# 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 Santoluctio
Barnard College, Columbia University
New York, USA
msantolu@barnard.edu

## Abstract

Drift is one of the most unpredictable factors to deal with when operating with nano-quadcopters. Caused by a variety of reasons—propeller orientation, battery positioning, motherboard defects—drift is detrimental to flight predictability, especially when attempting to navigate tight spaces such as narrow hallways, populated rooms, and crowded public spaces. Although drift may not have a permanent solution, especially if its source is inherently embedded in hardware variations like mechanical variability and alignment inaccuracies, understanding its effects on a drone is essential to deterministically combat flight path alterations. Furthermore, drift analysis is essential in scenarios where a precise positioning system is not available for use. Addressing drift directly within flight itself is a more effective and generalized solution to a mechanically induced difficulty. In this case study, we explored the typical drift seen in the Crazyflie quadcopters, analyzing its scale and cross-comparing the differences between version models. This analysis can help researchers determine whether the precision of these drones is sufficient for the research they will be conducting, or whether a lower level PID controller will be required.

## Keywords

Nano-quadcopters, Crazyflie, mechanical variability, drone drift

**ACM Reference Format:**
Vihaan Bhaduri and Mark Santoluctio. 2024. 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. 1, ISSUE 1)*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.5281/zenodo.14279494

## 1 Introduction

Drift is critical with respect to all aspects of flight. Its ability to alter trajectory makes it a factor that is necessary to combat in order to fly safely. In our study, we focused on addressing hardware and mechanically-caused drift within the Crazyflie quadcopter series from Bitcraze. This is a deviation that stems from non-uniformity within the build of the drone itself. Albeit a simple deformity, drift can cause damage to a drone through inadvertent contact with surrounding obstacles. In proceeding towards the intent of drone automation, it is critical for minor flight details to be perfected in order to create an optimally built drone. An ideal build would fly without drift, allowing for navigation through narrow regions where precise flight is critical for drone survival. Overall, a reduction of drift is essential when regarding a drone's health and its ability to successfully perform a task without any bodily damage.

In summary, we identify the key contributions of our work as:

(1) Graphing logger vs. manual drift data to determine whether the logger system provided by Bitcraze is capable of properly creating location data. The logger data proved to be quite erratic and oftentimes incorrect, and drift was undeniably visible in all models of the Crazyflie.
(2) Cross-evaluating model performance to see that the Crazyflie 2.1+ seemed to have the least net drift when referring to all aspects of flight.
(3) Attempting to see whether changing the texture/composition of the floor would effect the flow deck's position analysis of the quadcopter hence reducing drift: the floor composition failed to remedy drift.

## 2 Related Work

Although drift is essential to all forms of aerial flight, there is minimal work available on drift analysis and its implications on a quadcopter. A more significant and manually created drift is explored in [4]. 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 related to drone drift in fact argues that the concept is oftentimes negligible. In [5], built in and pre-implemented flight algorithms were utilized in order to reduce the effect of drift. 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-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 difficult problems. In [6], the use of the PULP-Shield as well as the convolutional neural network, PULP-DroNet, is able to both calculate the steering angle of the drone as well as the drone's crash probability, resulting in a functional automated nano-quadcopter.


*IJSCAR VOL. 1, ISSUE 1*

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 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 on 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 its ability to perform in certain settings quite difficult. However, this work could be improved with the use of implementing a drift correction. This would remove a confounding variable from the issue, resulting in a more effective model with improved speed and analysis efficiency.

When regarding all forms of aerial flight, flight precision is not something that can be overlooked. A high precision model could save battery life, increase efficiency, and reduce crash hazard. When regarding work similar to that conducted by ETH Zurich, drift analysis can be essential to furthering progress conducted 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 mitigates the potential of risk. Furthermore, in the new domain of swarm flight [2, 3, 7], drone drift could potentially create accidents between drones. If a quadcopter was to veer off its flight path, it could easily collide with other drones around it, leading to crashes.

A potential solution to drift mitigation is seen within the concept of runtime monitoring, a technique often used to detect threats to a system. As demonstrated in [1] the domain has heavy potential in drone autonomy and flight enhancement. The work used RT-Lola, a stream-based specification language that helped develop a dynamic monitoring framework for the DLR ARTIS, 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 advance a direct correction of drift. The safety validification system can be altered to prevent excessive displacement from the flight path. These technologies are essential to alert the system of any errors so it can correct itself.

## 2.1 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 out of two main elements, the VL53L1x ToF sensor and the PMW3901 optical flow sensor. The former is used for precise measurement in the drone's upwards movement while the latter measures movement relative to the ground. The sensor calculates the change in pixels between frames. Hence, when flying over neutrally colored surfaces, movement in the Crazyflie might occur differently. This phenomenon is something we will address later on in the paper. 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.

# 3 Methodology

In order to get a sufficient set of data for the experimentation process, we conducted a series of five different styles of flight upon two different drone models. We used the Crazyflie 2.1+ as our base test models, one of Bitcraze's latest releases. The light-weight drone was assembled with the Flow Deck v2 attached to its underside. This deck allows for the drone to understand its location and create estimates. We used these estimates to calculate the net drift of the drone during flight. All of our work utilized the Crazyflie's motion commander system.

Our experiment system consisted of two points, a takeoff and landing point. We measured out a distance of two meters to split the two. In order to calculate the net drift caused by the flight, we would measure the 'x' and 'y' displacement from the takeoff location. While we conducted manual calculations, the Bitcraze logger system 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 position framework in order to get a recording of the drone's location which we would use to calculate drift. All measurements that were manually recorded followed a different set of coordinate axes than the inbuilt Bitcraze axes. We used a traditional coordinate system in the respective orientation when assuming the drone's front was placed facing the positive 'y' direction. The Bitcraze logger measured movement in the following manner using a slightly unconventional set of axes: right (typically +x axis) → -y axis, forward (typically +y axis) → +x axis, left (typically -x axis) → +y axis, backward (typically -y axis) → -x axis. We would alter the data such that it conformed to our interpretation of a logical axes system.

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

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

This would elevate the drone 1.5 meters upwards where it would then hover for 10 seconds. The quadcopter would then descend. The other category we divided flight into was path flight. We conducted a set of four simple experiments to test the capabilities of the different forms of motion. We positioned the drone to be oriented in a manner such that its direction of flight would lead it to its ideal landing spot (ex: for forward flight, we positioned the drone with its front facing forwards). We followed the same flight motion as the in-place flight, raising the drone up to a height of 1.5 meters where it would promptly hover for 3 seconds. Then we would use the motion commander to fly the drone two meters in the direction we were testing drift upon. The quadcopter would then finally descend.

In total, we ran 50 tests of in-place flight calculations on the Crazyflie 2.1+ and 25 tests for each category of path flight: forward, backward, rightward, and leftward. We ran 40 tests for in-place flight on the Crazyflie 2.1 model in order to compare the pure drift, not affected by the erroneous and confounding variables that the motion commander would introduce through traversing flight.

Furthermore, we ran an additional series of 25 in-place flight runs to test the detrimental effects of propeller replacement. We replaced the pairs of propellers which come with the Crazyflie 2.1 bundle with replacement parts. We swapped the counterclockwise rotating propellers and ran 10 test runs on this modified quadcopter.

# 4 Results

## 4.1 Graphical Interpretation

The results from our collected data was represented in three different manners.

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 on the graph.

The (b) and (c) graphs represent a running average plot with the blue displaying x-drift while the orange indicated y-drift. The (b) graph represented the trend lines for logger collected data while the (c) graph was for manually collected data. The inability for some of the trend lines to plateau emphasize the effect of variability present within drift itself.

All outliers from our dataset were discounted within the plots due to their ability to warp the graphs. These points were a result of poorly calculated logger data.

In-place flight is the most accurate measurement of drift, as shown Fig. 1. It does not involved any potential confounding variables that a motion commander might impact. Hence we used in-place flight to base our typical comprehension of drift.

**Crazyflie 2.1+:** View Fig. 1, Fig. 2, Fig. 3, Fig. 4, and Fig. 5 in the Appendix.

**Crazyflie 2.1:** View Fig. 6 in the Appendix.

**Crazyflie 2.1 with new propellers:** View Fig. 7 in the Appendix.

**Crazyflie 2.1+ with a patterned floor:** View Fig. 8 in the Appendix.

## 4.2 Threats to Validity

Our experiments, despite their rigorous nature and proof, have potential to be faulty.

**Internal Validity:**

The quadcopters we used in our experiments could potentially have had mechanical defects. This could include an incorrectly balanced motherboard, warped propellers, or a misaligned battery. Despite our thorough balancing of the propellers and of the quadcopter, the potential of a mechanical alteration could have affected the drift.

**External Validity:**

There is no guarantee, in fact it is nearly impossible, for a replication of this study. Every quadcopter is made uniquely and hence drift is more noticeable in some than others. This generalization applies for all miniature drone models, not just the Crazyflie series by Bitcraze. However, drift has the potential to be mitigated, regardless of the type of quadcopter.

## 4.3 Analysis

Our different flight trials gave us a varying set of datapoints. Yet all plots indicated one key element: drift is evident throughout all of our unique flight types, especially visible in the Crazyflie 2.1.

The Crazyflie logger system as well as the motion commander also presented concerning results. The data from the logger often deviated heavily from the ground truth values in both the 'x' and 'y' directions. The use of the logger system when attempting to control flight is problematic, as its inaccurate positional estimates would make the flight adjustment process difficult. Furthermore, the motion commander often was not able to move the drone in the manner expected. For instance, rightward and leftward flight with the Crazyflie 2.1+ deviated by around 50 centimeters in the 'y' direction. For in-place flight, the drone deviated by around 10-15 centimeters in both the 'x' and 'y' directions. Even forward and backward flight demonstrated significant drifts with an average point of (-7.51, 17.14) for forward flight and (-0.34, -38.40) for backward flight.

In addition, we were able to see an undeniable inconsistency in drift when we replaced the propellers of the drone. Our trials for in-place flight on the Crazyflie 2.1 gave us an average point of (42.46, -53.62) when taking off from (0, 0). When the counterclockwise rotating propellers were replaced as shown in Fig. 7, the new average point for in-place flight was (-17.30, 46.93). This handicap proves concerning when regarding the Crazyflie as a drone that can be used for precise navigation and flight. Any form of mechanical deviation upon the quadcopter such as propeller replacement would result in an altered flight trend, making the drone's flight unpredictable. Consistency is hard to come by in the Crazyflie models. Even without a change in flight type nor mechanical components, the drone would still not produce repeatable results. This can be seen by the often fluctuating running mean plots shown in the Results section.

The variability within each of the datasets, as indicated in the (b) and (c) graphs, is also another point of concern. These trends can be visualized in Fig. 1 (b, c), Fig. 2 (c), Fig. 3 (c), Fig. 4 (c), Fig. 5 (b, c), and Fig. 7 (c). The difference between flight drift indicates an inconsistency in flight. This implies that drift can affect a quadcopter in a different manner per flight. However, it is possible that a simple human touch to a propeller or to the drone's framework could differ drift's effect.

We also ran experiments in order to test whether the PMW3901 optical flow sensor would alter the drift and correct flight better given a more textured/patterned floor. We created a rough checkerboard with random variability in order to test this theory. However, the results produced a consistent drift with no visible improvement. Hence there was no evidence that the texture and pattern of the floor would effect the Crazyflie's flight.

# 5 Conclusions

We conclude by exploring potential future work regarding drift and how it can be combated. Furthermore, we discuss techniques to improve quadcopter safety.

### 5.1 Future Work

This drone analysis has opened the door for fixes to flight trajectory for Crazyflie drones. In order to combat for mechanically and externally induced drift, we can use a simple box program to fix flight. Depending on the position of the drone, we can ensure the quadcopter will stay withing a certain box. Its displacement should not exceed the box's ranges, ensuring stable and accurate flight.

Another potential way to combat drift is to alter the drone's perspective of flight based on a thorough drift analysis. For instance, if the drone naturally drifts 'x' centimeters to the right when flying forward, a program could be coded such that the drone's new forward flight will account for this drift. If this can be achieved with the given of consistent drift, the implication of drift upon a quadcopter could be negated.

Furthermore, our work with the Crazyflie models has allowed us to explore a vast domain of potential ideas. For example, the potential of implementing a runtime monitoring framework such as RTLola on a small-scale drone like the Crazyflie models used in this paper offers great potential. It would allow for a nano-quadcopter to explore beyond existing safety mechanism boundaries and reduce the risk of destroying expensive equipment. Such a drone would be easily modifiable allowing for cost-effective prototyping.

## 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] 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)

[4] 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)

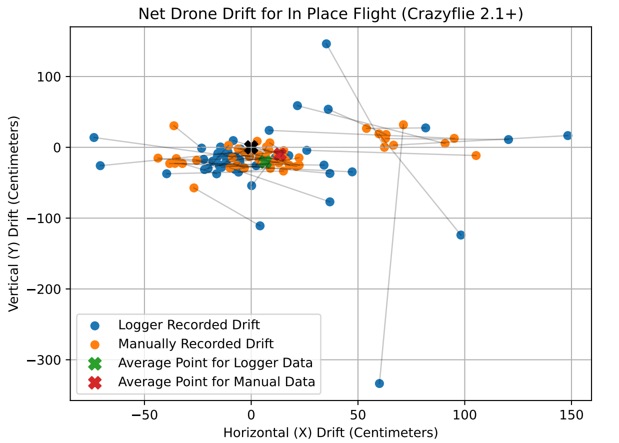
[5] Örnhag, M.V., Persson, P., Wadenbäck, M., Åström, K., Heyden, A.: Trust your imu: Consequences of ignoring the imu drift. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4468–4477 (2022)

[6] 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)

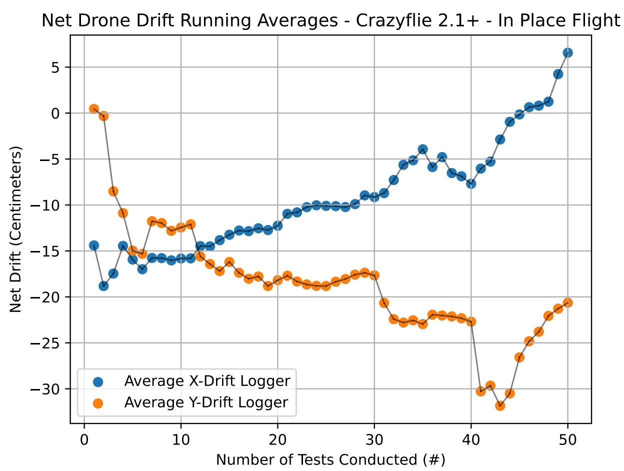
[7] Zhou, X., Wen, X., Wang, Z., Gao, Y., Li, H., Wang, Q., Yang, T., Lu, H., Cao, Y., Xu, C., et al.: Swarm of micro flying robots in the wild. Science Robotics **7**(66), eabm5954 (2022)

Received 20 February 2024; revised 12 March 2024; accepted 5 June 2024

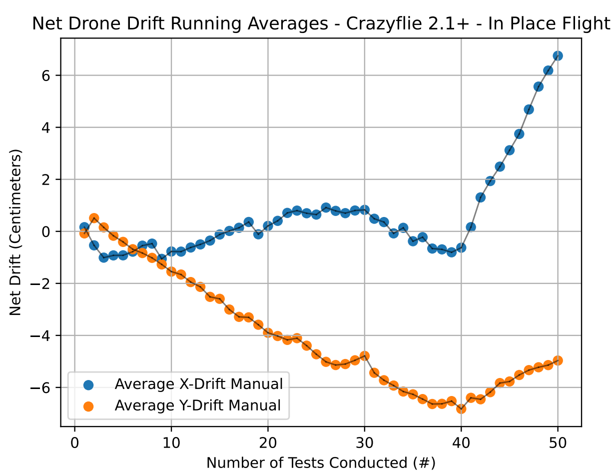
# Appendix




**((a)) In Place Drift Plot**



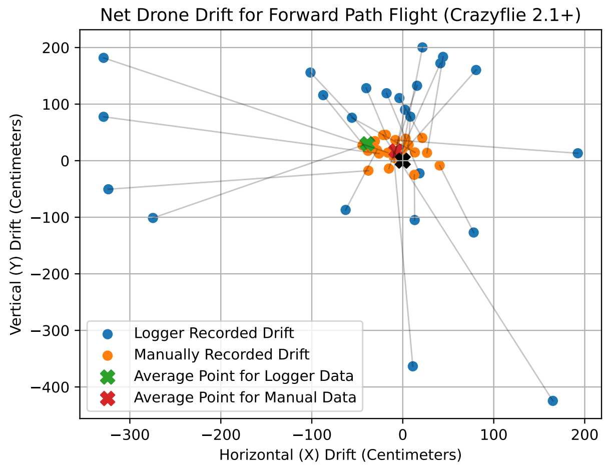

**((b)) Logger Running Avg. Plot**



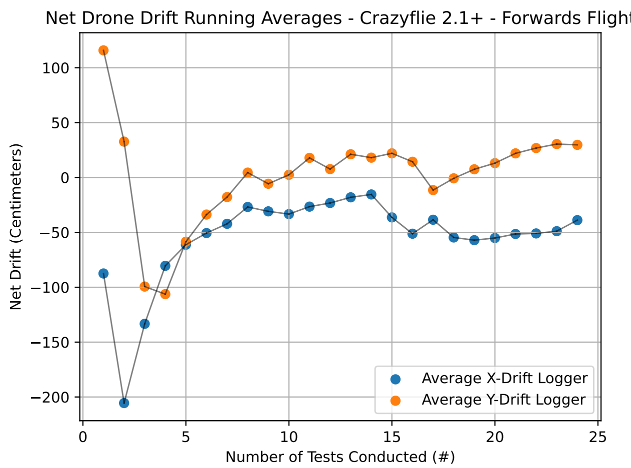

**((c)) Manual Running Avg. Plot**



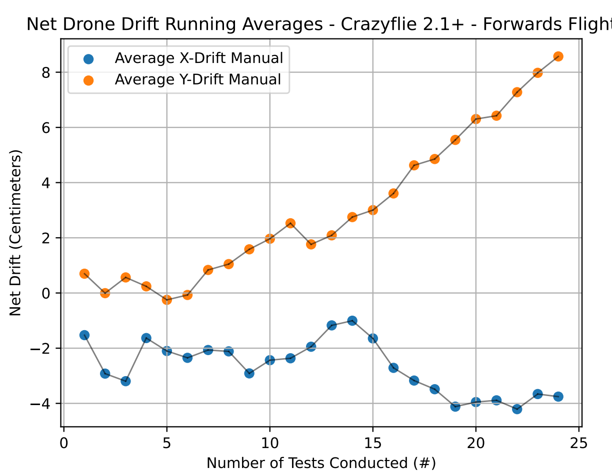
**Figure 1: Crazyflie 2.1+ Plot - In-Place Flight**
**Outliers:** None
**Average Point for Logger Data:** (6.57, -20.64)
**Average Point for Manual Data:** (13.50, -9.93)




**((a)) In Place Drift Plot**

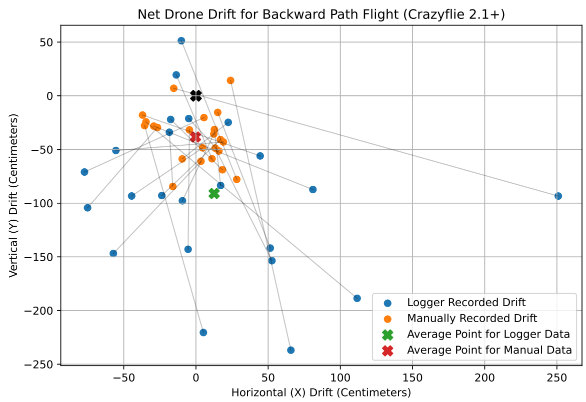



**((b)) Logger Running Avg. Plot**

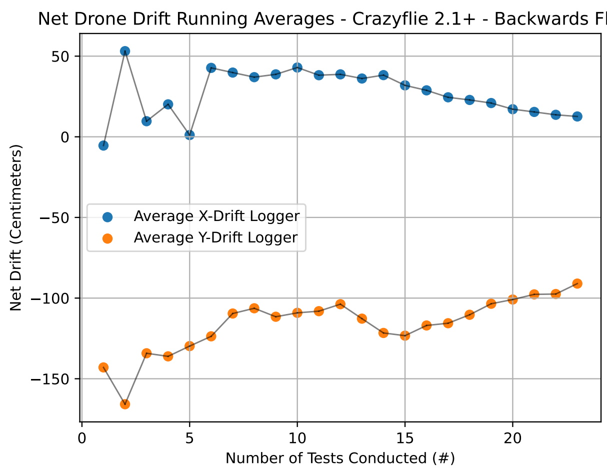



**((c)) Manual Running Avg. Plot**

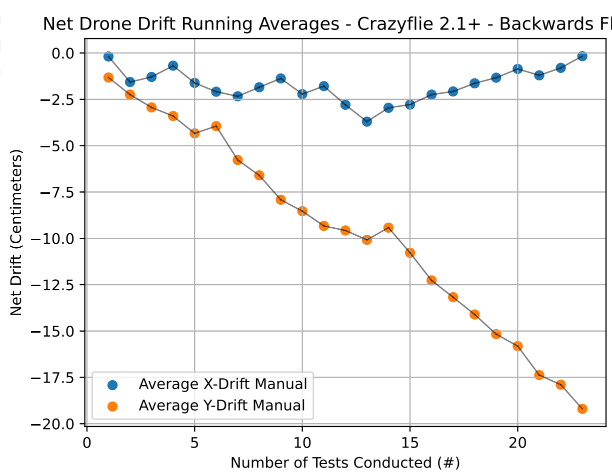
**Figure 2: Crazyflie 2.1+ Plot - Forward Path Flight**
**Outliers:** 1st Data Point Collected - (2.56, 22.55, 0.01)
**Average Point for Logger Data:** (-38.86, 29.74)
**Average Point for Manual Data:** (-7.51, 17.14)




**((a)) In Place Drift Plot**

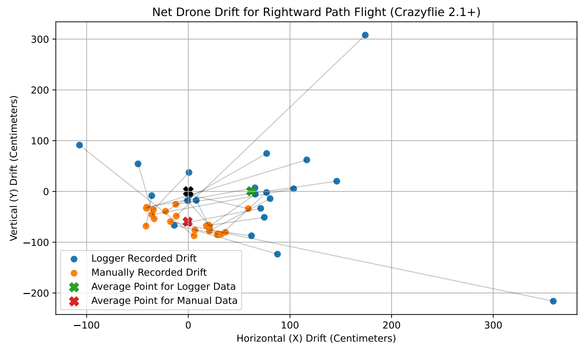



**((b)) Logger Running Avg. Plot**

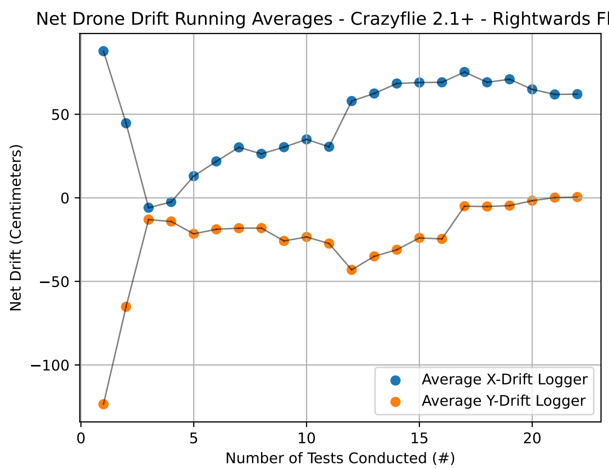



**((c)) Manual Running Avg. Plot**

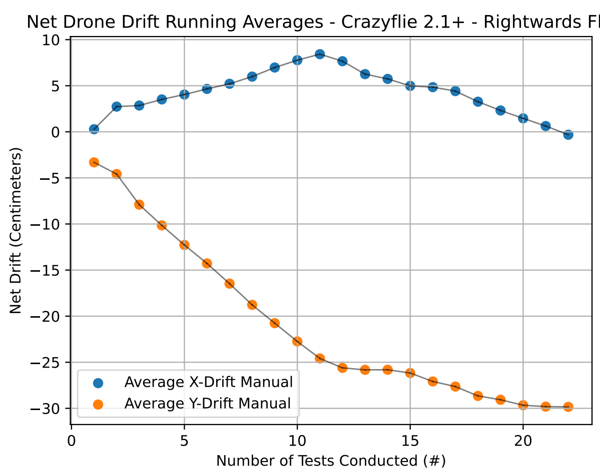
**Figure 3: Crazyflie 2.1+ Plot - Backward Path Flight**
**Outliers:** 1st Data Point: (-78.98, -69.41, 0.005); 22nd Data Point: (-29.00, 1.50, 0.004)
**Average Point for Logger Data:** (12.61, -91.01)
**Average Point for Manual Data:** (-0.34, -38.40)




**((a)) In Place Drift Plot**

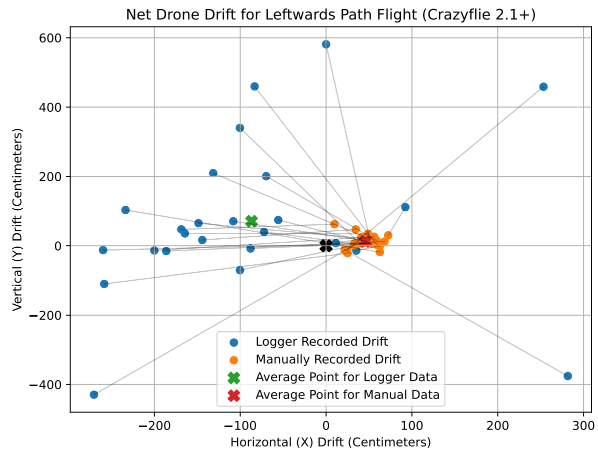



**((b)) Logger Running Avg. Plot**

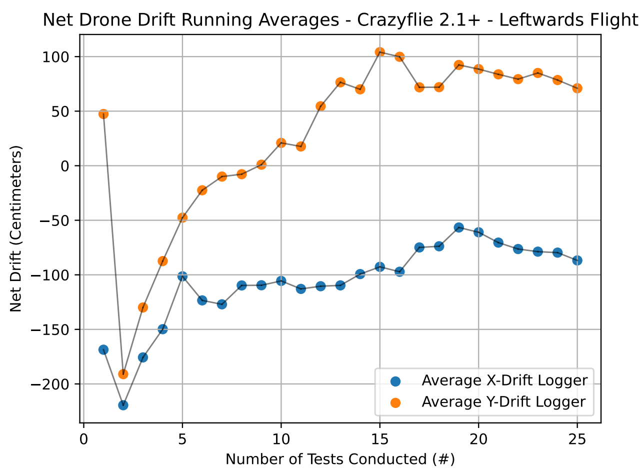



**((c)) Manual Running Avg. Plot**

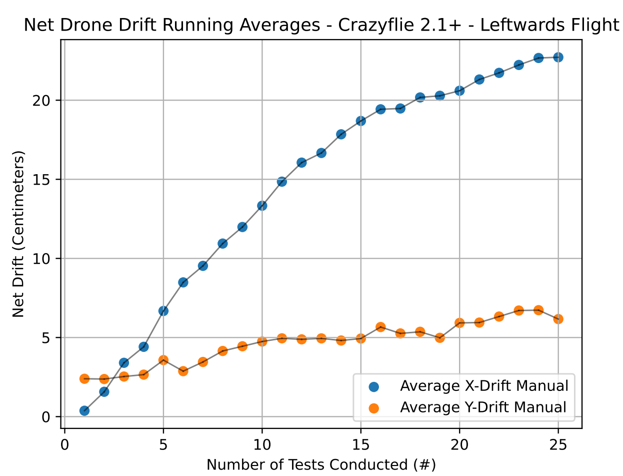
**Figure 4: Crazyflie 2.1+ Plot - Rightward Path Flight**
**Outliers:** 1st Data Point: (4.80, 0.03, 0.01); 6th Data Point: (-5.71, 1.98, 0.008); 12th Data Point: (-7.06, 3.35, 0.01)
**Average Point for Logger Data:** (62.04, 0.46)
**Average Point for Manual Data:** (-0.63, -59.69)




**((a)) In Place Drift Plot**

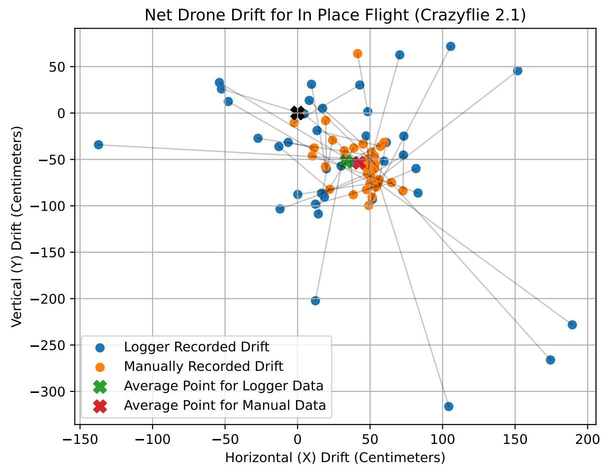



**((b)) Logger Running Avg. Plot**

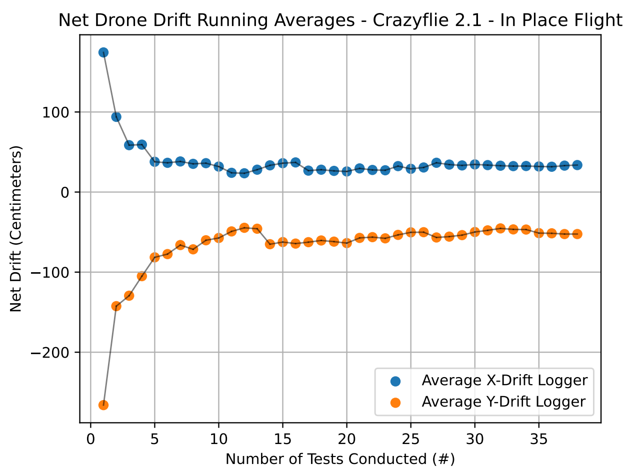



**((c)) Manual Running Avg. Plot**

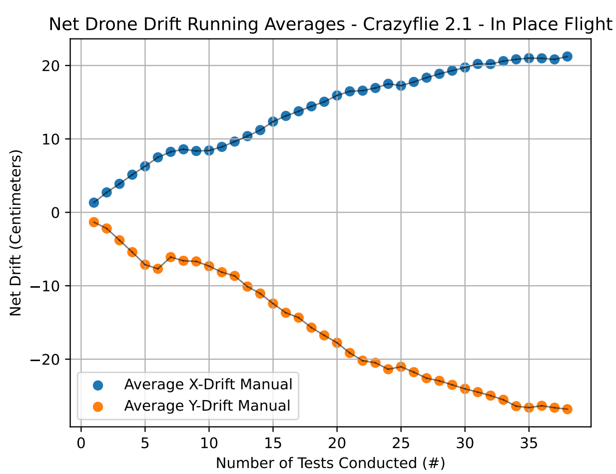
**Figure 5: Crazyflie 2.1+ Plot - Leftward Path Flight**
**Outliers:** None
**Average Point for Logger Data:** (-86.81, 70.99)
**Average Point for Manual Data:** (45.43, 12.33)




**((a)) Drone Drift Plot**



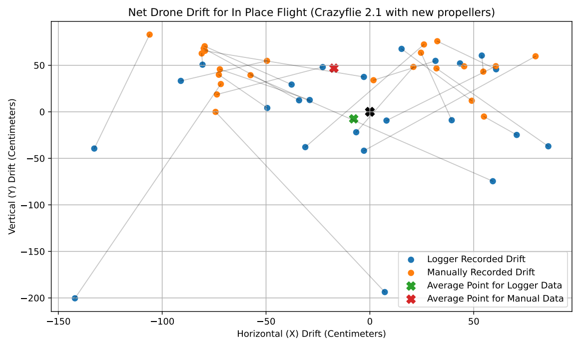

**((b)) Logger Running Avg. Plot**



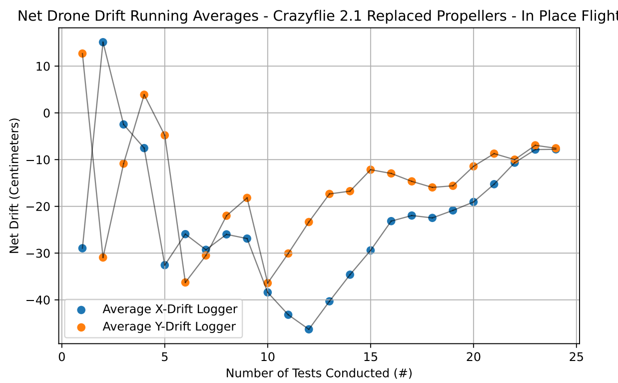

**((c)) Manual Running Avg. Plot**



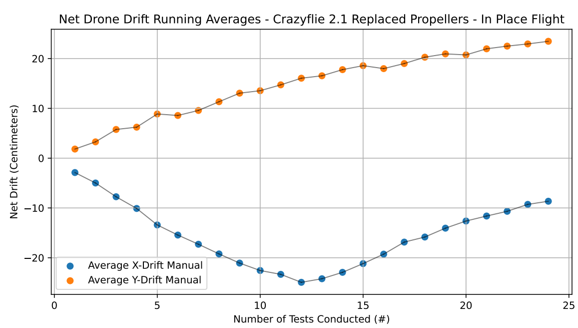
**Figure 6: Crazyflie 2.1 Plot - In-Place Flight**
**Outliers:** 26th Data Point: (0.26, 0.53, -3.07e-05); 36th Data Point: (-15.18, -33.78, 0.004)
**Average Drift for Logger Data:** (33.72, -52.43)
**Average Point for Manual Data:** (42.46, -53.62)




**((a)) Drone Drift Plot**

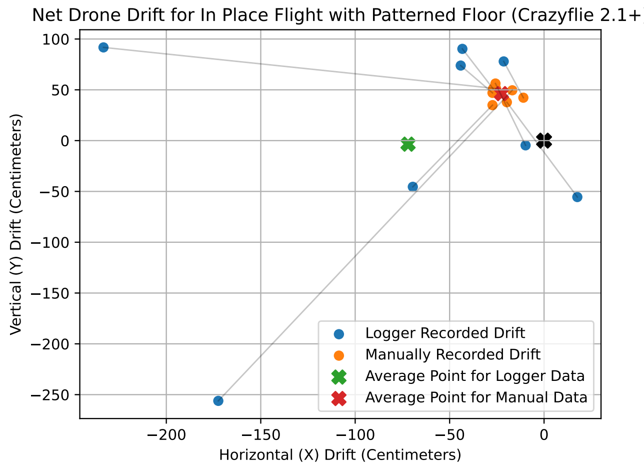



**((b)) Logger Running Avg. Plot**

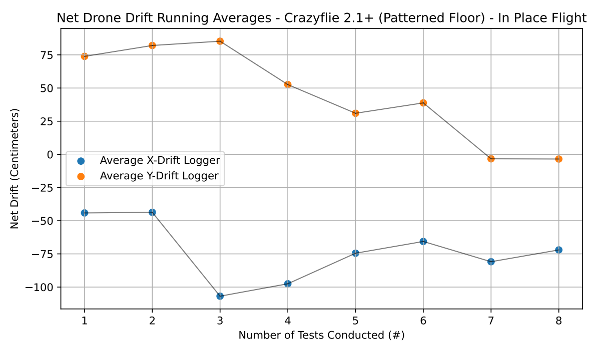



**((c)) Manual Running Avg. Plot**

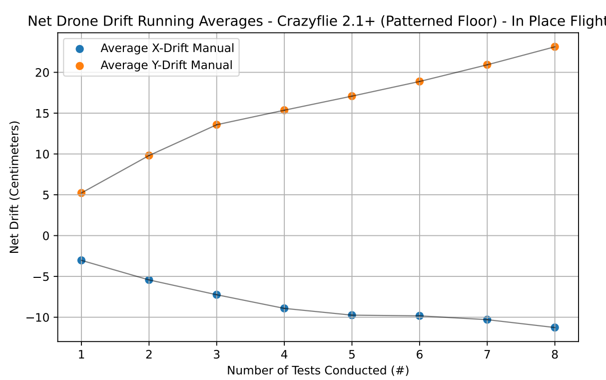
**Figure 7: Crazyflie 2.1 with Replaced Propellers Plot - In-Place Flight**
**Outliers:** 3rd Data Point: (-80.70, 83.47, 0.004)
**Average Point for Logger Data:** (-7.80, -7.57)
**Average Point for Manual Data:** (-17.30, 46.93)




**((a)) Drone Drift Plot**




**((b)) Logger Running Avg. Plot**




**((c)) Manual Running Avg. Plot**

**Figure 8: Crazyflie 2.1 with Patterned Floor Plot - In-Place Flight**
**Outliers:** 3rd Data Point: (-4.86, -3.92, 0.01); 6th Data Point: (-5.28, 3.75, 0.01)
**Average Point for Logger Data:** (-55.83, -104.25)
**Average Point for Manual Data:** (-21.10, 42.976)