2020 International Conference on Unmanned Aircraft Systems (ICUAS)
Athens, Greece. September 1-4, 2020

# Perching Upside Down with Bi-directional Thrust Quadrotor

Pengfei Yu[1], Gregory Chamitoff[2] and K.C Wong[1]

*Abstract*— **In this paper, we utilized a bi-directional thrust quadrotor with manipulators to mimic a bat's upside-down perching maneuver. This "perch and watch" capability could be beneficial to some specific mission involved extended endurance, such as monitoring wild animals in a natural reserve. This capability also avoids the downwash effect that commonly encountered during an upright perching. The bi-directional thrust is achieved by the dedicated Electric Speed Controller (ESC) firmware which could drive motor bi-directionally during the flight and with nearly symmetric fixed pitch propeller. This method of generating bi-directional thrust has a noticeable delay compared to a variable pitch propeller mechanism, thus a compensation strategy is adopted to minimize the delay in the process of reversing the propeller. So the altitude loss is minimized when a half flip maneuver is conducted. We also noticed that an inverted quadrotor generates more thrust compared to the upright configuration due to frame obstruction, thus controller gains also need to be compensated for accurate trajectory tracking. A customized offboard control method is developed to achieve this perching upside maneuvers with a motion capture system. Finally, a feasible perching trajectory is generated and the test result is presented.**

## I. INTRODUCTION

As the only known mammal capable of sustained flight, bats attract numerous people's attention by their marvelous manoeuvrability, especially their capability of perching upside down. In this paper, we try to mimic this bat's remarkable maneuver in a constructed environment using a bi-directional thrust quadrotor. This "perch and watch" capability would be beneficial to some specific mission involved extended endurance, such as monitoring wild animals in a natural reserve. This capability could also avoid the downwash effect that is commonly encountered during an upright perching. With a more versatile manipulator end-effector, the current platform could be extended for more complicated tasks, such as aerial grasping, capturing, and assembling.

### A. Bi-directional Thrust

bi-directional thrust could be achieved with a fixed pitch propeller or with a variable pitch propeller mechanism. A variable pitch propeller mechanism is able to achieve much faster changes in thrust output than a fixed pitch configuration, especially for larger radius or higher speed propellers, since the required change in inertia is larger. Some work has been done with a variable pitch propeller on small size quadrotor. In papers [1], [2], a servo mechanism and control algorithm is introduced to control thrust by varying the blade pitch, allowing a quadrotor to fly both inverted and upright.

There are also some papers that address the fixed pitch propeller problem [3], [4]. They conducted their experiments indoor with a motion capture system and outdoor with GPS separately. Both of the papers noticed the dead zone when reversing the propeller but without compensation. Another phenomenon not discussed by the above papers is that at the same throttle, the inverted quadrotor has more thrust than the normal upright configuration due to the absence of the downwash obstruction of the frame. If we do not vary the control gains matrix, the control response may degrade during inverted flight, and overshoot the desired position.

The capability of bi-directional thrust significantly improves the dynamic performance of a quadrotor since the controllable wrench simply doubles. One of the most obvious advantages is that a quadrotor now could approach ground with acceleration larger the gravitational acceleration or decelerate much faster compared to the uni-directional thrust quadrotor when climbing which is beneficial to constrained indoor flight. However, since the range of normalized motor speed expands from [0,1] to [-1, 1], the motor speed control resolution decreases to half of the uni-directional one in theory. Based on our tests, this disadvantage is not significant.

### B. Perching

Aerial perching has numerous potential applications for extending endurance and aerial manipulation. Researchers have demonstrated marvelous aerial perching with uni-directional quadrotor with many different kinds of manipulators, including passive biomimetic mechanism without consuming energy [5], a dry adhesive mechanism [6] and vacuum suction devices[7], [8]. Researchers have achieved nearly 90 degree perching with a uni-directional quadrotor [9]. However, to the authors' best knowledge, there hasn't been an inverted perching in any form. Here we expand the quadrotor's perching capability with a bi-directional thrust, so we can mimic a bat perching upside down.

One advantage of inverted perching or manipulating underneath the target is that it could avoid the downwash effect generated by propellers which have been mentioned in paper [6], [10]. If the quadrotor is trying to manipulate or perch on a target beneath it, the downwash effect will affect the quadrotor's position hold accuracy significantly. Especially if the quadrotor has a large weight, and its manipulator doesn't have enough operation space, it will be difficult for

[1]Pengfei Yu and K.C Wong are with School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, Sydney, NSW, Australia pengfei.yu@sydney.edu.au kc.wong@sydney.edu.au

[2]Gregory Chamitoff is with the Department of Aerospace Engineering, Texas A&M University, College Station, Texas, USA chamitoff@tamu.edu

978-1-7281-4277-7/20/$31.00 ©2020 IEEE

the quadrotor to overcome the downwash flow to reach the target.

### C. Organization of this Paper

The remainder of this paper is organized as follows: In Section II, quadrotor dynamics and control algorithm considering the delay and thrust difference are presented. In Section III, a feasible perching trajectory generating method is described. In Section IV, implementation including customized offboard control method is presented. In Section V, the result of upside-down perching test is shown. In Section V, we conclude this work and discuss future working directions.

## II. QUADROTOR DYNAMICS AND CONTROL

As Section III.A will show, our manipulator is very light-weight and has a symmetric configuration, thus the dynamics of the manipulator have little impact on the whole platform. We can therefore treat the movement of the manipulator as a disturbance to the quadrotor platform.

### A. Dynamics of Quadrotor

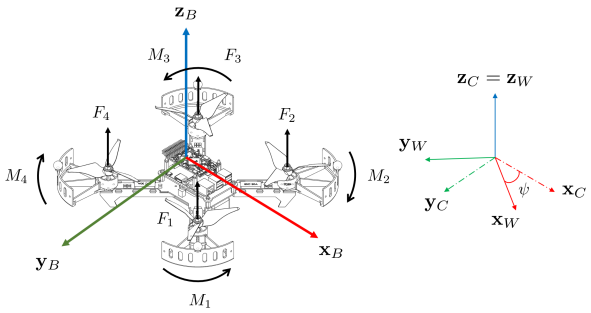


Fig. 1. Body, World and Intermediate Coordinates

The coordinates system is shown in Fig 1. The world frame $W$ is defined with $\mathbf{x}_W$ , $\mathbf{y}_W$ and $\mathbf{z}_W$. The origin of body frame $B$ is aligned with the Center of the Gravity (CG) of the quadrotor and $\mathbf{x}_B$ pointing forward and $\mathbf{z}_B$ pointing upward. Rotors' position and orientation is shown in figure 1. Intermediate frame $C$ has a certain desired yaw angle $\psi$ with the world frame $W$.

Let $\mathbf{r} \in \mathbb{R}^3$, represent the CG position of the quadrotor in the world frame, $\mathbf{v}^B$ represent the velocity in the body frame, ${}^W R_B$ represent the rotation matrix from body frame to world frame, and $\Omega \in \mathbb{R}^3$ represent the angular velocity in the body frame (or body rate). Then the state of the whole system $\mathbf{x}$ is

$$\mathbf{x} = \begin{bmatrix} \mathbf{r}^{\mathrm{T}} & \dot{\mathbf{r}}^{\mathrm{T}} & {}^W R_B & \Omega^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{1}$$

and the control input $\mathbf{u}$ of the system is

$$\mathbf{u} = \begin{bmatrix} f & \mathbf{M}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^4. \tag{2}$$

In equation (2), $f$ is total thrust generated by quadrotor in $\mathbf{z}_B$ direction and $M$ is the moment about the axes in frame $B$. Explicitly, thrust $f$ could be approximately calculated as

$$f = \sum_{i=1}^{4} F_i = \sum_{i=1}^{4} k_t w_i^2. \tag{3}$$

Moment $\mathbf{M}$ could be approximately calculated as

$$\mathbf{M} = \begin{bmatrix} -\frac{k_t l}{\sqrt{2}} & \frac{k_t l}{\sqrt{2}} & \frac{k_t l}{\sqrt{2}} & -\frac{k_t l}{\sqrt{2}} \\ \frac{k_t l}{\sqrt{2}} & -\frac{k_t l}{\sqrt{2}} & \frac{k_t l}{\sqrt{2}} & -\frac{k_t l}{\sqrt{2}} \\ k_m & -k_m & k_m & -k_m \end{bmatrix} \begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix}, \tag{4}$$

where $k_t$ is the propeller thrust constant, $k_m$ is the propeller drag constant, $l$ is the arm length of the quadrotor and $w_i$ is the RPM (rotation per minute) of the $i$th motor.

The quadrotor's dynamics can be described as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \frac{f}{m} {}^W R_B \mathbf{z}_B - g\mathbf{z}_W \\ {}^W R_B \hat{\Omega} \\ J^{-1}(\mathbf{M} - \Omega \times J\Omega) \end{bmatrix}, \tag{5}$$

where $m$ is the mass of the quadrotor, $J$ is the inertia tensor of the quadrotor in frame $B$ and $g$ is the gravity constant. The symbol $\hat{}$ is a vector to its skew-symmetric matrix operation.

Although the dynamics of the quadrotor is underactuated and nonlinear, the differential flatness property significantly simplifies the control and planning problem. The differential flatness property simply put is that the *states* and the *flat inputs* of the system are related to the selected *flat outputs* and their derivatives. In the case of a quadrotor, differential flatness means there is a direct mapping from the *flat outputs* of position and heading, plus their derivatives to the quadrotor state and also the flat inputs which are speed squared for each motor. The proof of this property can be found in literature[11], [12].

The main advantage of the differential flatness property is that if the continuous flat output is carefully planned then there must be a continuous flat input (motor speed squared). Thus it is convenient for generating a feasible trajectory as long as it fulfills the actuators constrains given the position, velocity, acceleration or other constrains.

### B. Controller

Figure 2 shows how we split our controller into a high-level controller and a low-level controller. The input of the high controller is the desired trajectory and feedback state of the quadrotor. The high-level controller consists of a position controller and an attitude controller. The position controller generates the desired attitude and the desired throttle. The attitude controller generates the desired body rate based on attitude error. The output of the high controller is the desired thrust and the desired body rate of the quadrotor. The low-level controller runs a rate controller that tracks the desired body rate command at a high frequency and a mixer that decides each motor's speed based on throttle and body rate. Since the low-level control process is at high frequency (usually at least 1kHz, in our case it is 8kHz) and is critical for the system stability, so it had better be run on a real-time computer, such as a microcontroller.

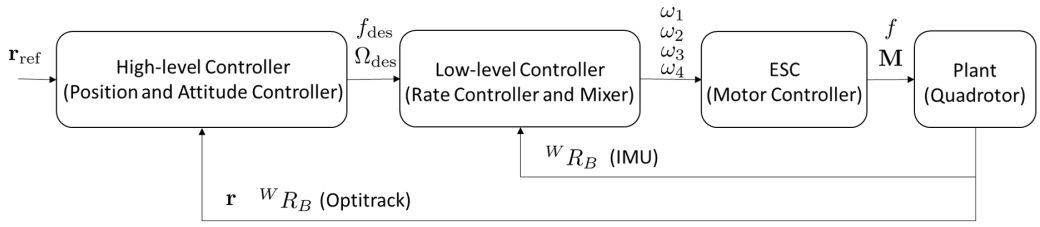


Fig. 2. Control Structure

The high level controller consists of position controller and attitude controller.

*1) Position Controller:* First we compute the desired acceleration $\mathbf{a}_{\text{des}}$ of the quadrotor

$$\mathbf{a}_{\text{des}} = \mathbf{a}_{\text{fb}} + \mathbf{a}_{\text{ref}} + g\mathbf{z}_{\text{W}}. \tag{6}$$

In equation (6), $\mathbf{a}_{\text{ref}}$ is the feedforward term, it can be computed from the reference trajectory. Note this term could be critical for certain desired trajectory and attitude.

The feedback term is $\mathbf{a}_{\text{fb}}$ which in our case is essentially a PD controller and can be expressed as the position and velocity error

$$\mathbf{a}_{\text{fb}} = \sigma\left(\mathbf{k}_{\text{pos}}\left(\mathbf{r} - \mathbf{r}_{\text{ref}}\right) + \mathbf{k}_{\text{vel}}\left(\mathbf{v} - \mathbf{v}_{\text{ref}}\right)\right). \tag{7}$$

where $\sigma = \begin{cases} 1 & upright \\ -1 & inverted \end{cases}$ and $\mathbf{k}_{\text{pos}}, \mathbf{k}_{\text{vel}}$ are diagonal gain matrix and they should be different for upright or inverted state for better tracking performance. Note that $\mathbf{a}_{\text{fb}}$ is related to the direction of the quadrotor.

Based on the desired acceleration and desired yaw direction, we can compute the desired attitude. Since the quadrotor could only generate thrust alongside its $\mathbf{z}_B$ axis

$$\mathbf{z}_{B,\text{des}} = \frac{\mathbf{a}_{\text{des}}}{\|\mathbf{a}_{\text{des}}\|}. \tag{8}$$

By projecting the desired acceleration onto the current $\mathbf{z}_B$, we can get the desired thrust

$$f_{\text{des}} = \mathbf{a}_{\text{des}}^{\text{T}}\mathbf{z}_B. \tag{9}$$

The desired yaw angle $\phi$ sets the direction of $\mathbf{x}_C$. Based on the right hand coordinates, the other desired axes and desired orientation $\mathbf{R}_{\text{des}}$ could be expressed as

$$\mathbf{x}_C = \left[\cos\left(\psi\right), \sin\left(\psi\right), 0\right], \tag{10}$$

$$\mathbf{y}_{B,\text{des}} = \frac{\mathbf{z}_{B,\text{des}} \times \mathbf{x}_C}{\|\mathbf{z}_{B,\text{des}} \times \mathbf{x}_C\|}, \tag{11}$$

$$\mathbf{x}_{B,\text{des}} = \mathbf{y}_{B,\text{des}} \times \mathbf{z}_{B,\text{des}}, \tag{12}$$

$$\mathbf{R}_{\text{des}} = \left[\begin{array}{ccc} \mathbf{x}_{B,\text{des}}, & \mathbf{y}_{B,\text{des}}, & \mathbf{z}_{B,\text{des}} \end{array}\right]. \tag{13}$$

Note in equation (11), a singularity can occur when $\mathbf{z}_{B,\text{des}}$ is parallel to $\mathbf{x}_C$. The singularity pose is actually the dividing pose that motor orientation should be changed. A control strategy is adopt in here, more details will be discussed in III.C.

*2) Attitude Controller:* The desired orientation $\mathbf{R}_{\text{des}}$ can be represented equivalently as a quaternion $\mathbf{q}^{\text{des}}$. The error quaternion $\mathbf{q}_e$ could then be calculated as

$$\mathbf{q}_e = \mathbf{q}^{\text{inv}} \odot \mathbf{q}^{\text{des}}, \tag{14}$$

where $\mathbf{q}^{\text{inv}}$ represents the quaternion inverse of the attitude quaternion $\mathbf{q}$. Thus the angular error about the frame $B$ axes could be represented

$$\mathbf{e}_a = 2\,\text{sgn}\left(q_{e,0}\right)\mathbf{q}_{\mathbf{e},1:3}, \quad \text{sgn}\left(q_{e,0}\right) = \begin{cases} 1, & q_{e,0} \geq 0 \\ -1, & q_{e,0} < 0 \end{cases} \tag{15}$$

where $\mathbf{q}_{\mathbf{e},1:3}$ represents the vector part of the quaternion. It essentially refers to the angle error alongside each corresponding axis, and $q_{e,0}$ represents the scalar part of the quaternion. To avoid the undesired unwinding phenomena where the feedback unnecessarily rotates the rigid body up to a full rotation, a simple fix is to reverse the sign of $\mathbf{q}_{\mathbf{e},1:3}$ when $q_{e,0} < 0$ [13].

We use a simple P controller to calculate the desired body rate. The desired body rate is

$$\Omega_{\text{des}} = \mathbf{k}_a\mathbf{e}_a, \tag{16}$$

where $\mathbf{k}_a$ is the diagonal gain matrix.

### C. Compensation Strategy

*1) Delay in Changing Orientation:* Compared to a variable pitch propeller mechanism a fixed pitch propeller has the disadvantage of a noticeable delay in reversing the rotation direction of the motor and propeller combination.

One part of this problem is that ESCs need a neutral zone to separate the normal direction and reversed direction. Often ESCs work in the Pulse Width Modulation (PWM) ranges from 1000$\mu$s to 2000$\mu$s, and in theory 1500$\mu$s is the midpoint. However, if the ESC is configured as bi-directional, then usually there is around 50$\mu$s set as the neutral zone to avoid unnecessary frequent changing motor orientation due to control signal jumping. And it is also the starting zone when arming the vehicle.

The other part of this problem is due to the inertia of the motor and propeller, it takes time to decelerate from near hover throttle and accelerate again to the inverted hover throttle. Although modern ESCs adopt strategies to quickly decelerate the propeller, there is still a noticeable delay in the orientation switch. This phenomenon is also mentioned in paper [1], [2].

When a bat perches upside-down, it relies more on changing the wing inertia[14]. Consequently, our strategy for compensating the delay is that when the quadrotor is near the singularity pose, given by the equation (11), it is around pitch or roll angle 90 degree which is also the pose for starting reversing motor's direction, we bring the throttle down to the neutral zone to stop the motor. One benefit of doing this is that the quadrotor will still rotate due to inertia, while the motor is already braked and ready for speeding up after the singularity pose. The other benefit is that since the singularity pose should be avoided and the nearby area is quite sensitive even with small change[15], we can simply not control quadrotor during this very short period with singularity pose. A more aggressive approach would be starting the reversing direction process earlier before reaching the singularity pose. This could be a future attempt.

*2) Thrust Difference:* Another phenomenon worth mentioning is that the motor thrust is actually different depending on the pose of the quadrotor. Assuming that motors are mounted on the top side of the frame, due to the obstruction of the airflow by the frame, it has less thrust compared to the inverted quadrotor with no obstruction in the way. This phenomenon is very similar to a tractor or pusher configuration problem. Differences in thrust of nearly 15% have been measured [16], [17]. So in order to have better trajectory tracking performance, the controller gains need to be tuned separately in an upright and inverted state.

## III. TRAJECTORY GENERATION

As briefly discussed at the end of section II.A, current state of the art trajectory planner utilizes differential flatness property to reformulate the trajectory generation problem into a convex optimization problem[18], [19], [20]. In this paper, we follow this methodology and utilize a widely adopted piecewise polynomial trajectory[21].

*1) Trajectory Generation:* Based on the differential flatness property, the trajectory needs to be at least fourth-order differentiable. The following reference trajectory is a 9th-order polynomial trajectory with $\mathrm{m}$ selected waypoints

$$\mathbf{r}_{\mathrm{ref}} = \begin{cases} \mathbf{r}_{\mathrm{ref},1} = \sum_{i=0}^{9} c_{i,0} t^i & t_0 \le t \le t_1 \\ \mathbf{r}_{\mathrm{ref},2} = \sum_{i=0}^{9} c_{i,1} t^i & t_1 < t \le t_2 \\ \cdots & \cdots \\ \mathbf{r}_{\mathrm{ref},m} = \sum_{i=0}^{9} c_{i,m-1} t^i & t_{m-1} < t \le t_m \end{cases} \tag{17}$$

The cost function is a positive-definite function with respect to the polynomial function coefficients vector $\mathbf{c}$. The trajectory that minimize the cost function is

$$\mathbf{r}_{\mathrm{ref}} = \arg\min_{\mathbf{r}} \int_0^{t_m} \left\| \mathbf{r}^{(4)} \right\|_2^2 dt. \tag{18}$$

To meet the continuity constrains, we impose:

$$\mathbf{r}_{\mathrm{ref},j}^{(k)}(t_j) = \mathbf{r}_{\mathrm{ref},j+1}^{(k)}(t_j) \ \forall k \in [0,4], j \in [1, m-1]. \tag{19}$$

To ensure the trajectory does not exceed the quadrotor's physical capability, we impose

$$\mathbf{r}_{\mathrm{ref}}^{(k)}(t) \le \mathbf{r}_{\mathrm{max}}^{(k)} \ \forall k \in [0,4], t \in [t_0, t_m]. \tag{20}$$

To meet the desired initial states and target states including position, velocity, acceleration, jerk and snap, we impose

$$\mathbf{r}_{\mathrm{ref},j}^{(k)}(t_j) = \mathbf{r}_{\mathrm{ref}}^{(k)} \ \forall k \in [0,4], j \in \{0, m\}. \tag{21}$$

Note that when using acceleration constraints to specify the desired attitude, the gravity needs to be considered.

To make sure that the quadrotor to go through all the desired waypoints, we impose

$$\mathbf{r}_{\mathrm{ref}}(t_j) = \mathbf{r}_{\mathrm{ref}} \ \forall j \in [1, m-1]. \tag{22}$$

Equations (18) to (22) construct a Quadratic Program (QP) problem. This problem can be solved by MATLAB optimal control toolbox GPOPS or other commercial optimization solvers such as Gurobi and MOSEK. Our implementation uses Gurobi.

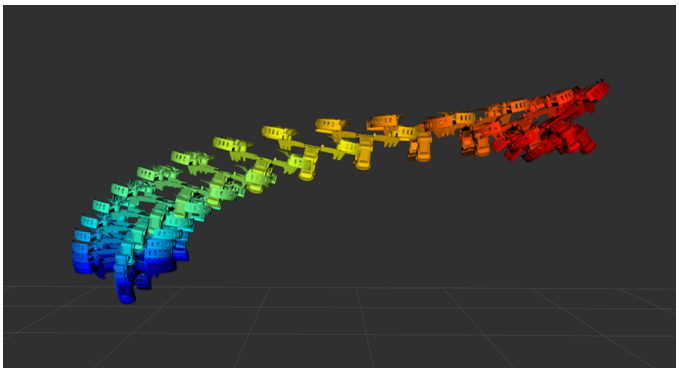

Fig. 3. Trajectory Example (Perch at 45 dgree)

Figure 3 shows an example trajectory from a hovering to 45 degrees pitched state. States of the quadrotor with manipulators changing from blue at the start of the manoeuvre to red at the final state, from hovering to 45 degrees pitching.

*2) Perching Trajectory:* With our upside-down perching trajectory, there are two stages. The waypoint for the first stage is the desired position that the quadrotor finishes a half flip. The waypoint for the second stage is the desired position for quadrotor engaging target grasping. The altitude in $\mathbf{z}_W$ direction in the first stage is slightly smaller than the second one. This configuration has three advantages. Firstly, the quadrotor is perching from underneath the simulated "branch", we can make sure that when the quadrotor finishing the half flipping, it won't dash into the "branch". Secondly, there would be a bigger altitude error in $\mathbf{z}_W$ direction, so there would be a sudden spike in the desired thrust setpoints, thus the altitude loss would be less. Thirdly, the last stage allows the quadrotor to have some time to adjust itself for a more accurate position holding. These advantages lower the difficulty of the perching problem. A more aggressive way of doing upside-down perching may omit the second stage. This will be a huge challenge on the quadrotor controller, trajectory generator (including servo control command) and could be the future's working direction. The detailed trajectory can be seen in figure 7.

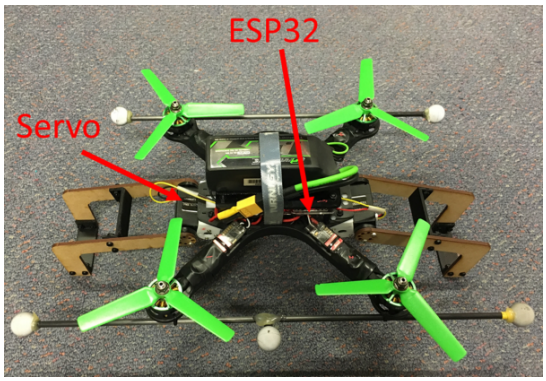


Fig. 4. Test Platform

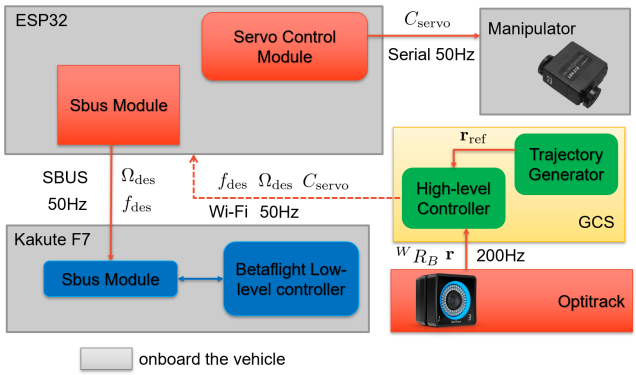


Fig. 5. System Architecture

TABLE I
MAIN COMPONENTS AND WEIGHT

| Component | Mass(g) | Quantity | Total(g) |
|---|---|---|---|
| T-motor KV2350 | 32.5 | 4 | 130 |
| E-MAX 45A ESC | 6.5 | 4 | 26 |
| Frame | 190 | 1 | 190 |
| Propeller 5045 | 5.8 | 4 | 23.2 |
| Kakute F7 | 10 | 1 | 10 |
| ESP32 | 7 | 1 | 7 |
| Power Distribution Board | 15 | 1 | 15 |
| Servo LDX-218 | 60 | 2 | 120 |
| Arm | 29 | 2 | 58 |
| Battery(1400mAh) | 188 | 1 | 188 |
| BEC (7.4V) | 8 | 1 | 8 |
| Other | 32.8 | 1 | 32.8 |
| Total | | | 808 |

## IV. IMPLEMENTATION

### A. Platform

Our custom built 250-size quadrotor (250 mm motor axle to motor axle diagonally) shown in figure 4 is built from off-the-shelf components. The main components and quantity are listed in I. The low-level controller is a Kakute F7 microcontroller running open-source Betaflight firmware. The manipulator consists of two LDX-218 digital servos and customized laser-cut and 3D-printed arms.

Since both of the servos are fixed to the quadrotor platform, the only moving part is the arm. The total weight of one moving arm of the manipulator is only 29 grams. Thus it has little impact on dynamics. The symmetric configuration has a relatively smaller disturbance to the platform[10] compared to a single arm configuration. It could also act as landing gear as shown in figure 4. This servo's stall torque is 15 kg·cm, strong enough for perching and landing with our 808 gram quadrotor.

The key component for reversing the motor's orientation is the ESC. Their firmware should support bi-directional operation. We use BLHeli32 ESC firmware. There are a few options in the firmware that could make a difference with the performance of the reversing orientation. The first one is motor timing, which is worth tuning since the default parameter is relatively conservative. Another important one is startup power which affects the smoothness of direction changing.

### B. System Overview

We use an ESP32 chip to receive commands via WiFi from the high-level controller running on a laptop to the low-level controller which is running on Kakute F7 microcontroller onboard. The advantages of the ESP32 chip are its compact size and the capability of controlling up to 16 servos simultaneously, which permits a more complicated manipulator.

Figure 5 shows the whole system architecture. States feedback is sent to the Ground Control Station (GCS) by motion capture system Optitrack at 200Hz. The trajectory generator and high-level controller are running on the GCS at 50Hz. The control commands from the high-level controller are desired thrust, desired body rates and servo position. The commands are sent to the low-level controller via ESP32 using SBUS protocol. The ESP32 also controls the servos' position by the command from high-level controller. The ESP32 communicates with the GCS via WiFi. The WIFI latency is relatively small(no more than 2ms), so no compensation policy is made in this setup.

## V. EXPERIMENT AND RESULT

We use an aluminum bar to mimic a branch in a natural environment. The quadrotor takes off right beneath the aluminum "branch", then finishes doing a half flip at the first desired position, then reaches the next desired position height and perched on the "branch". From hovering height (1m) to the perching position (1.7m), the quadrotor needs to finish one half flip (refer to the picture 1 in figure 6) and slightly moves upwards to the perching position(refer to the picture 2 and 3 in figure 6), during the whole process, it should neither overshoot the target (otherwise it will crash into the "branch") nor lose significant altitude. The flight log is shown in figure 7. Note that the final oscillation in the Z direction is due to the "branch" vibration caused by the quadrotor perching. The whole process can be seen from video footage at https://youtu.be/bOdo3Y8QKpo.

Figure 6 shows the upside-down perching process. Figure 7 shows the trajectory following during the upside-down perching.

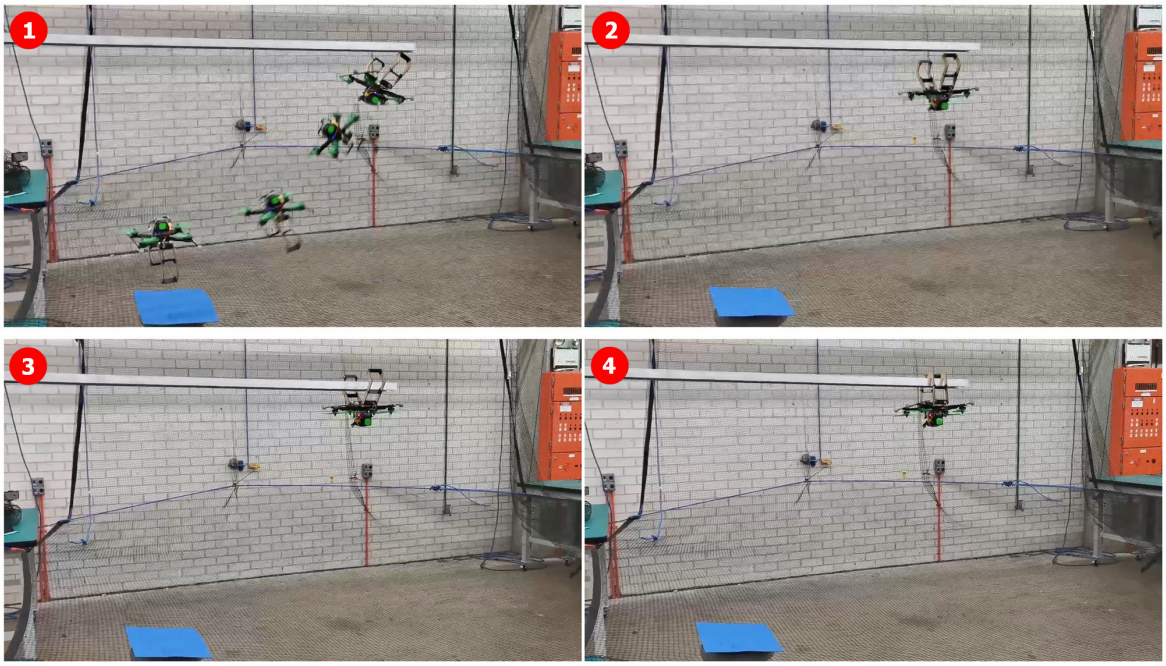


Fig. 6. Perching Upside Down Sequence 1)finishing a half flip at desired position; 2)moving toward perching position; 3)reaching perching position; 4)perched on the "branch"

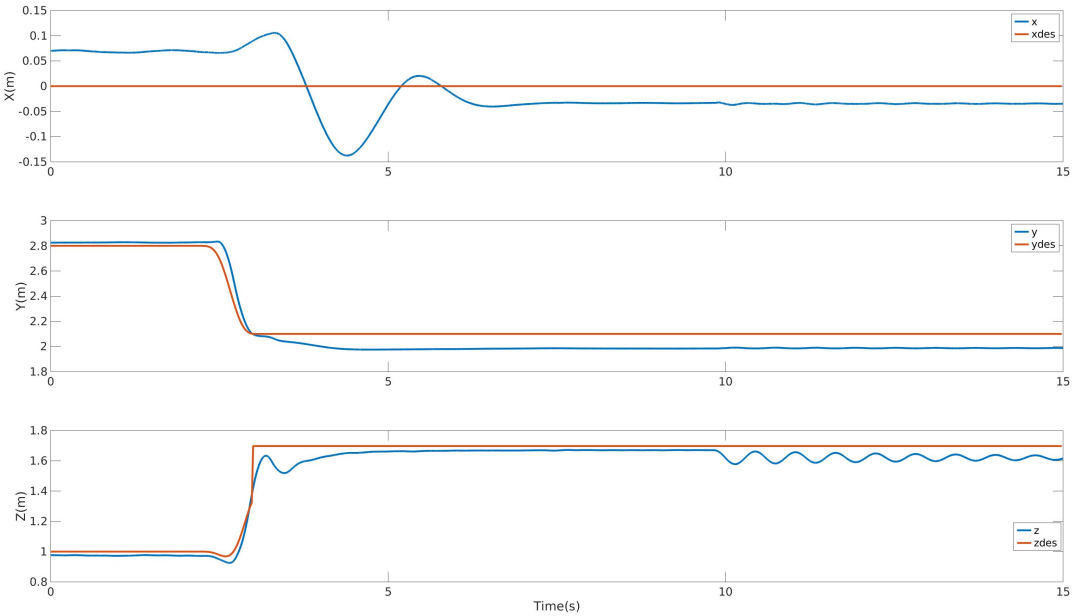


Fig. 7. Flight Log for upside-down perching

## VI. CONCLUSIONS

In this work, we present a bi-directional thrust quadrotor that could mimic a bat to perform an upside-down perching. We built the quadrotor platform with manipulators and designed a high-level controller with a compensation strategy for the delay in the motor reversing. We also describe a method of generating a two-stage feasible perching trajectory. With the above methods, we demonstrated an upside-down perching maneuver with our customized offboard control method in a constructed environment. For future work, a more adaptable manipulator could be developed and used for aerial grasping and manipulation. A more aggressive one-stage perching trajectory is worth exploring with bi-directional thrust capability.

## ACKNOWLEDGMENT

The authors would like to thank Derrick Ho and Jeremy Cox for their assistance in setting up the test environment.

## REFERENCES

[1] M. Cutler and J. P. How, "Analysis and Control of a Variable-Pitch Quadrotor for Agile Flight," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 10, pp. 1–14, 2015.

[2] N. Gupta, M. Kothari, and Abhishek, "Flight dynamics and nonlinear control design for variable-pitch quadrotors," *Proceedings of the American Control Conference*, vol. 2016-July, pp. 3150–3155, 2016.

[3] M. Maier, "Bidirectional Thrust for Multirotor MAVs with Fixed-Pitch Propellers," *IEEE International Conference on Intelligent Robots and Systems*, pp. 5727–5734, 2018.

[4] W. Jothiraj, C. Miles, E. Bulka, I. Sharf, and M. Nahon, "Enabling Bidirectional Thrust for Aggressive and Inverted Quadrotor Flight," *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, no. Section II, pp. 534–541, 2019.

[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*, vol. 18, pp. 506–517, apr 2013.

[6] J. Thomas, M. Pope, G. Loianno, E. W. Hawkes, M. A. Estrada, H. Jiang, M. R. Cutkosky, and V. Kumar, "Aggressive flight with quadrotors for perching on inclined surfaces," *Journal of Mechanisms and Robotics*, vol. 8, no. 5, 2016.

[7] M. T. Pope, C. W. Kimes, H. Jiang, E. W. Hawkes, M. A. Estrada, C. F. Kerst, W. R. T. Roderick, A. K. Han, D. L. Christensen, and M. R. Cutkosky, "A Multimodal Robot for Perching and Climbing on Vertical Outdoor Surfaces," *IEEE Transactions on Robotics*, vol. 33, pp. 38–48, feb 2017.

[8] H. W. Wopereis, T. D. van der Molen, T. H. Post, S. Stramigioli, and M. Fumagalli, "Mechanism for perching on smooth surfaces using aerial impacts," in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 154–159, IEEE, oct 2016.

[9] J. Thomas, M. Pope, G. Loianno, E. W. Hawkes, M. A. Estrada, H. Jiang, M. R. Cutkosky, and V. Kumar, "Aggressive Flight With Quadrotors for Perching on Inclined Surfaces," *Journal of Mechanisms and Robotics*, vol. 8, no. 5, p. 051007, 2016.

[10] P. Yu, Z. Wang, and K. C. Wong, "Exploring aerial perching and grasping with dual symmetric manipulators and compliant end-effectors," *International Journal of Micro Air Vehicles*, vol. 11, 2019.

[11] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 5420–5425, IEEE, dec 2010.

[12] D. Mellinger, *Trajectory generation and control for quadrotors*. PhD thesis, University of Pennsulvania, 2012.

[13] C. G. Mayhew, R. G. Sanfelice, and A. R. Teel, "On quaternion-based attitude control and the unwinding phenomenon," *Proceedings of the American Control Conference*, no. 3, pp. 299–304, 2011.

[14] A. J. Bergou, S. M. Swartz, H. Vejdani, D. K. Riskin, L. Reimnitz, G. Taubin, and K. S. Breuer, "Falling with Style: Bats Perform Complex Aerial Rotations by Adjusting Wing Inertia," *PLoS Biology*, vol. 13, no. 11, pp. 1–16, 2015.

[15] B. Morrell, M. Rigter, G. Merewether, R. Reid, R. Thakker, T. Tzanetos, V. Rajur, and G. Chamitoff, "Differential Flatness Transformations for Aggressive Quadrotor Flight," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7, 2018.

[16] B. Theys, G. Dimitriadis, P. Hendrick, and J. De Schutter, "Influence of propeller configuration on propulsion system efficiency of multi-rotor Unmanned Aerial Vehicles," *2016 International Conference on Unmanned Aircraft Systems, ICUAS 2016*, no. 2, pp. 195–201, 2016.

[17] P. Sharma and E. Atkins, "Experimental investigation of tractor and pusher hexacopter performance," *Journal of Aircraft*, vol. 56, no. 5, pp. 1920–1934, 2019.

[18] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," *Springer Tracts in Advanced Robotics*, vol. 114, pp. 649–666, 2016.

[19] H. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, and E. Galceran, "Continuous-time trajectory optimization for online UAV replanning," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 5332–5339, 2016.

[20] M. Burri, H. Oleynikova, M. W. Achtelik, and R. Siegwart, "Real-time visual-inertial mapping, re-localization and planning onboard MAVs in unknown environments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1872–1878, IEEE, sep 2015.

[21] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A system for autonomous flight using onboard computer vision," *IEEE International Conference on Robotics and Automation*, pp. 2992–2997, 2011.