Implementing monopod perching on a nano quadcopter

Francisco Gómez Medina

Abstract

bla

Contents

List of Figures

List of Tables

1 Introduction

Small-sized drones ,UAVs (Unmanned Aerial Vehicles) and UASs (Unmanned Aerial Systems) currently face two major challenges which affect their applicability and practicality: Battery life and choice of landing surface. Battery life poses a considerable limitation in terms of flight time, inevitably setting a limit to the amount of data gathered during each flight operation, with flight times rarely exceeding 20-30 minutes. As in helicopters, these vehicles also require flat, stable and sufficiently wide surfaces for landing, and in many cases this can become an issue in applications such as rainforest canopy research or natural disaster zone reconnaissance operations.

A significant amount of attempts at solving these issues involve a bio-inspired solution known as perching: When flying beings such as birds or insects do not need to stay airborne, they are able to land in locations such as power lines, posts or branches where they can rest and perform other tasks in the meantime. [1] compiles a comprehensive array of perching techniques currently investigated, which include chemical adhesion, suction cups, synthetic gecko skin, claws, microspines and electroadhesion. The Biomimetics & Dexterous Manipulation Laboratory at Stanford University have an extensive research record on the use of microspines to successfully achieve perching in ceilings and vertical surfaces in microquadrotors [2] and small fixed-wing UAVs [3]. Harvard University's Robobee [4] makes use of electrostatic adhesion to allow an insect-sized drone to perch on a variety of surfaces. However, this technology has only been currently proven to work on very small-sized systems and scalability to larger machines might be compromised by large power requirements. Worth noting as well are the works of [5] on the development of an avian-inspired passive perching mechanism and [6] for its insect counterpart, which, although more lightweight, only offers limited landing location flexibility.

The vast majority of the exposed projects share a common flaw: They depend to some extent on the available landing surface area, necessary for mechanisms such as microspines to be satisfactorily deployed, which could render them ineffective in situations where a sufficiently large landing spot is not available. This limitation is mitigated in [5] by using a bird-like claw mechanism, but at the added cost of a heavier assembly which could severely affect active flight time in small-sized UAVs. Owing to this, it would be desirable to develop a platform light enough to be carried by a small-sized drone without significantly affecting flight time and manoeuvrability and which allowed it to perch on locations such as edges or other complex geometries where deployment of claw-like mechanisms would be difficult.

A recent study investigated the feasibility of a novel technique termed "monopod perching" on a 3 degree-of-freedom Quanser helicopter for a single (pitch) axis of rotation with cascaded PID attitude control [7]. With this method the weight of the drone is held by a single lightweight rod (figure 1), allowing the drone to perch on a single point while its motors operate at minimal power for balancing. This form of active perching would obviate the need for a sufficiently smooth/rough landing surface, and the quality of perching would be unaffected by the surface area of the landing location. Furthermore, yawing about its support axis for reorientation and/or data gathering in different directions would be possible, becoming a useful feature in reconnaissance missions.

The aim of this paper is focused on the implementation of this technology on a Crazyflie 2.0 nano quadcopter, studying its feasibility and characterising its behaviour as a function of monopod length. The performance metrics will involve measurement of perching and hovering time, disturbance rejection and assessment of stability on various surfaces and geometries. The re-

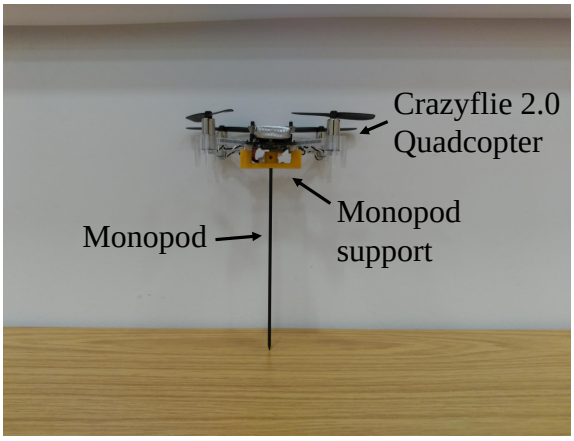

Figure 1: Monopod perching drone

sults obtained in this project will shed light on the capabilities and scalability of this approach, allow for the creation of PID tuning guidelines and aid identification of possible limitations.

2 Experimental set-up and testing

The Crazyflie 2.0 nano quadcopter was used as the test platform for this project. Its small size and high modularity made it an attractive choice, as well as its open-source code, which allows for extensive modifications to suit each application. A Crazyradio PA was used to communicate with the quadcopter, and development was performed in a Bitcraze virtual machine, built specifically for development of this quadcopter and freely available online. This drone has a flight time of only 7 minutes while hovering, which poses a serious limitation to its practicality in mission-based or research-based scenarios.

2.1 Dynamic model of drone

A dynamic model of the drone was developed to be compared against experimental results and identifying sensible monopod lengths. A nano quadrotor can be depicted as a central mass joined via massless and infinitely stiff arms to its motors. The monopod is modelled as an infinitely stiff, massless rod of length L, pivoting on a single point without friction. The maximum angle θ of inclination sustained on any axis (pitch/roll) for a given monopod length and quadcopter geometry will be achieved when the quadcopter is oriented such that all motors move within the axis's plane of rotation. Therefore, this angle will be dependent on the maximum thrust achievable by any pair of adjacent motors. Symmetry of the vehicle allows for a simplified 2D depiction of the dynamic model (Figure 2).

Analysing moments about the perching point, p, the maximum inclination angle can be found by adopting a condition of equilibrium, where $\ddot{\theta} = 0$ (equation 1):

$$\begin{aligned} mgasin(\theta + \theta_1) + MgLsin(\theta) + mgasin(\theta - \theta_1) \\ -T_1asin(\theta_1) + T_2asin(\theta_1) = 0 \end{aligned} \tag{1}$$

Equilibrium at maximum tilt angle will take place when one pair of motors is operating at full

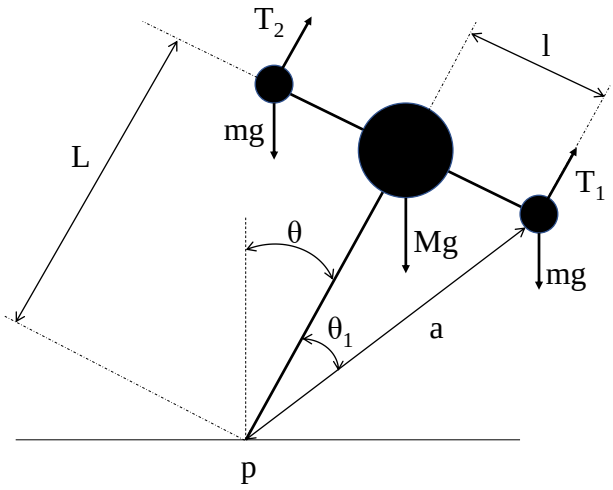

Figure 2: Dynamic model of perching drone. T_1, T_2 and m represent thrust and mass from each pair of motors respectively, L is length of the monopod, M mass of the quadcopter, l motor arm length and θ inclination angle

thrust while the opposite pair remains at zero power. For consistency with figure 2, T_2 will be set to 0, and T_1 to full power. By considering the trigonometric identity

$$sin(A+B) + sin(A-B) = 2sin(A)cos(B) \tag{2}$$

and rearranging, the maximum tilt angle θ_{max} can be found:

$$\theta_{max} = sin^{-1}\Big(\frac{T_1 l}{2mgL + MgL}\Big) \tag{3}$$

The parameters T_1, l, m and M were experimentally determined, and were recorded in table 1. Figure 3 shows θ_{max} as a function of monopod length L.

T_1 [N]	l [m]	M [kg]	m [kg]	g [ms^{-2}]
0.05625	0.035	0.032	0.004	9.8

Table 1: Experimentally determined dynamic parameters

In an effort to maintain the system sufficiently resilient to external disturbances, a maximum tilt angle of at least 2 degrees was decided upon. In order to maintain the infinitely stiff and massless monopod assumptions relevant, a maximum monopod length was assigned such that its weight would account for $<5\%$ of the weight of the drone, and would not deflect more than 5 mm under cantilever loading with a mass equal to the weight of the drone. This set the maximum monopod length at $\approx$0.2 m. Monopod lengths of 1,2,3,4,6,8,10,12,14,16,18 and 20 cm were subsequently tested.

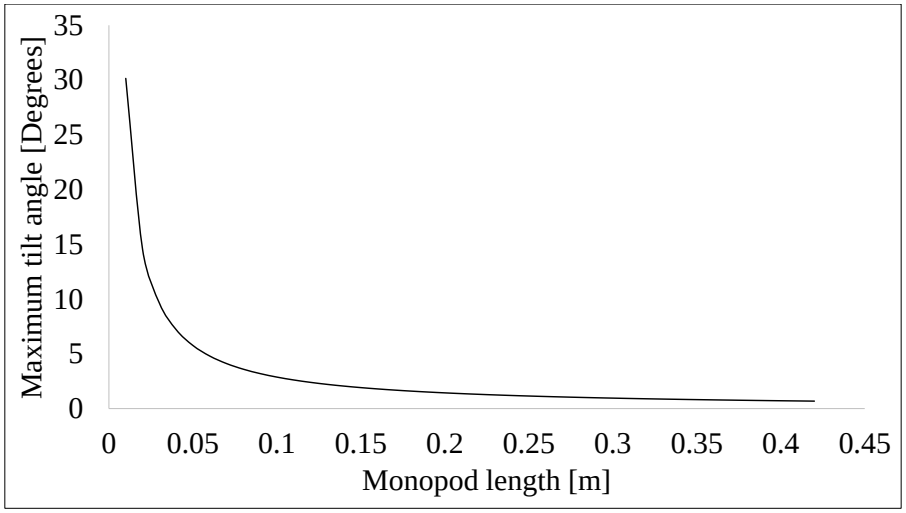

Figure 3: Maximum tilt angle θ_{max} as a function of monopod length

2.2 Control structure and perching implementation

The Crazyflie 2.0 nano quadcopter implements a cascaded PID control with Kalman filtering to respond to setpoints requested by the user. This consists of an outer loop running at low frequency which calculates the error between the desired and measured angle and outputs a desired angular rate to the inner loop. This inner loop is updated at a higher frequency and calculates the motor voltage required to induce a certain turning moment on the quadcopter, which translates into an angular rate which is compared against the desired angular rate. Figure 4 illustrates the cascaded PID loop implementation.

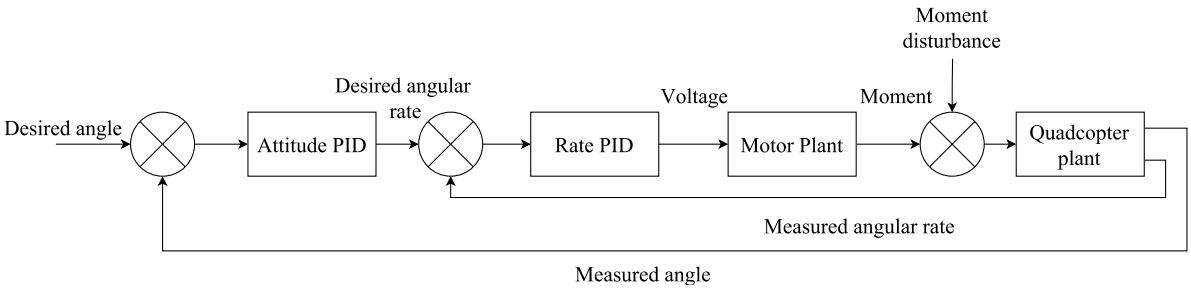

Figure 4: Cascaded PID controller on nano quadcopter

Three independent cascaded PID control loops are used to control roll, pitch and yaw requests. The motor voltage requests from each axis are decoupled and linearly added together, allowing each PID controller to be independently tuned from the other two.

A PID controller takes the general form [8]

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) + K_d \frac{de(t)}{dt} \tag{4}$$

Where K_p, K_i, K_d are constant, non-negative gains, $e(t)$ is the error and $u(t)$ the output of the controller. A PID controller with constant gains can be used in linear, or weakly non-linear systems to closely approach optimal control. [7] found in his study that a PID controller was suitable for implementation of monopod perching on a 1-axis experiment, as the tilt angles are

generally very small ($<14°$), and thus small angle approximations are applicable, allowing for linearization of the plant. However, furher non-linearities could arise from the motor plant, especially when rotating at low speeds, where the mapping between thrust and voltage may become non-linear.

The PID tuning approach will involve adjusting each axis separately. A typical approach CITE HERE involves first tuning the inner control loop for a zero angular rate reference until the quadcopter remains irrotational, but not necessarily upright. Finally, the outer loop will be tuned such that the quadcopter can be brought to the upright position, or to the correct yaw angle. This method proved ineffective for this application and its implications are further assessed in the discussion.

2.3 Experimental set-up

A carbon fibre monopod was used to minimise weight impact while ensuring sufficient stiffness. A 3D-printed ABS plastic support was used to couple it to the quadcopter (Figure 5(a)). The tip of the monopod was sharpened with a rotary tool to ensure single-point perching (Figure 5(b)). Care was taken to place the monopod underneath the center of gravity (CoG) of the quadcopter to minimise any imbalances. Any imbalances were corrected by shifting the position of the battery.

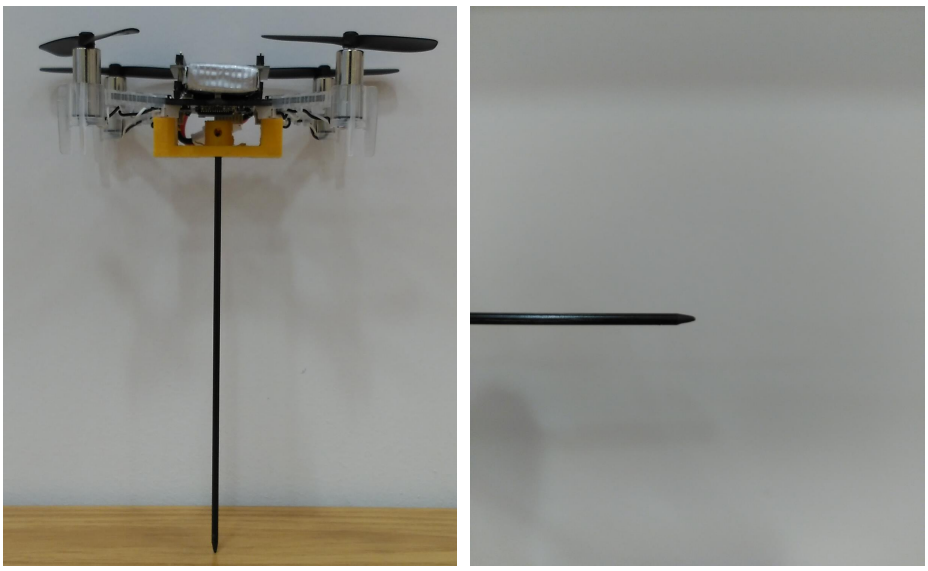

Figure 5: (a) Monopod assembly. (b) Detail of sharpened monopod tip

A 1 DOF test rig was designed and built to adjust the three PID controllers (Figure 6 (a)). A high-precision, low-friction bearing was used to minimise damping effects which would lead to excessively unresponsive PID gains, and the pivoting point was located at ground level to simulate true perching conditions. The damping on this test rig, together with the tuning method, was ineffective and a new test rig (Figure 6(b)) and tuning method (see section 3) were used. The new hand-held test rig removes the friction component from the bearings and minimises air flow disruption by incorporating a long and thin handle. A small slot was created in an aluminium plate to avoid slip of the monopod.

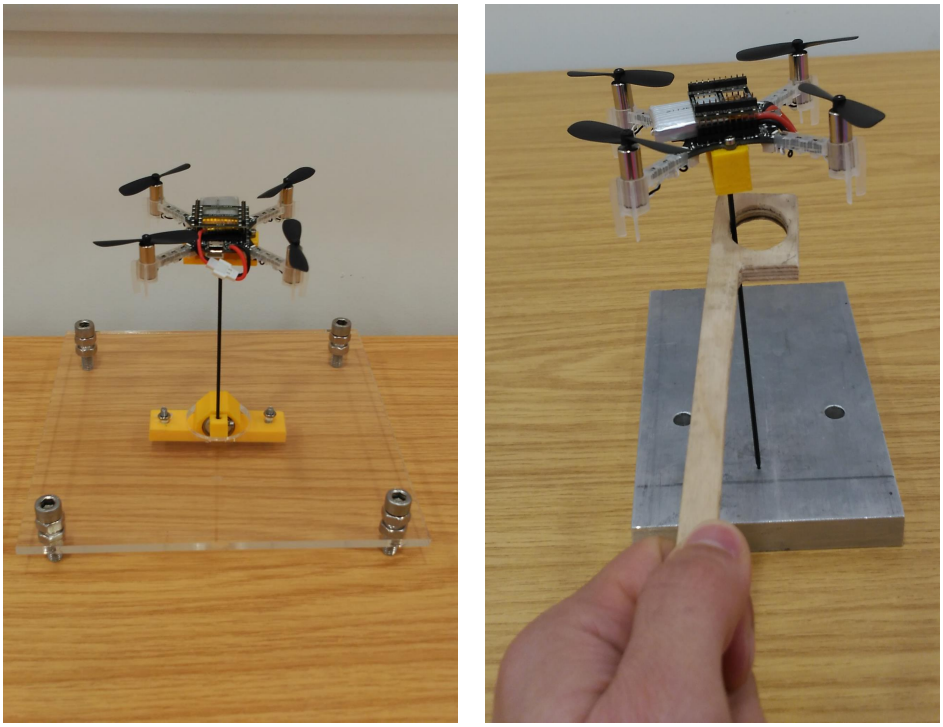

Figure 6: (a) Initial test rig. (b) Improved test rig

3 Results and discussion

3.1 PID controller tuning

One of the most challenging aspects of this project involved determining suitable PID parameters for each monopod length: The very low mass and therefore inertia of the quadcopter made it very sensitive to small changes in CoG positioning and damping, greatly affecting the repeatability of results.

As previously explained, a 1 DOF test rig was initially employed to adjust roll, pitch and yaw PIDs. The following procedure was performed for each axis: Tune rate PID until quadcopter stayed in position and then tune attitude PID to maintain a desired angle. The base thrust component was kept at zero value, since the weight of the drone is already held by the monopod and the base thrust does not contribute to stabilisation.

This procedure resulted in K_p and K_i gains of the order of 10^5 for the rate PID, and very low gains for the attitude PID, as well as K_d gains. The yaw PID, however, yielded much lower values. All parameters for a 8 cm monopod are compiled in table 2.

A strange behaviour was observed when the quadcopter was located on various surfaces to assess the efficacy and robustness of perching: On surfaces with significant damping, such as human skin, the quadcopter would remain consistently upright. On slippery surfaces, it would drift along the surface plane while remaining balanced. However, when the drone was located on surfaces with very low friction and where it was not possible to drift, it would instantly fall. This phenomenon was initially believed to be due to a slightly incorrect location of the CoG. However, this hypothesis was later disproved by performing balancing experiments on the 1 DOF test rig, and noting that the quadcopter fell with a different orientation each time. Figure

Rate PID								
Pitch			Roll			Yaw		
K_p	K_i	K_d	K_p	K_i	K_d	K_p	K_p	K_p
3200	50	0.01	3200	50	0.01	2500	30	0
Attitude PID								
Pitch			Roll			Yaw		
K_p	K_i	K_d	K_p	K_i	K_d	K_p	K_p	K_p
3.4	0	0.05	3.4	0	0.05	9	0.1	0

Table 2: PID parameters for a 8 cm monopod

7 (a) and (b) illustrate the behaviour under 1 DOF test on test rig and test on slippery surface, respectively. Figure 8 shows test results on low friction surface with restricted movement on surface plane.

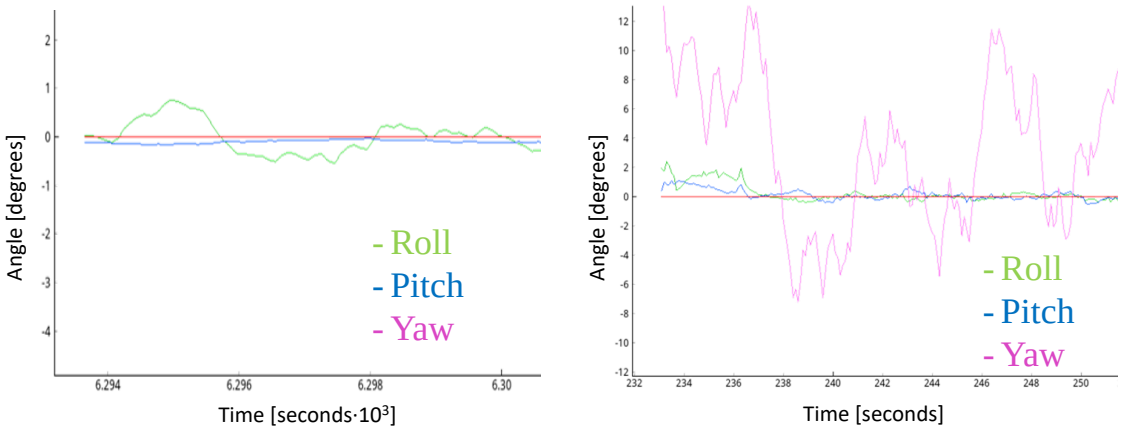

Figure 7: Roll, pitch and yaw angles on: (a) 1 DOF test on rig, (b) slippery surfaced

A second hypothesis was devised, attributing this instability to the unusually high gains found, and, more crucially, the tuning method and test rig used. It was observed that the motors would most of the time spin either at zero or full speed, even for inclinations as small as 0.01° (Figure 9(a)). This, and the fast and violent vibrations about 0°, suggested that the PID gains found were far from optimal. An assessment on the damping of the 1 DOF test rig was performed by letting it freely fall with no thrust input, and this was compared to a free fall without test rig (Figure 9(b)). The results obtained suggested that the 1 DOF test rig was significantly modifying the dynamics of the system, and a second test rig was designed to overcome these limitations (Figure 6(b))

The most obvious limitation of this new test rig was that 1 DOF PID adjustment was not possible. However, since the controllers for each axis are decoupled, the symmetry of the quadcopter allowed for both roll and pitch control loops to be tuned simultaneously using the same parameters for both axes. The yaw controller also had to be enabled to avoid uncontrolled spinning of the quadcopter, which would eventually unsettle it and fall. A desirable tuning goal was to balance the quadcopter using mainly proportional gains to avoid integrator build-up in cases

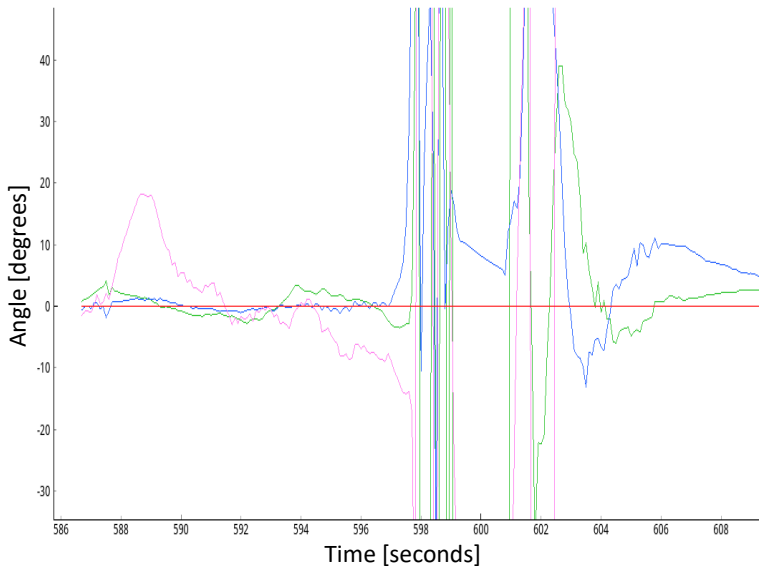

Figure 8: Roll, pitch and yaw angles on low friction surface with restricted movement on surface plane

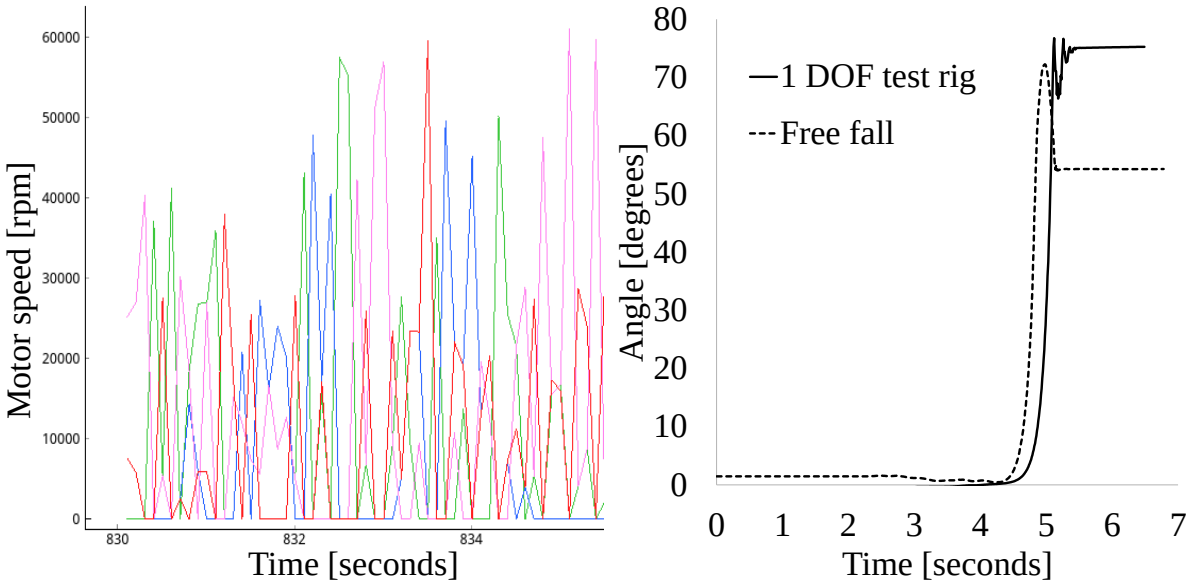

Figure 9: (a) Motor speed plot, (b) Free fall comparison on 1 DOF test rig and no test rig for damping assessment

of sustained disturbances, which would cause the drone to remain unstable for a certain time until this term decreased sufficiently. This was the case with the old gains, and many times the controllers would have to be fully reset after a fall due to the very large integral gain that had accumulated.

After extensive trials, it was found that balancing the quadcopter on rate mode was not possible: The K_p and K_i terms were so large that motor saturation would cause them to simply turn either at zero or full speed, as was previously the case, but still fall.

A different approach was used and the attitude PID gains were given a value of 1 each to aid in balancing. This allowed for much lower PID parameters to be determined, with an order of

10^3 for roll and pitch, subsequently resulting battery consumption. Final PID gains for a 8 cm monopod are compiled in table 2.

While performing preliminary endurance tests to assess how long perching could be sustained for on a single charge, another strange phenomenon was observed: On some occasions, the drone would suddenly become highly unstable and fall, with no influence from external disturbances or presence of imbalances in the drone. Figure 10(a) illustrates this behaviour: preceding the instability, no external disturbances have affected the quadcopter, yet the pitch and roll angles grow unbounded. This was initially attributed to sudden connectivity losses between the on-board controller and the pc client. However, subsequent tests verified that this was not the cause of falling. Further balance and external disturbance verifications were performed, but once again were found not to be the causes.

The cause of falling was finally attributed to the yaw controller: Yaw control was thought to be only necessary to avoid uncontrolled spinning, so a very rough adjustment, with oscillations of $\pm 20^\circ$ was initially performed to focus attention on data gathering mainly related to pitch and roll. The unsettling was believed to happen when all pitch, roll and yaw were at their maximum amplitudes at the same time: The yaw controller would exert a fast turning moment, causing the quadcopter to quickly rotate about its monopod axis, changing roll and pitch angles a the same time. These fast changes would not allow the motors to respond fast enough to these changes due to their own inertia, eventually causing the drone to fall. The yaw PID was then readjusted, and an oscillation of $\pm 0.5^\circ$ was found to be small enough to eliminate this phenomenon.

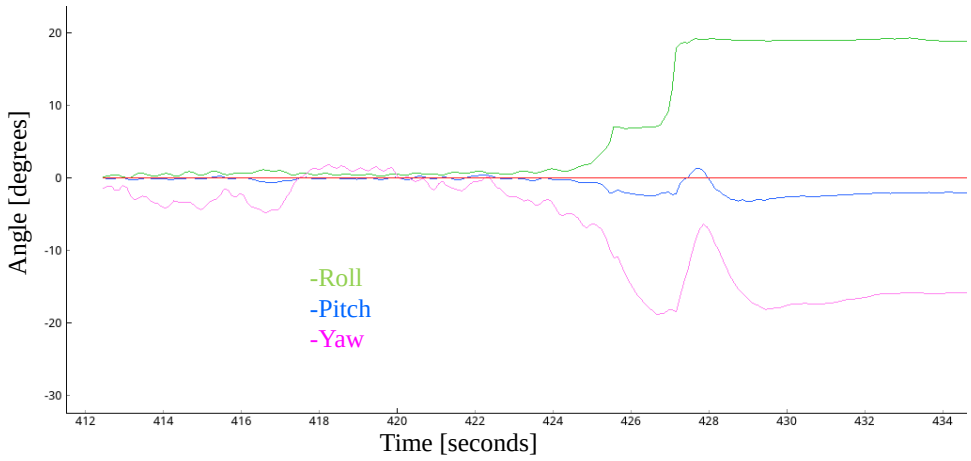

Figure 10: Sudden unsettling of quadcopter

3.2 Characterisation and analysis of perching quadcopter

Once the correct PID parameters were found for each monopod length, a study of the properties and behaviour for each length was performed. To aid in future scalability studies with larger or smaller quadcopters, the dimensionless variable L/ℓ (ratio of monopod to motor arm length. In this paper this will be referred to as the moment arm ratio) was used instead of monopod length.

One of the most important matters in perching involves measuring how much longer mission time can be prolonged: A series of tests were performed to measure perching time for each monopod length (Figure 11(a)). The hover time for the non-instrumented Crazyflie 2.0 quad-

copter was measured at 6 minutes 45 seconds (405 seconds). The experimental results indicate perching times between 40 minutes (2400 seconds) and 63 minutes (3780 seconds), which constitutes a significant 500% and 800% increase in operation time with respect to simple hovering. The longest perching times (5 trial average) were obtained for moment arm ratios of 0.57 and 0.85, both yielding 63 minutes. By fitting a trendline around the measured datapoints, it can be inferred that the optimal moment arm ratio lies between these two values. If lower ratios were used, a steep decrease in performance was observed. This could be due to the high ground effect turbulence, causing a more frequent unsettling of the quadcopter due to recirculation of air underneath the rotors and counterbalancing the reduction in thrust and requiring a larger power draw.

A much gentler loss is observed for larger ratios, which seem to approach an asymptote at about 40 minutes. This could in part be attributed to a decrease in ground effect, but a definitive answer cannot be provided, given the difficulty of experimentally determining the extent of the impact that ground effect might have on the structure. Another hypothesis relates to the magnitude of the θ_{max} angle and its rate of change per unit length of monopod for long monopods (Figure 3): The very small value of θ_{max} indicates that further changes in monopod length will almost imperceptibly affect both the inclination of the drone and its horizontal displacement from the pivot point, therefore having a minimal impact on battery consumption. This rationale is based on the assumption that the monopod can still be considered massless: All of the monopods used weighted well below 1 gram (3% of the weight of the drone), so was considered a suitable assumption.

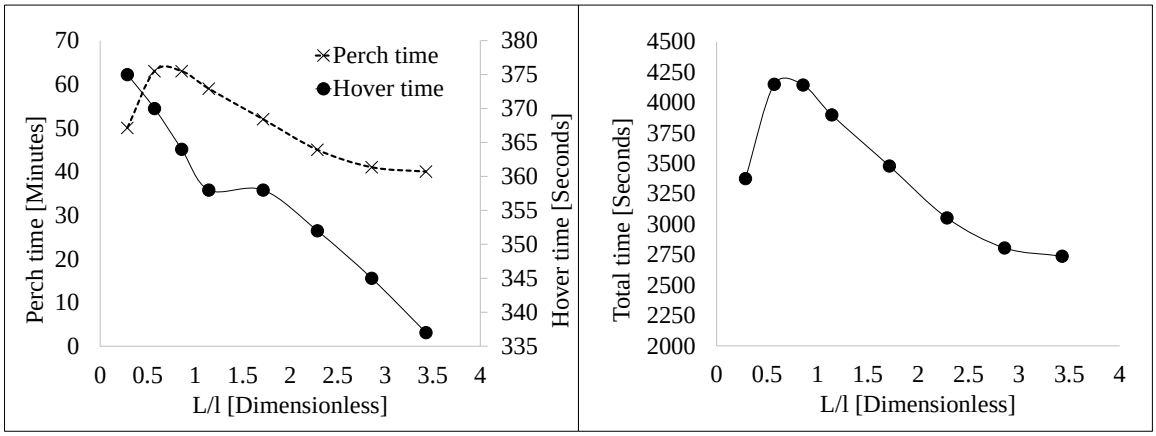

Figure 11: (a) Perch time and hover time as a function of L/l (5 trial average), (b) Total operation time as a function of L/l

As expected, instrumenting a quadcopter will negatively impact hover times. These were recorded for all monopod lengths (Figure 11(a)), and the clearest trend observed was that as monopod length increases, hovering time decreases. For the shortest moment arm ratio, a reduction in flight time of 30 seconds (7.4%) can be observed, while for a moment arm ratio of 3.4 the loss increases to 68 seconds (16.7%). A closer study of the results reveals two different slopes, a steeper one with a decrement of 19.9 minutes per unit L/ℓ for small moment arm ratios, and a gentler slope of 12.2 minutes per unit L/ℓ. A plateau of constant hovering time can be observed between 1.14 and 1.71 moment arm ratios. This peculiarity could be related to the way in which the hover experiment was implemented: A 1 cm clearance to the ground was maintained from the tip of each monopod when hovering, with vertical and horizontal position of the drone being manually controlled via joystick. The thrust level required for each

monopod was found by trial and error until the drone would consistently hover at the required height. As monopod length increased, so would the distance of the quadcopter's body to the ground, requiring small increases in thrust each time [9]. The results indicate the possibility of the existence of two different ground flow regimes when hovering at low heights, which could be an important consideration when implementing an automatic landing mechanism for this technology. A clear limitation in these results is that the initial goal of the test, assessing the effect of monopod length on flight time, is not fully answered, since the impact that ground effect has on the required thrust might have hindered the effect of increased mass in the quadcopter.

Finally, perching time and hover time were added to find the moment arm ratio that would give the longest operation time, with 0.57 resulting the best ratio of all that were measured (Figure 11(b)).

An important measurement was the response of the quadcopter to a step signal, and finding the experimental maximum tilt angle. A laser-printed test rig was built for these purposes (Figure 12). It was designed to be thin enough to minimally disrupt airflow underneath the drone, and long enough to allow for experimentation with longer monopods.

To measure the maximum inclination angle, the tilt angle θ was gradually incremented each time on the pitch axis, until the quadcopter was no longer able to rotate. To avoid integrator build-up, the controllers were all disabled until the angle was completely set-up.

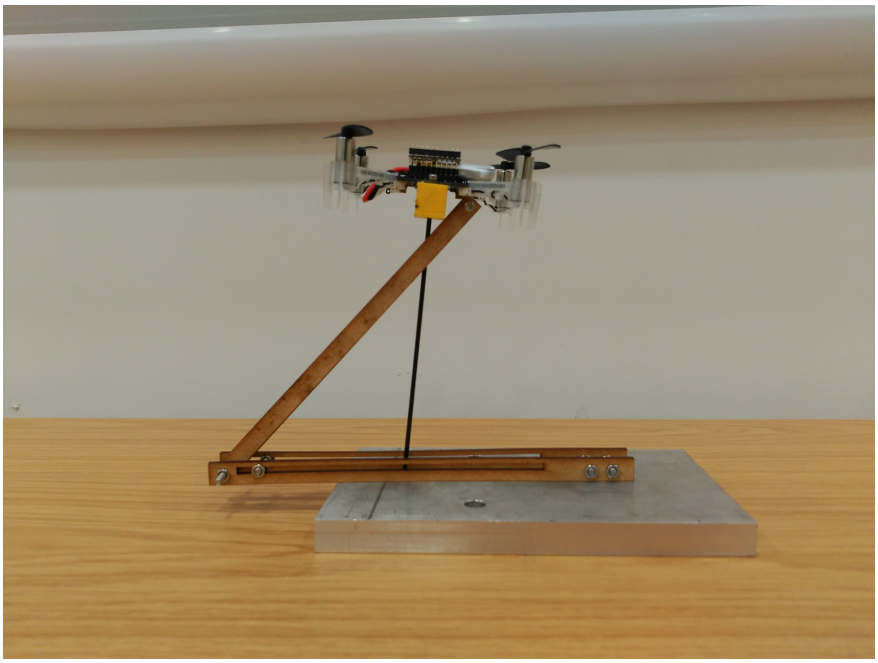

Figure 12: Angle measurement test rig

Figure 13 shows a plot of calculated and measured maximum tilt angles. At very short monopod lengths, the agreement is clear. However, for longer monopods the measured angle is higher than expected. This is thought to be due to the centre of gravity being lower than expected: The monopod length is measured from the lowest point on the centre of the quadcopter itself, without accounting for the monopod support. Therefore, adding a monopod support would lower the position of the CoG with respect to the ground. Despite these differences, the exponential decay as a function of moment arm ratio can still be observed in the experimental results.

As previously explained in the introduction, this novel technology would allow for yawing

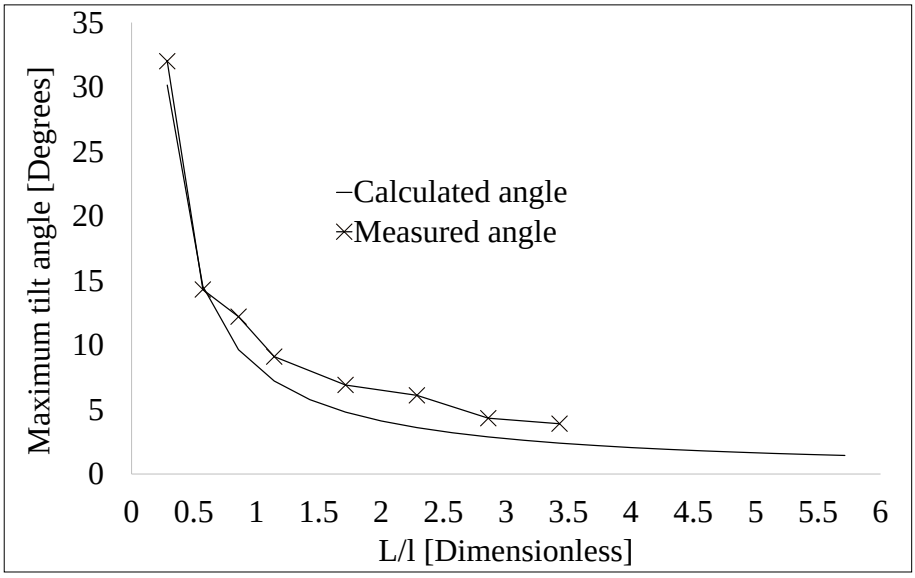

Figure 13: Measured and calculated maximum tilt angles as a function of L/ℓ

while perching. To assess this hypothesis, the maximum yaw rate at which the drone could rotate without falling was measured for every monopod length used (Figure 14). The results suggest that as monopod length increases, the ability to yaw decreases. However, a steep drop in maximum yaw rate appears between 1.14 and 1.71 moment arm ratios. Much milder gradients take place outside of this region. This transition region coincides with the one observed in Figure 11 for hover times, suggesting there is indeed a change in dynamic behaviour of the drone for these moment arm ratios.

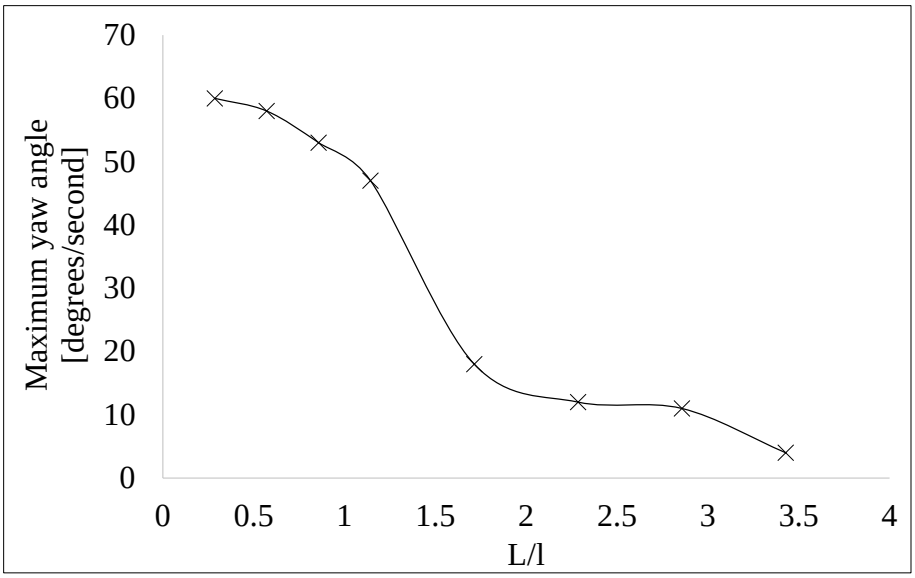

Figure 14: Maximum yaw rate as a function of L/ℓ

Finally, trends were also explored in the magnitude of PID parameters for the rate and attitude PID loops (Figures 15 and 16). It was found that as monopod length increased, the only parameters that needed to be modified were proportional gains, both for the attitude and rate controller loops. Results show an increase in gains as monopod length increases, although the increase is more marked for rate roll and pitch proportional gains. It should be noted that roll

and pitch gains remain the same, except for very short monopods, where values diverge. Yaw gains in contrast experience much milder variations.

Various hypotheses were formulated to justify the increase in K_p values. The larger horizontal displacements for the same angle and higher moment of inertia due to a longer monopod would require higher motor thrusts and responsiveness, achieved via increased K_p gains.

For flying, the factory PID parameters allowed for sufficiently accurate flight control for the monopod drone, although on occasions roll and/or pitch trims would be necessary to counter mass imbalances.

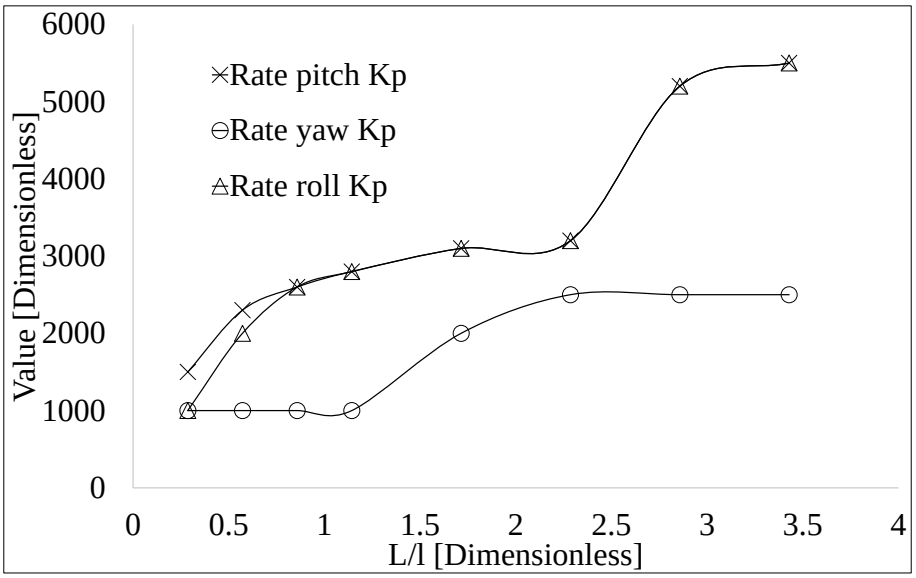

Figure 15: (a) Proportional gains of rate controller as a function of L/ℓ

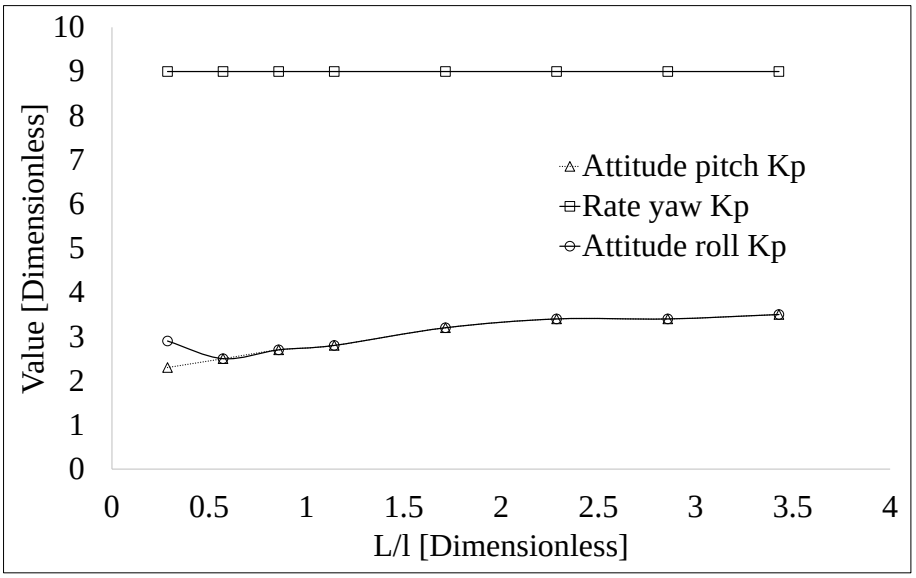

Figure 16: (a) Proportional gains of attitude controller as a function of L/ℓ

4 Conclusions and future work

The results obtained in this project show that it is possible to implement monopod perching on a real drone, and the very promising results suggest a realistic possibility of future implementation of this technology in commercial and research UAVs: Perching times were observed to last 500 to 800% longer than the hovering time of an unloaded drone, while hovering times of a drone with perching equipment only involved a reduction of between 7.4 and 16.7% compared to the unloaded drone.

Yawing while perching, another facet of monopod perching, was shown to work for all the monopod lengths studied, showing a decrease in maximum yaw rate as monopod length was increased, with yaw rates of $60°s^{-1}$ for a 1 cm monopod and $4°s^{-1}$ for a 12 cm monopod.

The small tilt angles involved meant monopod perching could be implemented via a cascaded PID controller, reducing the task to only finding the appropriate PID gains for each monopod length. However, this proved very challenging for many reasons which have been explained, all of which connected to the very small mass of the drone: The minimal inertia of the Crazyflie 2.0 quadcopter meant it was very sensitive to both damping at the perching point and very slight variations in positioning of its center of gravity during tuning. This in turn impaired the ability to troubleshoot and identify causes for the very low repeatability in behaviour of the quadcopter.

Relating to the experiments performed with the setup used in this report, some further research and experimentation will provide a deeper understanding of the technology: Only responses to step disturbances were explored to find the maximum tilt angle for each monopod. However, a very useful dataset would involve the response to impulse disturbances, similar to a gust of wind or impact with a surface element.

Some limited work was performed to implement automated landing using a VICON system with ROS parameter setting. An important step during landing is the smooth and effective transition between the landing controller and the perching controller, which could be challenging due to the impact at the moment of contact with the landing surface. As discussed in section 3, a variation in behaviour was observed between moment arm ratios of 1.14 and 1.71. This could play an important role in landing/take-off implementation and might become an important consideration when troubleshooting. A high-level Python script was written to implement both landing and take-off, allowing for transition between flying, landing and perching regimes. However, further work is still needed for real-time PID gain setting.

An interesting concept to be explored might be the inclusion of wireless charging while perching: The single-cell LiPo batteries that are used by the Crazyflie 2.0 drone take between 35 and 40 minutes to fully charge. Considering perching times can last up to 60 minutes, wirelessly charging the drone while perching via methods such as induction or solar power would improve mission times even further.

Owing to the issues with repeatability due to the mass of the Crazyflie 2.0, it is suggested that the experiments performed in this report are repeated on a larger drone: The larger inertia would allow for easier accommodation of CoG displacements, as well as diminishing the effect of damping at the perching spot while tuning, perhaps permitting the use of a scaled-up version of the 1 DOF test rig initially created, which would greatly simplify the task of tuning the PID controllers. Using a larger drone will permit exploration of the scalability of this technology by comparing data as a function of moment arm ratios (L/ℓ) with results in this report.

A further suggestion regarding the repeatability of results and PID tuning consists of adding a damping element to the tip of the monopod: In the experiments in this study, the monopod tip was completely rigid, which caused the 1 DOF test rig to be unsuitable for its purpose. Adding this damping element would be somewhat equivalent to using a larger drone in the sense that the quadcopter would be less sensitive to slight dynamic changes, but would also entail the advantages of cheaper and easier development (the Crazyflie 2.0 would still be usable, and its open-source code is easy to modify and develop), and would provide a shock-absorbing element for landing. Re-using the Crazyflie 2.0 would permit the inheritance of the dataset obtained in this report for comparison and verification.

References

[1] Harsha Prahlad. Electroadhesive robots—wall climbing robots enabled by a novel, robust, and electrically controllable adhesion technology. *IEEE*, 2008.

[2] Morgan T. Pope. A multimodal robot for perching and climbing on vertical outdoor surfaces. *IEEE*, 2016.

[3] Alexis Lussier Desbiens and Mark R. Cutkosky. Landing and perching on vertical surfaces with microspines for small unmanned air vehicles. *Journal of Intelligent and Robotic Systems*, 57(1), 2009.

[4] M. A. Graule, P. Chirarattananon, S. B. Fuller, N. T. Jafferis, K. Y. Ma, M. Spenko, R. Kornbluh, and R. J. Wood. Perching and takeoff of a robotic insect on overhangs using switchable electrostatic adhesion. *Science*, 352(6288):978–982, 2016.

[5] C. E. Doyle, J. J. Bird, T. A. Isom, J. C. Kallman, D. F. Bareiss, D. J. Dunlop, R. J. King, J. J. Abbott, and M. A. Minor. An avian-inspired passive mechanism for quadrotor perching. *IEEE/ASME Transactions on Mechatronics*, 18(2):506–517, April 2013.

[6] Akihiko Fukushima and Yoichiro Kawaguchi. Insect leg inspired friction attachment for miniature quadcopter. In *Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology*, ACE '15, pages 34:1–34:4, New York, NY, USA, 2015. ACM.

[7] Hao Wang. Monopod perching quadrotor. *University of Bristol and University of the West of England*, 2016.

[8] Michael A. Johnson and Mohammad H.Moradi. *PID Control*. Springer, 2005.

[9] Nita Darshini Nathan. *The rotor wake in ground effect and its investigation in a wind tunnel*. PhD thesis, University of Glasgow, 2010.

Appendices

A Quick start guide on development of Crazyflie 2.0 quadcopter

- Development of the Crazyflie 2.0 quadcopter is best done via an Ubuntu Virtual Machine specially designed by Bitcraze. A guide on how to download it and set it up is available in this link: https://www.bitcraze.io/getting-started-with-the-crazyflie-2-0/
- The VM (Virtual Machine) needs an extension pack to allow for USB compatibility. Download it here: https://www.virtualbox.org/wiki/Downloads
- A lot of useful information can be found on the Bitcraze wiki: https://wiki.bitcraze.io/ and Bitcraze forum.
- The Crazyflie can be flown out of the box using the Crazyflie Client in the Virtual Machine. This client also allows various parameters to be set in real-time and allows plotting of these parameters.
- The client (PC) side of the quadcopter is written in Python. A "crazyflie" class is used to connect to the crazyflie and send/receive data to/from it. The Crazyflie uses a library called cflib in which all these functionalities, incuding the crazyflie class, are implemented. You can find cflib in the VM on /home/bitcraze/Desktop/projects/crazyflie-lib-python/cflib. A set of simple examples are available under /home/bitcraze/Desktop/projects/crazyflie-lib-python/examples to learn how the Crazyflie code works. Install Pycharm on the Virtual Machine to aid development (note: you might have to install Java 8 in the virtual machine).
- The firmware in the Crazyflie is written in C, and the Bitcraze wiki offers a simple tutorial on how to modify the code and flash it to the microcontroller in the Crazyflie. The firmware code can be found in /home/bitcraze/projects/crazyflie-firmware/ in the virtual machine.
- You might want to install "pip" for Python 3.4 on the Virtual Machine to easily download packages for this Python version, as most of the code is based on Python 3.4
- A GitHub repository can be found with the code developed during this project and results. You can find it on https://github.com/f-rower/EF08. Three folders can be found: Crazyflie-firmware-Manual_Thrust, 000_Python_Flying_Code and Results :
  - Crazyflie-firmware-Manual_Thrust contains a modification made to the /home/bitcraze /projects/crazyflie-firmware/src/modules/src/controller_pid.c file in the Crazyflie firmware. This modification allows a base thrust level to be set as a parameter in the crazyflie client without the need of a gamepad, which is very useful when tuning the PID parameters, as it still allows the yaw, pitch and roll PID controllers to function. To implement this change, just substitute this file with the controller_pid.c file inside Crazyflie-firmware-Manual_Thrust and flash the new firmware to the crazyflie. As of 03/09/2017, this has already been done, so there would be no need to need to implement this step.

For perching experiments, the ManualThrust parameter was set to 1 (one) via the cfclient. The maximum value possible is roughly 60000. A base thrust level of 1 is so small that the motors will only run if yaw, pitch or roll angles are different from their reference values, which is desirable for minimising battery consumption while perching. The ManualThrust parameter is set to zero by default, and can be modified in the ”parameter” tab in the CFclient (figure 17) when a Crazyflie is connected.

Figure 17: Parameter tab in CFClient

If the gamepad thrust is used, the ManualThrust parameter will be overridden. Once the joystick thrust is back to zero, ManualThrust will again set the thrust level to whatever level it was set to.

– The 000_Python_Flying_Code contains the high-level Python script with which a landing, perching and flying framework was to be developed. This is written in Python 3.4, and is dependent on the cflib library and the gamepad.py module. This module is used to capture a gamepad and read inputs from it. You need to install the pygame module for Python 3.4 in your Virtual Machine.

  Under 000_Python_Flying_Code two folders can be found: _base_Flying_Code ,which is not necessary, and Flying_Code. Inside Flying_Code, the Flying_EF08.py script contains the main framework for switching between flying, perching and landing, as well as writing data from a specified parameter to a file (called ”fileread.txt”) in real-time, which is implemented by using the logging framework from cflib (see ”basiclog.py” example in /home/bitcraze/Desktop/projects/crazyflie-lib-python/examples). A second script, Plotter.py, allows for plotting of data from fileread.txt in real-time

if it is run alongside Flying_EF08.py.

– The Results folder contains this report, as well as the previous work by [7], and results obtained in this project. CAD files for the monopod support are also provided.

Expt_Data.xlsx contains perching and hovering time data, maximum yaw rate and PID gains for each of the monopod lengths studied. TiltAngle.xlsx contains results from maximum tilt angle and response to step inputs. Damping.ods contains the data from the 1DoF test rig damping assessment.

- A number of elements were used during this study, most of which has been left in a Farnell cardboard box in the flying arena. This box contains:

Item	Quantity
Crazyflie 2.0 quadcopter	2
Single-cell LiPo batteries, 3.7V, 240mAh	3
Crazyradio PA USB dongle	1
Logitech gamepad	1
1 DoF test rig	1
Improved test rig	1
Tilt angle test rig	1
LiPo battery USB charger	4
Ball bearings	6
1 m Carbon fibre tubes	5
Crazyflie 2.0 spare motors	6
Set of spare propellers	?
Set of spare motor mounts	?
Carbon fibre monopods, various lengths	1 each
Monopod supports	?

Table 3: Inventory