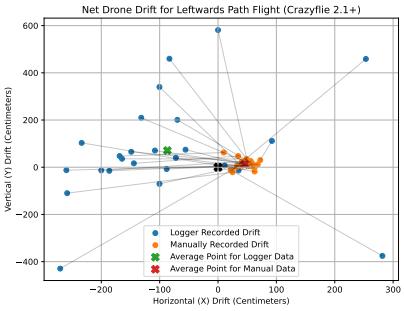


(b) Logger module: Running Avg. Plot

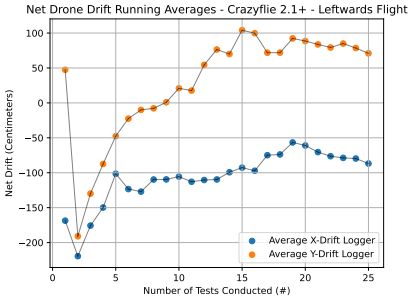


(c) Manual measurement: Running Avg. Plot

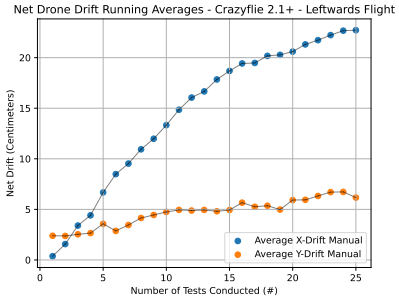
Figure 9: Crazyflie 2.1+ Plot - Rightward Path Flight
Outliers: 1st Data Point: (4.80 cm, 0.03 cm, 0.01 cm); 6th Data Point: (-5.71 cm, 1.98 cm, 0.008 cm); 12th Data Point: (-7.06 cm, 3.35 cm, 0.01 cm)
Average Point for Logger Data: (62.04 cm, 0.46 cm); **Average Point for Manual Data:** (-0.63 cm, -59.69 cm)



(a) In Place Drift Plot

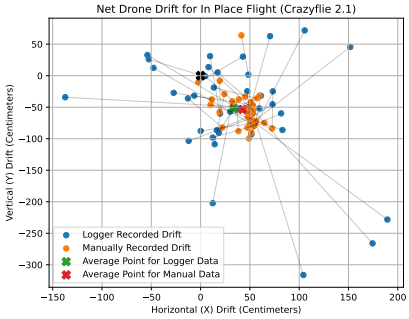


(b) Logger module: Running Avg. Plot

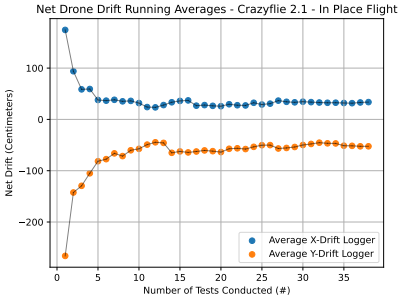


(c) Manual measurement: Running Avg. Plot

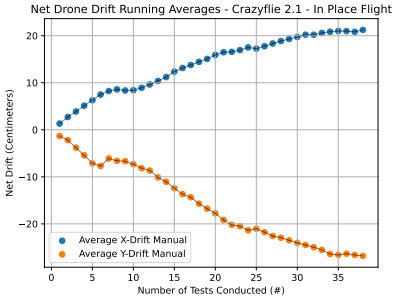
Figure 10: Crazyflie 2.1+ Plot - Leftward Path Flight
Outliers: None
Average Point for Logger Data: (-86.81 cm, 70.99 cm); **Average Point for Manual Data:** (45.43 cm, 12.33 cm)



(a) Drone Drift Plot

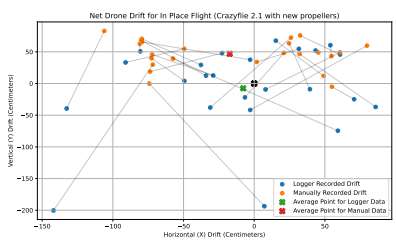


(b) Logger module: Running Avg. Plot

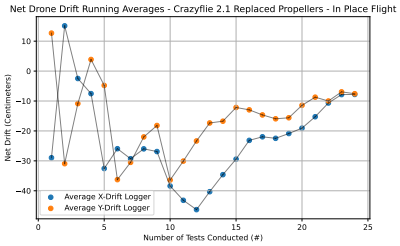


(c) Manual measurement: Running Avg. Plot

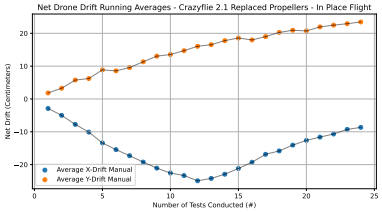
Figure 11: Crazyflie 2.1 Plot - In-Place Flight
Outliers: 26th Data Point: (0.26 cm, 0.53 cm, -3.07e-05 cm); 36th Data Point: (-15.18 cm, -33.78 cm, 0.004 cm)
Average Drift for Logger Data: (33.72 cm, -52.43 cm); **Average Point for Manual Data:** (42.46 cm, -53.62 cm)



(a) Drone Drift Plot

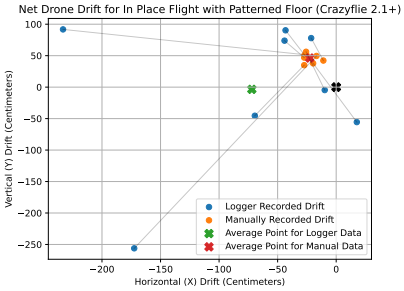


(b) Logger module: Running Avg. Plot

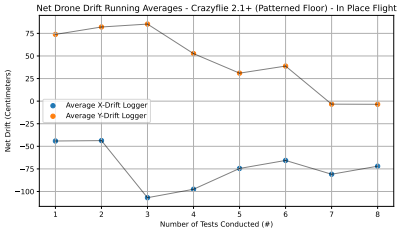


(c) Manual measurement: Running Avg. Plot

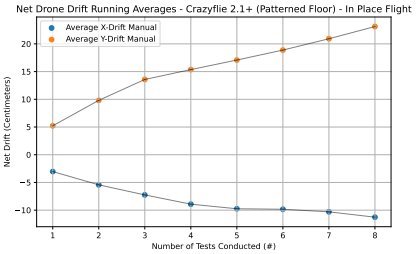
Figure 12: Crazyflie 2.1 with Replaced Propellers Plot - In-Place Flight
Outliers: 3rd Data Point: (-80.70 cm, 83.47 cm, 0.004 cm)
Average Point for Logger Data: (-7.80 cm, -7.57 cm); **Average Point for Manual Data:** (-17.30 cm, 46.93 cm)



(a) Drone Drift Plot



(b) Logger module: Running Avg. Plot



(c) Manual measurement: Running Avg. Plot

Figure 13: Crazyflie 2.1 with Patterned Floor Plot - In-Place Flight
Outliers: 3rd Data Point: (-4.86 cm, -3.92 cm, 0.01 cm); 6th Data Point: (-5.28 cm, 3.75 cm, 0.01 cm)
Average Point for Logger Data: (-55.83 cm, -104.25 cm); **Average Point for Manual Data:** (-21.10 cm, 42.976 cm)