2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Learning Tethered Perching for Aerial Robots

Fabian Hauf[1], Basaran Bahadir Kocer[1,2], Alan Slatter[1], Hai-Nguyen Nguyen[3], Oscar Pang[1], Ronald Clark[4], Edward Johns[5], and Mirko Kovac[1,6]

***Abstract*— Aerial robots have a wide range of applications, such as collecting data in hard-to-reach areas. This requires the longest possible operation time. However, because currently available commercial batteries have limited specific energy of roughly $300\,\mathrm{W\,h\,kg^{-1}}$, a drone's flight time is a bottleneck for sustainable long-term data collection. Inspired by birds in nature, a possible approach to tackle this challenge is to perch drones on trees, and environmental or man-made structures, to save energy whilst in operation. In this paper, we propose an algorithm to automatically generate trajectories for a drone to perch on a tree branch, using the proposed tethered perching mechanism with a pendulum-like structure. This enables a drone to perform an energy-optimised, controlled 180° flip to safely disarm upside down. To fine-tune a set of reachable trajectories, a soft actor critic-based reinforcement algorithm is used. Our experimental results show the feasibility of the set of trajectories with successful perching. Our findings demonstrate that the proposed approach enables energy-efficient landing for long-term data collection tasks.**

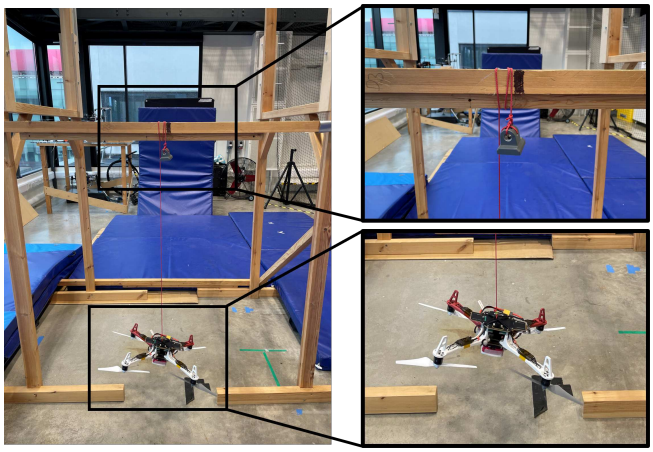

Fig. 1: Perched drone on a representative branch.

I. Introduction

In recent years, interest in drones has grown in all kinds of application areas [1]. Due to the independence and flexibility of drones, they can often be used to tackle tasks in hard-to-reach areas. A common example of inaccessible regions is large forests that are spread all over the world. There is also a growing demand for long-term spatial and temporal data collection from forests [2] but the long-term navigation of aerial robots is still a challenge to be resolved because of battery limitations [3]. While current aerial robots can function within the battery life limitations, there is a multi-modal approach proposed to expand the time for data collection by perching and staying idle [4].

One of the biggest limitations of drone tasks is having a limited flight time. This mainly depends on the ratio of the total drone weight and the stored energy that can be used to power the drone. As the maximum energy density in available batteries has an upper limit, aerial robots cannot fly for operations that require long-term data collection. To tackle this limitation, there are initial studies considering battery recharging and direct charging including multi-robot with charging stations for continuous missions [5], [6], mid-air battery swap and releasing the exhausted ones [7], [8], and powered over tether with a battery [9]. In terms of the power resource; solar cell endowed drones [10], [11], and laser beams to charge in mid-air [12], [13] are proposed. Recent activities focus on alternative sources of energy including fuel cells [14], [15] and multi-modality to sustain the data collection process with the aerial robots. Furthermore, there is ongoing research on battery chemistry, cell and pack engineering to have longer flights [16]. The other solutions are based on exploiting the aerodynamic interactions such as dynamic soaring [17], [18] and ceiling/ground effect [19], [20] as well as trajectory optimization [21].

One major category of aerial systems is the use of multi-modal methods, in which the system either modifies its structure or interacts physically with the surrounding infrastructure [22]. This category draws inspiration from observations of birds, which frequently perch on trees as a means of conserving energy and increasing safety. By applying a similar principle to drones, we can increase their available operation time by having them fly only when necessary and perch or land for the remainder of the time. This allows the drone to conserve energy and remain in the air for longer periods, ultimately improving its overall performance.

The literature already presents various methods of perching drones on trees, including physically grabbing the branch with a bistable mechanism [23], adhering to the branch with microspines [24], using a hemispherical cage and high friction material for support [25], and hanging from the

[1]Authors are with the Aerial Robotics Laboratory, Imperial College London, London SW7 2AZ, UK.
[2]Basaran Bahadir Kocer is with the Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1TR, UK.
[3]Hai-Nguyen Nguyen is with LAAS-CNRS, Université de Toulouse, CNRS, 7 Avenue du Colonel Roche, 31400 Toulouse, France.
[4]Ronald Clark is with the Department of Computer Science, University of Oxford, U.K.
[5]Edward Johns is with the Department of Computing, Imperial College London, London SW7 2RH, U.K.
[6]Mirko Kovac is with Laboratory of Sustainability Robotics at the Swiss Federal Laboratories for Materials Science and Technology, Switzerland.

branch with an anchor-like construction [26]. Recently, a new mechanism has been proposed for grasping the branch with the drone's bistable arm [27]. While these approaches have primarily focused on designing and implementing grasping mechanisms for perching on tree branches, this paper's main focus is on an automated method of perching a tethered drone on a branch.

To achieve an automated approach for tethered perching, we generated a set of feasible trajectories that can ensure safe perching without the propellers touching the tether while conserving energy. Additionally, we provided a reachable region for the design of tether-based perching, taking into account the weight of the tether and the diameter of the branch. This independent landing capability enables the drone to perform data collection tasks more efficiently. The proposed approach combines analytical and learning-based methods to safely perch the drone by approaching the target, wrapping the tether around the branch, and performing a 180-degree flip manoeuvre, as illustrated in Fig. 1. To the best of our knowledge, this is the first implementation of a neural network performing an automated tethered perching and flip manoeuvre on branch-like natural or man-made structures.

II. Problem Formulation

This paper addresses the problem of estimating a complete perching trajectory for a drone to safely connect to a branch and shut down its motors. The tethered perching manoeuvre is divided into two phases: approaching the target and wrapping the tether around the branch, followed by a flipping manoeuvre. The possibility of using reinforcement learning to solve the flipping problem is explored while ensuring that the drone's propeller does not touch the tether. This approach can enable long term environmental data collection, such as biodiversity soundscape monitoring. The effectiveness of the approach is evaluated by comparing the operational time of the drone with and without perching. The paper presents an automated tethered perching and flipping manoeuvre on branch-like environmental structures using analytical and learning-based approaches.

III. Methodology

To safely perch the drone on a branch-like structure, it is necessary to wrap the tether around the target branch, which can be broken down into two main steps as depicted in Fig. 2. The first step involves finding a manoeuvre that allows the tether to wrap around the targeted branch. The second step involves flipping the drone 180 degrees around the branch and shutting down the motors once it is securely perched. As these tasks can be approached independently, they were each addressed separately.

A. Approaching

For the drone to successfully perch on the tree, the tether needs to make sufficient contact with the branch. Sufficient wrapping is largely determined by the moment when the tether first comes into contact with the branch. The tethering manoeuvre is only possible if the contact point and the energy of the tether with the sensor weight allow it to be wrapped around. At the same time, the energy should be kept as low as possible to reduce the overall required energy.

Depending on the final relative position of the drone at the final moment during the approaching step, the contact point between the tether and the targeted branch will change. Eq. (1) can be derived from the energy obtained from the sum of the potential energy and the kinetic energy. The two cases in Fig. 3 are equated based on energy conservation. This formula describes the relationship between the length of the tether L, the velocity v and the diameter of the targeted branch. The contact angle θ and the lengths L_1 and L_2 are shown in Fig. 3. In Eq. (1), the left-hand side is proportional to the kinetic energy of the contact moment and must be reduced to minimise the needed energy.

$$\frac{v^2}{2g} = \sqrt{L_2^2 + \left(\frac{d}{2}\right)^2} + L_1 sin(\theta) + \frac{d}{2}\cos(\theta) \qquad (1)$$

The boundary conditions from this equation are determined by the symmetry of the branch. Depending on the approaching direction of the drone, only the side of the branch facing the drone is relevant as a contact point. As the tether is mostly straight underneath the drone because of the weight, the boundary conditions for θ are $[0, \pi/2]$. To avoid colliding with the drone while wrapping and ensuring the minimum length to perform n wraps around the branch, the length can be determined as

$$L = 2nd\pi. \qquad (2)$$

To successfully perch, the kinetic energy given in the lowest point has to be sufficient to lift the weight above the highest point while wrapping around the target. From Eq. (1), the ideal contact angle for energy minimisation can be determined to be $\pi/2$ within the given boundaries. At the final point, the horizontal velocity of the drone must be sufficient to ensure a wrapping of the tether.

To meet all these requirements, this manoeuvre was first performed manually to get a better understanding of the first step of the tethering manoeuvre. The successful manual manoeuvre was then analysed for the positions and their

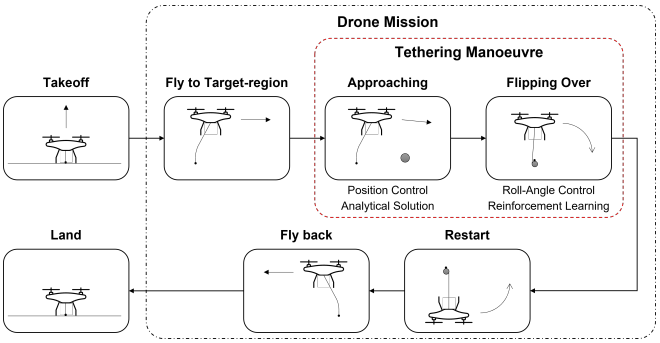

Fig. 2: Overview of the complete drone mission including the tethering manoeuvre. Our methodology proposes a tethering manoeuvre illustrated in the box with red dashed lines.

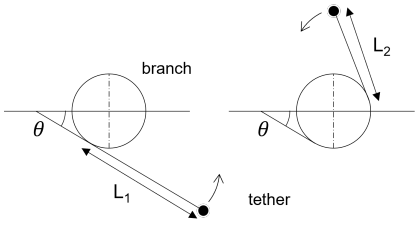

Fig. 3: Sketch of the first contact of the tether of the drone with the branch and the moment in which the weight overcomes the highest point of the spiral.

velocities to produce a trajectory which can lead to a reproducible successful tethering manoeuvre of the drone. The complexity of the task is addressed by imitating the trajectory of the manual behaviour of the system. In this approach, the drone performs a swing motion to wrap the tether. The experimental results are imitated considering the velocities for the swing motion.

It is important to make sufficient contact with the branch and to hold the weight of the drone. The weight that can be supported by the rope depends on the number of wraps around the branch. The relation can be estimated with the Euler-Eytelwein relation [28].

B. Flipping Over

In the last step, the drone is connected to the branch and should be able to safely disarm without colliding with the tether or the branch. The system can be moved to the point which has the same height as the branch, but with a distance of $L/2$ to the side. This point is easy to reach as the drone does not need to act with aggressive manoeuvres until this point. To ensure a safe flipping manoeuvre, the roll angle is adjusted along the trajectory while keeping the thrust at lower ranges. Therefore, the power requirements of the drone can be minimised. At the lowest possible position of the drone, the thrust is reduced to zero and the drone can be safely disarmed.

The flipping over manoeuvre is tackled dynamically, using a reinforcement learning algorithm. Thus, this part of the tethering manoeuvre is defined as a Markov Decision Process. Thereby, the current state $s_t \in \mathcal{S}$ at each time t is detected by the algorithm through the observation $o_t \in \mathcal{O}$. Together with the policy $\pi(s_t)$, this observation leads to an action $a_t \in \mathcal{A}$. As feedback, the reward r_t is provided from the reward function $R(s_t, a_t)$. This is aimed to be maximised for all future steps from t to t_{max} by optimising the used policy π^*. For the flipping over, the observation o_t of the state is given by the relative position of the drone to the branch, its roll angle and the velocity. To reduce the gap to the real-world application, sensor noise is added to the position of the drone. The resulting action a_t determines the change in the roll angle of the drone. The maximum simulation time t_{max} is set to be $10\,\mathrm{s}$.

To increase the learning performance of the reinforcement algorithm, two different reward functions $R_1(s_t, a_t)$ and $R_2(s_t, a_t)$ are defined and the weighted average is used. $R_1(s_t, a_t)$ favours following the baseline of the algorithm by minimising the difference in the roll rate between the given baseline and the actual simulation. This baseline solution gives a possible manoeuvre that has a constant roll rate over the whole flipping step. The roll rate of the baseline is picked manually. $R_2(s_t, a_t)$ rewards the algorithm for reaching the final position quickly and safely, by rewarding the system for permanently moving towards the final position.

$$R_1(s_t, a_t) = \begin{cases} I(s_t, a_t), & \text{for } t \leqslant t_I \\ I(s_t, a_t)\ M(t), & \text{else} \end{cases}$$

$$I(s_t, a_t) = 3 \times 10^5 \times (0.1 - \min(|s_\alpha - s'_\alpha|,\ 0.1))^4$$

$$M(t) = \frac{t_{\max} - t}{t_{\max} - t_I}$$

where s_α and s'_α are the roll angel of the current and the baseline state; t_{max} is the maximum time of the simulation; t_I is the time where the final point should be reacted.

$$R_2(s_t, a_t) = \begin{cases} R_L + \frac{\Delta X_{\text{target}} - 0.005}{l_r} \times 50, & \text{for } \frac{d_{\text{target}}}{d_{\text{drone}}} > 1 \\ R_L + \frac{\Delta X_{\text{branch}}}{d_{\text{drone}}} \times 100, & \text{else} \end{cases}$$

where R_L is the reward from the last step d_{target} is the distance to the target position; d_{drone} is the size of the drone; ΔX_{target} and ΔX_{branch} is the change in the distance made to the final position and to the branch, and l_r is the length of the rope.

In the first part of the training, the reinforcement learning algorithm is only rewarded by $R_1(s_t, a_t)$ to learn to follow the baseline solution. In this way, the algorithm showed improved learning and performance as it started at a possible solution. In the middle part, the algorithm transitions to approximately $20\,\%$ $R_1(s_t, a_t)$ and $80\,\%$ $R_2(s_t, a_t)$. This evolution of the weights during training is shown in Fig. 4. In the last part, the algorithm explores the solution space and tries to improve the initially given baseline solution. The main reason for our proposed method is based on having a feasible baseline that can allow shorter training time. However, our solution also improves energy efficiency and a smooth set of trajectories by exploring with designed reward function.

Real-world experiments were conducted to validate the trajectories. The specifications of the drone used in the experiment are detailed in Table I. A wooden framework was utilized as a branch, capable of withstanding the forces exerted during the tethering manoeuvre. To ensure safe flipping and disarming, the shutting down phase was executed in various attempts within the flight arena. Initially, the drone was secured to the branch to prevent it from flying away while descending. The baseline solution, which employs manually crafted decrements for the roll rate, was utilized to execute the final step. Once feasibility was established, the set of learned trajectories generated by the neural network was employed. Subsequently, approaching trajectories were integrated to allow the tether to wrap around the branch, resulting in successful and seamless flipping trajectories. Multiple experiments were conducted at this stage with varying battery rates to confirm the feasibility of the trajectories.

C. Evaluation of the Tethering Manoeuvre Mechanism

To assess the usability of the proposed tethering manoeuvre, we compared the operational time of a drone equipped with the perching capability to that of a standard free-flying drone. The two systems differ in weight and functionalities, with the tethered drone carrying the additional weight of the perching mechanism and requiring extra energy to approach the branch for each tethering manoeuvre. However, the perching mechanism also extends the operational time by enabling the drone to remain perched and conduct data collection tasks. As the operational time of the two systems approaches a certain threshold, they become equal. The additional energy required to perch the drone was quantified through the conducted experiments, with the energy consumption of the drone hovering used as the baseline measurement.

As an approximate estimate, we assume that the tether increases the weight of the drone by $8\,\%$. To evaluate the impact of this additional weight on flight time, we compare the energy density of the drone. Energy density is defined as the total energy available from the batteries divided by the total weight of the drone. Since energy density and flight time vary by drone manufacturer, we utilized available online resources, including the xcopterCalc - Multikopter Calculator, to construct different drone models and approximate their flight performance. Using this tool, we created a model of the Astro Flight - 819M-9T drone. Our results show that an increase in weight of approximately $8\,\%$ reduces flight time by roughly $12\,\%$. However, in our scenario, the system involves a mock-up sensor for sound data collection, and the additional weight of the tethering mechanism is the primary component.

IV. Experimental Setup

All experiments were conducted indoors within the Aerial Robotics Laboratory's flight arena, using a Vicon motion capture system. The specifications of the drone used for testing can be found in Table I. During the experiments, a tether measuring $1.6\,\mathrm{m}$ was attached to the drone, and the weight at the end of the tether was varied between $100\,\mathrm{g}$ and $200\,\mathrm{g}$ to simulate the weight of the sensor.

Flight Controller	Pixhawk 4
Companion Computer	Raspberry Pi 4 with Ubuntu-18 Server & ROS Melodic
Diagonal Length	450 mm
Motor	DJI 2312E 940KV
ESC	DJI E-Series 420Lite
Propellers	9450
Battery	4S 1500 mAH LiPo
Total Take-off Weight	1.2 kg

TABLE I: Specification of the drone used in the flying arena for the practical experiments.

V. Results

A. Approaching

The proposed solution is demonstrated to be feasible for reproducible trajectories that enable the drone to approach the branch, with the contact point of the drone at the branch being centred, as shown in Fig. 1. By estimating the waiting time, it can be concluded that an increase in branch diameter or the number of wraps will lead to a longer time needed to achieve full contact, as demonstrated in Fig. 5. In our experiments, a rectangular branch was used, with an equivalent radius of $0.036\,\mathrm{m}$ with respect to its circumference. Based on the trajectories observed in the flight arena, it was determined that it takes approximately $1.6\,\mathrm{s}$ on average for the drone to make three wraps around this branch.

B. Flipping Over

The learning framework proposed in this study was trained and the resulting trajectories were utilized in real time. During the manoeuvre, the drone was initially positioned at a distance of $L/2$ from the branch, and the learned feasible trajectories were then employed to execute a controlled flip while maintaining low thrust.

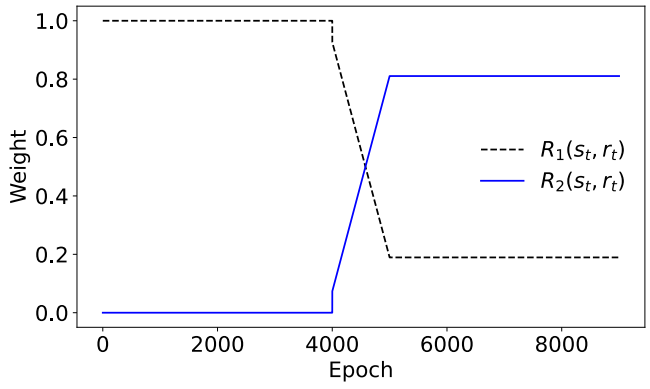

Fig. 4: Figure shows the weighting during training of reward functions $R_1(s_t, r_t)$ which follows the base line solution and the $R_2(s_t, r_t)$ which enhance the exploring of the solution space.

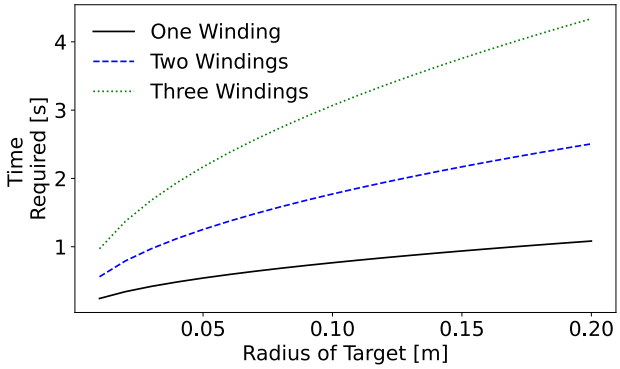

Fig. 5: The figure shows the numerically calculated waiting time from the first contact of the tether to the end of looping around the targeted branch.

The development of the reward function during the neural network training is depicted in Fig. 6, with respect to the measured baseline at an epoch around 4100. The weights assigned to the two reward functions, namely imitation and exploration, are plotted in Fig. 4. The figure shows that the algorithm initially learns to imitate the provided solution, followed by a transition to exploring different possibilities, ultimately resulting in a performance improvement of about 18 % above the baseline. Moreover, the standard deviation decreases over time during the imitation phase, while it peaks during the transition phase and then decreases to a low level of approximately 0.2.

Figure 7 (a) and (b) present a comparison between the basic trajectory and the improved trajectory generated by the neural network. The drone's actual path is illustrated with a dashed line while the angle of the drone at each point is denoted by bars. The time elapsed during the manoeuvre is depicted by grey shading, with the lighter shades indicating the initial stage. The green circle in the centre of the figure represents the target branch.

The real-world experiments demonstrated the feasibility of both the baseline and the improved solution generated by the neural network, resulting in a successful trajectory with the drone reaching the lowest position without colliding with the target or the rope. However, it was observed that the drone often deviated from the intended path during the manoeuvre. As shown in Fig. 8, the drone first moved to the starting position (negative y position), descended towards the target, and finally swung underneath the branch. Due to the drone's downward motion, the motion capture system occasionally failed to detect it, resulting in data gaps.

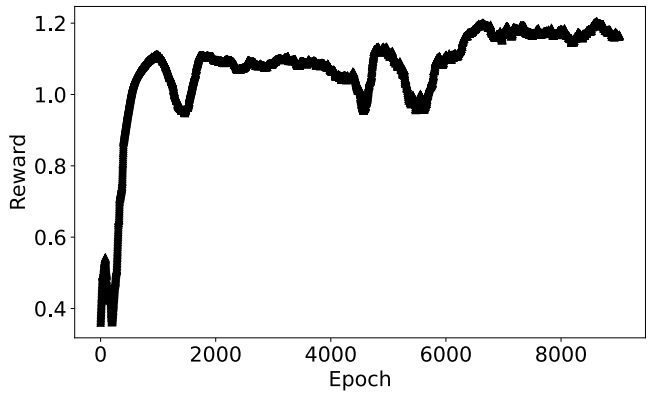

Fig. 6: Evolution of reward relative to the baseline of the model.

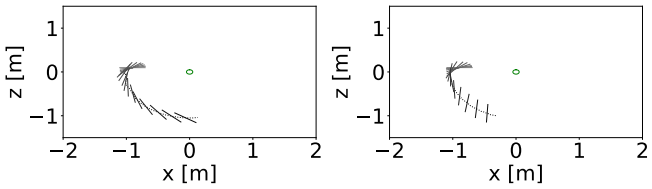

Fig. 7: Figure (a) shows the trajectory resulting from the baseline imitated in the first learning phase. Figure (b) shows an improved trajectory suggested by the neural network at the end of the training.

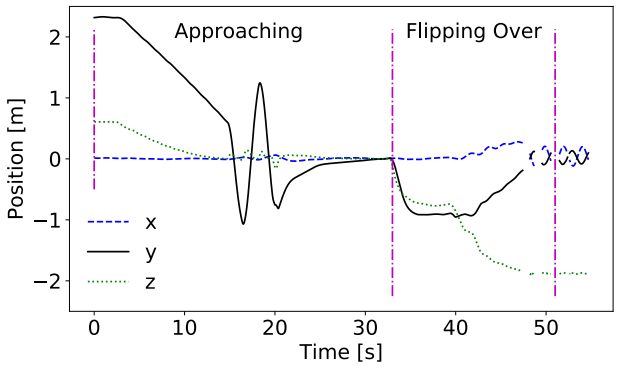

Fig. 8: Experimental data of the drone's position was measured throughout the automated tethering manoeuvre in the flight arena. Gaps in the data occur when the motion capture system has not captured the drone due to the target branch.

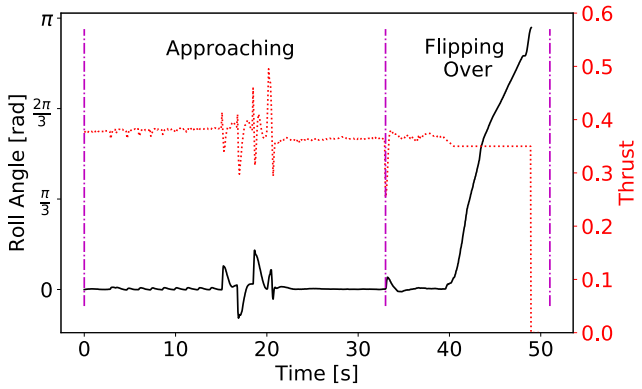

Fig. 9: The angles and thrust of the drone were recorded while performing all steps of the tethering manoeuvre. The thrust recorded here is a dimensionless parameter of the drone that indicates the current motor load in relation to the current maximum possible load (varies depending on the battery status).

C. Full Set of Trajectories

The complete tethering manoeuvre, consisting of all the individual steps, was executed and evaluated in the flight arena. Fig. 8 presents the resulting data, depicting the drone's trajectory from its starting point, which was set 2.5 m away from the target. After descending and reaching the starting position of the tether, the drone swung the tether and proceeded to the initial point for the flipping over phase. Finally, the drone executed the 180-degree flip using the learned trajectories. The transitions between each step were determined based on the respective running time.

In the flight arena, the drone attempted to approach the targeted branch twenty-one times. Sixteen out of these

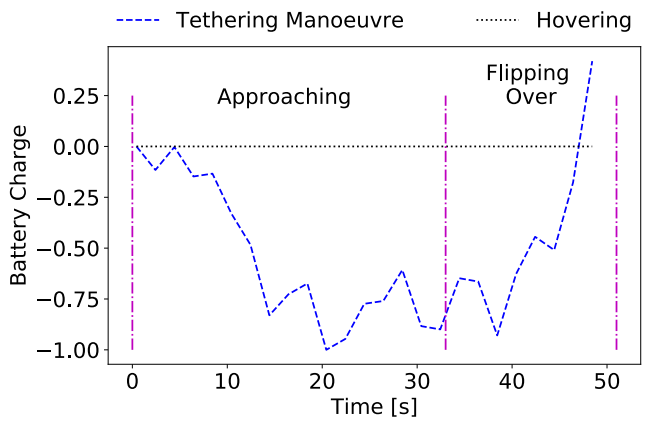

Fig. 10: This graph shows the normalised reduction in battery state-of-charge while performing the tethering manoeuvre compared to the drone just hovering in the air.

twenty-one attempts resulted in successful perching of the drone. Throughout the whole manoeuvre, the drone operated in fully autonomous mode. However, in the remaining five attempts, the drone failed to make sufficient contact with the targeted branch, resulting in a collision of the tether with itself as it wrapped around the branch. This collision caused the tether to bounce back instead of performing complete loops around the branch.

D. Energy Consumption for Tethering Manoeuvre

To further evaluate the system's performance, an estimation of the energy consumption during the tethering manoeuvre was conducted. The battery voltage was used as an indicator, with the baseline measured from the drone hovering in the air. As shown in Fig. 10, the overall energy consumption during the tethering manoeuvre did not exhibit a significant difference when compared to the energy consumption during hovering.

E. Discussion

The flight behaviour of the drone was found to be influenced by the dynamics of the tether. It was observed during the experiments that the non-slip condition between the rope and the branch after the initial contact might not hold true. In this case, the drone continues to pull the tether while it is not fully attached, resulting in a slip that reduces the energy required at the first contact point. To account for this, Eq. (1) needs to be corrected by including the term that reflects this effect. To compensate for the slip, the drone was positioned lower over the branch during the experiment, which increased the initial contact point and resulted in the slip of the tether along the branch. Another observation was that the final deceleration of the drone from the maximum velocity to the minimum range did not solely rely on the drone, as the tether made sufficient contact with the branch to hold the drone while still in flight, thus automatically reducing the drone's velocity. This phenomenon can be observed in Fig. 8, where the final point in the second swing is not reached.

The feasibility of flipping over was demonstrated in the second section, with the neural network showing improved performance compared to the baseline approach. The neural network was able to generate stable trajectories for the flipping over phase. Comparing the baseline solution with a trajectory produced by the neural network in Fig. 7, the drone's angle is seen to be increased during the latter. This results in earlier attainment of the final conditions and thus, energy savings.

To enhance the performance of the neural network, it is suggested to vary the length of the rope and the thrust during training, which simulates the conditions in a real environment. By doing so, the algorithm can adapt to these changes and potentially yield a better solution. This approach can further improve the performance of the algorithm and enable it to handle a broader range of initial conditions.

The solution presented in this paper has demonstrated the capability to produce a feasible trajectory for the tethering manoeuvre. The results showed that the drone successfully perched in sixteen out of twenty-one attempts, indicating the repeatability of the performed manoeuvre. The only observed cause of failure was when the tether struck itself during alignment in the in-plane configuration, resulting in unsuccessful perching. To mitigate this issue, the geometry of the weight could be modified to reduce the probability of the tether hitting itself. Alternatively, performing figure-of-eight-like trajectories in the x and y planes could also decrease the likelihood of the tether colliding with itself during loops.

VI. Conclusion

This paper presents a drone tethering manoeuvre approach that significantly reduces energy consumption during long-term aerial data collection operations. However, the drone must carry additional weight in the form of a tether and weight, necessitating at least 12 % perching time to achieve energy-saving benefits. Future work will test the proposed manoeuvre under real-life conditions, considering space and take-off constraints. The proposed tethering manoeuvre has the potential to expand drone applications by reducing energy usage and increasing operational time. Future studies will also explore mechanisms for recovering from the upside-down position, including cutting the tether to place the sensor on the environment.

Acknowledgment

We also thank former Aerial Robotics Lab members who have explored the work at various levels, particularly Bojia Mao and Kobi Kelemen. This work was partially supported by funding from EPSRC (award no. EP/N018494/1, EP/R026173/1, EP/R009953/1, EP/S031464/1, EP/W001136/1), NERC (award no. NE/R012229/1) and the EU H2020 AeroTwin project (grant ID 810321). Mirko Kovac is supported by the Royal Society Wolfson fellowship (RSWF/R1/18003). For the purpose of open access, the author(s) has applied a Creative Commons Attribution (CC BY) license to any Accepted Manuscript version arising.

REFERENCES

[1] K. W. Chan, U. Nirmal, and W. G. Cheaw, "Progress on drone technology and their applications: A comprehensive review," *AIP Conference Proceedings*, vol. 2030, no. 1, p. 020308, 2018. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/1.5066949

[2] B. B. Kocer, H. Stedman, P. Kulik, I. Caves, N. Van Zalk, V. M. Pawar, and M. Kovac, "Immersive view and interface design for teleoperated aerial manipulation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4919–4926.

[3] B. Ho, B. B. Kocer, and M. Kovac, "Vision based crown loss estimation for individual trees with remote aerial robots," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 188, pp. 75–88, 2022.

[4] B. B. Kocer, B. Ho, X. Zhu, P. Zheng, A. Farinha, F. Xiao, B. Stephens, F. Wiesemüller, L. Orr, and M. Kovac, "Forest drones for environmental sensing and nature conservation," in *Aerial Robotic Systems Physically Interacting with the Environment (AIRPHARO)*, 2021, pp. 1–8.

[5] A. Williams and O. Yakimenko, "Persistent mobile aerial surveillance platform using intelligent battery health management and drone swapping," in *IEEE International Conference on Control, Automation and Robotics (ICCAR)*, 2018, pp. 237–246.

[6] B. Galkin, J. Kibilda, and L. A. DaSilva, "Uavs as mobile infrastructure: Addressing battery lifetime," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 132–137, 2019.

[7] K. P. Jain and M. W. Mueller, "Flying batteries: In-flight battery switching to increase multirotor flight time," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3510–3516.

[8] Y. Guetta and A. Shapiro, "On-board physical battery replacement system and procedure for drones during flight," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9755–9762, 2022.

[9] M. Kishk, A. Bader, and M.-S. Alouini, "Aerial base station deployment in 6g cellular networks using tethered drones: The mobility and endurance tradeoff," *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 103–111, 2020.

[10] A. Malaver, N. Motta, P. Corke, and F. Gonzalez, "Development and integration of a solar powered unmanned aerial vehicle and a wireless sensor network to monitor greenhouse gases," *Sensors*, vol. 15, no. 2, pp. 4072–4096, 2015.

[11] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmann, K. Alexis, and R. Siegwart, "Perpetual flight with a small solar-powered UAV: Flight results, performance analysis and model validation," in *IEEE Aerospace Conference*, 2016, pp. 1–8.

[12] M. C. Achtelik, J. Stumpf, D. Gurdan, and K.-M. Doth, "Design of a flexible high performance quadcopter platform breaking the mav endurance record with laser power beaming," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 5166–5172.

[13] M. Lu, M. Bagheri, A. P. James, and T. Phung, "Wireless charging techniques for uavs: A review, reconceptualization, and extension," *IEEE Access*, vol. 6, pp. 29 865–29 884, 2018.

[14] J. Apeland, D. Pavlou, and T. Hemmingsen, "Suitability analysis of implementing a fuel cell on a multirotor drone," *Journal of Aerospace Technology and Management*, vol. 12, 2020.

[15] N. Belmonte, S. Staulo, S. Fiorot, C. Luetto, P. Rizzi, and M. Baricco, "Fuel cell powered octocopter for inspection of mobile cranes: Design, cost analysis and environmental impacts," *Applied energy*, vol. 215, pp. 556–565, 2018.

[16] V. Viswanathan, A. H. Epstein, Y.-M. Chiang, E. Takeuchi, M. Bradley, J. Langford, and M. Winter, "The challenges and opportunities of battery-powered flight," *Nature*, vol. 601, no. 7894, pp. 519–525, 2022.

[17] M. Deittert, A. Richards, C. A. Toomer, and A. Pipe, "Engineless unmanned aerial vehicle propulsion by dynamic soaring," *Journal of guidance, control, and dynamics*, vol. 32, no. 5, pp. 1446–1457, 2009.

[18] P. L. Richardson, "Upwind dynamic soaring of albatrosses and uavs," *Progress in Oceanography*, vol. 130, pp. 146–156, 2015.

[19] B. B. Kocer, V. Kumtepeli, T. Tjahjowidodo, M. Pratama, A. Tripathi, G. S. G. Lee, and Y. Wang, "UAV control in close proximities-ceiling effect on battery lifetime," in *IEEE International Conference on Intelligent Autonomous Systems (ICoIAS)*, 2019, pp. 193–197.

[20] B. B. Kocer, M. E. Tiryaki, M. Pratama, T. Tjahjowidodo, and G. G. L. Seet, "Aerial robot control in close proximity to ceiling: A force estimation-based nonlinear mpc," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2813–2819.

[21] F. Morbidi, R. Cano, and D. Lara, "Minimum-energy path generation for a quadrotor uav," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1492–1498.

[22] B. Stephens, L. Orr, B. B. Kocer, H.-N. Nguyen, and M. Kovac, "An aerial parallel manipulator with shared compliance," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 902–11 909, 2022.

[23] H. Zhang, J. Sun, and J. Zhao, "Compliant bistable gripper for aerial perching and grasping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1248–1253.

[24] H.-N. Nguyen, R. Siddall, B. Stephens, A. Navarro-Rubio, and M. Kovač, "A passively adaptive microspine grapple for robust, controllable perching," in *IEEE International Conference on Soft Robotics (RoboSoft)*, 2019, pp. 80–87.

[25] E. Aucone, S. Kirchgeorg, A. Valentini, L. Pellissier, K. Deiner, and S. Mintchev, "Drone-assisted collection of environmental dna from tree branches for biodiversity monitoring," *Science Robotics*, vol. 8, no. 74, p. eadd5762, 2023.

[26] S. Kirchgeorg and S. Mintchev, "Hedgehog: Drone perching on tree branches with high-friction origami spines," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 602 – 609, 2022.

[27] P. Zheng, F. Xiao, P. H. Nguyen, A. Farinha, and M. Kovac, "Metamorphic aerial robot capable of mid-air shape morphing for rapid perching," *Scientific Reports*, vol. 13, no. 1, p. 1297, 2023.

[28] J. A. Eytelwein, *Handbuch der Mechanik fester Körper und der Hydraulik: mit vorzüglicher Rücksicht auf ihre Anwendung in der Architektur aufgesetzt*. Fleischer, 1842.