2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

PogoDrone: Design, Model, and Control of a Jumping Quadrotor

Brian Zhu, Jiawei Xu, Andrew Charway, and David Saldaña

***Abstract*— We present a design, model, and control for a novel jumping-flying robot that is called PogoDrone. The robot is composed of a quadrotor with a passive mechanism for jumping. The robot can continuously jump in place or fly like a normal quadrotor. Jumping in place allows the robot to quickly move and operate very close to the ground. For instance, in agricultural applications, the jumping mechanism allows the robot to take samples of soil. We propose a hybrid controller that switches from attitude to position control to allow the robot to fall horizontally and recover to the original position. We compare the jumping mode with the hovering mode to analyze the energy consumption. In simulations, we evaluate the effect of different factors on energy consumption. In real experiments, we show that our robot can repeatedly impact the ground, jump, and fly in a physical environment.**

I. Introduction

In robotics, jumping mechanisms have been introduced based on bio-inspired locomotion principles [1], [2]. A jumping robot has a strong ability to overcome high obstacles [1], but it is unable to stay in the air. Although many researchers have investigated miniature jumping robots over the last decade [3], [4], [5], only a few have shown the integration of other locomotion modes. The jumpglider [6] is one of the hybrid robots that successfully achieved the integration of two locomotion modes. Their robot is equipped with foldable gliding wings that can improve travel distance and reduce the impact on landing. Though jumping robots are able to move in a complex changeable environment with high obstacles, the height of their jumps is limited by their structural design. For example, the flea-inspired jumping robot, designed by Noh et al. [3], can jump over a height of no more than 30 times its body height.

Quadrotors have become popular across various industries and as a research topic in recent years. Some of the common applications include search and rescue operations [7], exploration, aerial surveillance [8], and transportation [9]. One challenge that quadrotors must face is the ground effect [10], which inhibits the vehicle from operating close to the ground due to the turbulent flow produced by its rotors. Additionally, a major challenge that researchers have to overcome is the short flight time [11], which is typically around 5-30 minutes. To deal with this challenge, researchers have come up with strategies and techniques including using power sources with a higher energy density such as fuel cells [12], applying laser-beam in-flight recharging [13], [14], connecting with a power source through long-range tethering [15], and swapping batteries in flight [16].

B. Zhu, J. Xu , and D. Saldaña are with the Autonomous and Intelligent Robotics Laboratory (AIRLab), Lehigh University, PA, USA. Email: {blz222, jix519, anc619, saldana}@lehigh.edu

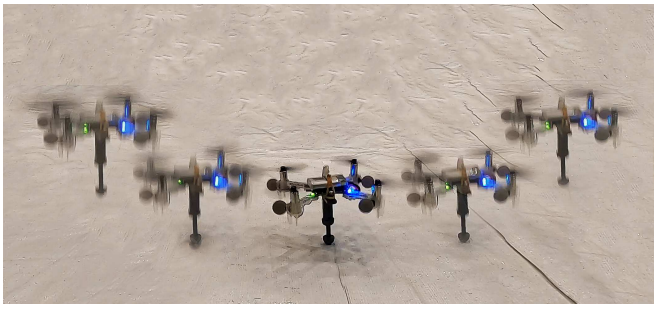

Fig. 1: A PogoDrone during a bounce. It descends, compresses, rebounds, and flies back to its original location. The PogoDrone at the center shows the case when the spring is fully compressed.

Although adding a gliding capability to a jumping robot [6] brings together the benefits of fast travel over a long distance in-air and energy-efficient obstacle clearance, the vehicle is not able to actuate in the air. The integration of multi-rotors in a jumping system gives it more maneuverability in flight. In a recent publication [17], the authors propose an active jumping mechanism with two rotors and a tail. The rotors are used to control roll and yaw angles, but they do not help the robot to stay in the air.

Inspired by the combination of jumping and aerial robots, we propose a jumping-flying robot, called the *PogoDrone*. We design, model, and control a novel robot that combines a quadrotor with a spring-based mechanism. Fig. 1 shows the PogoDrone during a jump. One of the main principles of creating this hybrid system is the conservation of energy. Typically, when a normal quadrotor drops from a certain height and collides with the ground, the kinematic energy is dissipated through the collision. In contrast, when a PogoDrone makes contact with the ground, the kinetic energy is stored through the compression of a spring as potential energy. Upon decompression, the spring releases the energy back to the PogoDrone. Note that not all the energy is conserved in such a process because of the imperfection of the spring. Thus, we still need to compensate for the loss of energy by actuating the rotors. A PogoDrone requires less power to function due to the passive jumping capability that preserves energy, which increases energy efficiency and the maximum flight time.

In comparison to a single-mode jumping robot, such as the Salto-1P [17], we offer a significantly simpler design that can jump and fly continuously. The simplicity gives our robot superior maneuverability which actively flying and addressing the Salto-1P's inability to perform in scenarios that require continuous hovering. Compared to a traditional

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9811970

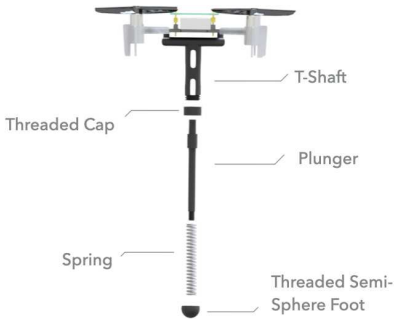

Fig. 2: Components of the PogoDrone

quadrotor, our PogoDrone allows operations close to the ground by deactivating its rotors when descending, which mitigates the ground effect problem [18].

II. Design

Our robot design adopts one of the simplest and most common methods to store potential energy, a spring in a pogo-stick. A PogoDrone is composed of a quadrotor fitted with a miniature and lightweight pogo-stick. The main components are described as follows.

a) Flying Vehicle: The quadrotor platform used for the PogoDrone is the Crazyflie 2.1. Open-source software and hardware, and its popularity in aerial robotics research make it an ideal choice for this project. The vehicle weighs 27g with the battery, and can carry a maximum payload of 15g. The dimensions are $92 \times 92 \times 29$ mm, and the 1-cell LiPo battery allows up to five minutes of hover time in the air.

b) Pogo-stick Mechanism: We show the parts of the miniature pogo-stick in Fig. 2. The dimensions of the pogo-stick are $35.25 \times 11 \times 55$mm. It is comprised of four 3-D printed parts along with a spring. The spring's length is 17.88mm. The spring constant was obtained experimentally to be 394.58N/m using weights and calipers. Using built-in mounting holes and a lightweight skeletonized frame, the primary T-shaped housing of the pogo-stick extends down the center of mass by approximately the same distance as the spring length, keeping the spring from generating unwanted torque when compressed. Our spring length, however, is constrained by the instability of the Crazyflie quadrotor. The longer the spring length, the longer the shaft and the lower the center of gravity. This increases the inertia tensor of the system which causes small changes in angular acceleration to generate large magnitudes of torque, thus making the system extremely difficult to fly. A plunger, held in by a threaded cap, inserts into the main shaft, compressing the spring on impact. Lastly, the semi-sphere foot at the end of the pogo-stick allows rotational motion from a pivot point. Given tight weight constraints, lightweight design of the pogo-stick is integral. Our current design weighs 4g, which is under the payload capacity of the Crazyflie (15g).

III. Model

Our PogoDrone is composed of a quadrotor with a bottom side attached jumping device, modeled as a spring-damper. The robot has a mass m and an inertia tensor $\boldsymbol{I}$. Since the

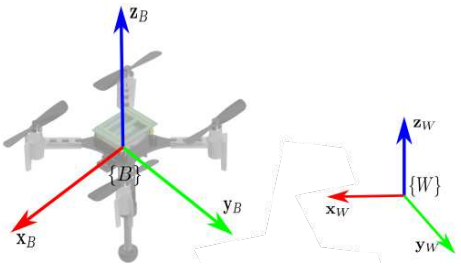

Fig. 3: The PogoDrone with its coordinate frame

mass of the jumping mechanism is small in comparison with the quadrotor, we assume that the center of mass of the PogoDrone is at the same location as the center of mass of the quadrotor. The coordinate frame for the quadrotor is denoted by $\{\boldsymbol{B}\}$ and its origin is at the center of mass of the quadrotor. The x-axis is aligned with the front of the quadrotor and the z-axis is pointing upwards. The spring-damper mechanism has one of its ends attached to the origin of $\{\boldsymbol{B}\}$ and extends along the negative z-axis of $\{\boldsymbol{B}\}$ (see Fig. 3). The pogo has a natural length l_0 and a minimum length l_{min}. The world coordinate frame is denoted by $\{\boldsymbol{W}\}$ with the z-axis pointing upwards. We denote the rotation matrix from body frame to world frame as ${}^{W}\boldsymbol{R}_{\boldsymbol{B}} \in \mathrm{SO}(3)$.

A. Sensors and Actuators

We use internal and external sensors to measure the location of the robot $\boldsymbol{r} \in \mathbb{R}^3$ in $\{\boldsymbol{W}\}$, velocity $\boldsymbol{v} \in \mathbb{R}^3$, attitude ${}^{W}\boldsymbol{R}_{\boldsymbol{B}}$, and angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$. The robot has four motor, as active actuators, that generate a thrust $f_q \in \mathbb{R}$ along the z-axis of the body frame and a torque $\boldsymbol{\tau} \in \mathbb{R}^3$. Therefore, our control input is the tuple $(f_q, \boldsymbol{\tau})$ which, through motor power distribution, can map to individual motor forces [19]. In addition, the pogo mechanism, as a passive actuator, generates a force,

$$f_s = -k\Delta L - b\frac{d}{dt}\Delta L, \tag{1}$$

where $k > 0$ is the spring constant, $b > 0$ is the damping factor, and $\Delta L = l - l_0$ is the amount of deformation of the spring. Since the spring is aligned with z-axis of the quadrotor, it does not generate any torque. Although the robot design has a spring and no damper, we included the damping term to take into account the energy lost due to friction and spring imperfections. The total force of the robot is along z-axis in $\{\boldsymbol{B}\}$,

$$f = f_q + f_s. \tag{2}$$

B. Dynamics

The dynamics of the robot depends on whether or not the PogoDrone makes a contact with the floor. The robot can be either in *a)* a flying state or *b)* a contact state. We describe the dynamics of the vehicle with Newton-Euler's equations.

a) Flying State

$$m\ddot{\boldsymbol{r}} + mg\boldsymbol{e}_3 = f\,{}^{W}\boldsymbol{R}_{\boldsymbol{B}}\,\boldsymbol{e}_3, \tag{3}$$

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} = \boldsymbol{\tau}, \tag{4}$$

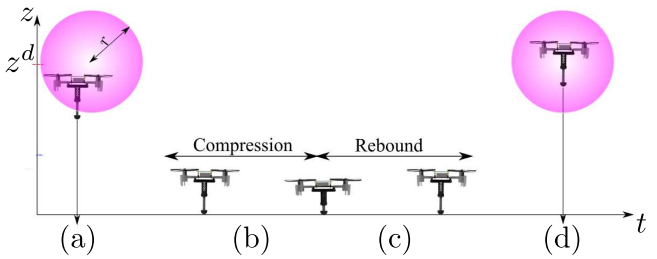

Fig. 4: Phases of a jump: (a) descend, (b) compression, (c) rebound, and (d) ascend.

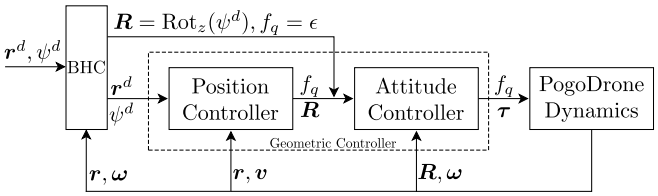

Fig. 5: Control Diagram for the PogoDrone. The BHC block reads the current position and angular velocity of the PogoDrone and switches the control mode between the full geometric control and attitude-only control as described in Section IV.

where $\boldsymbol{e}_3 = [0, 0, 1]^\top$ is a standard unit vector in $\mathbb{R}^3$ along z-axis. In this state, the robot does not have contact with the ground, and therefore, there is no spring compression, i.e, $f_s = 0$.

b) Contact State During contact, the connection between the pogo and the floor can be modeled as a spherical joint. Therefore, the robot rotates around the contact point instead of the center of mass. The Newton equation (3) holds the same but we need to consider that force provided by the floor is transferred through the spring to the PogoDrone as $f_s \neq 0$. The Euler equation has some differences from (4) since the rotation point changed,

$$\boldsymbol{I'}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I'}\boldsymbol{\omega} = \boldsymbol{\tau} + {}^W\boldsymbol{R}_B(l_0 + \Delta L)\boldsymbol{e}_3 \times (-mg\boldsymbol{e}_3), \quad (5)$$

where $\boldsymbol{I'}$ is the moment of inertia about the contact point using the parallel axis theorem. In the experiments, the inertia is not adjusted online because the contact state is ephemeral, i.e., according to Section V-B, the contact state lasts less than 100 ms for each bounce. Therefore, we rely on the attitude controller to compensate for any shifts in the inertia matrix. Note that there is an additional term due to the moment generated by the gravity force. We assume no slipping during the contact between the pogo tip and the ground.

IV. Control

The control objective focuses on allowing the PogoDrone to jump repetitively from a desired position $\boldsymbol{r}^d$ with a desired yaw orientation ψ^d. Our robot passes through four phases during each jump (illustrated in Fig. 4):

a) Descend: The robot falls maintaining its attitude to approach the ground vertically, so the spring will be perpendicular to the ground. During this phase, the propellers are used to maintain attitude only, so the robot can fall vertically to convert as much gravitational potential energy into kinetic energy.

b) Compression: The robot contacts with the ground and the spring compresses until it reaches its maximum compression length. During this phase, the spring converts the kinetic energy accumulated during the descend phase into elastic potential energy.

c) Rebound: The spring expands, pushing the robot upwards. The spring converts the stored energy into kinetic energy. Note that we apply a realistic spring model in (1), meaning that there exists a loss of energy during the compression and rebound. The amount of energy that we can store with this strategy will depend on the length, stiffness of the spring, loss of energy due to friction, and spring imperfections.

d) Ascend: The robot ascends to its original point $\boldsymbol{r}^d$. Since there is a loss in energy, the robot uses its propellers to reach the goal location.

We rely on the geometric controller on $\mathsf{SE}(3)$ [19], [20] for position and attitude control. We illustrate the control architecture on Fig. 5. Our module Bounce-Hover Controller (BHC) switches between the position and attitude controller depending on the phase of the jump. Our PogoDrone acts as a mass-damper-spring system when passively jumping, and as a quadrotor when actively hovering. During (a) descend and (b) compression, the robot does not need to compensate gravity or maintaining the position, so we use an attitude controller with $\boldsymbol{R} = \mathrm{Rot}_z(\psi^d)$ to guarantee that the robot will stay horizontal, roll and pitch equal to zero, while holding its desired orientation in yaw. Since we are not compensating for gravity, we do not need to generate a thrust force. However, the geometric controller does not accept $f_q = 0$ as an input, so we define an infinitesimally small constant $\epsilon > 0$ such that $f_q = \epsilon$. During (c) rebound and (b) ascending, the PogoDrone uses the stored energy to push upwards and the quadrotor's force to fly back to the original position. For

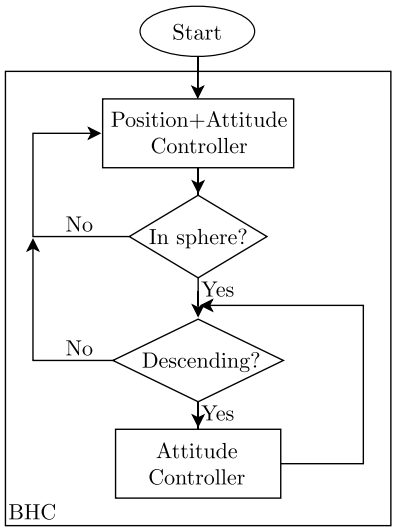

Fig. 6: Flowchart of the Bounce-Hover Controller (BHC).

this purpose, the PogoDrone uses position control with $\boldsymbol{r}^d$ and ψ^d as an input.

Our switching policy for the PogoDrone is defined by the algorithm in the BHC module. We summarize our algorithm in the flowchart in Fig. 6. The algorithm starts using the position controller to move the robot from any position to the desired position $\boldsymbol{r}^d$ with the desired orientation ψ^d. Due to the error in the position controller, we use a sphere with center at center $\boldsymbol{r}^d$, and radius $r > 0$. If the robot is within the sphere $\|\boldsymbol{r}^d - \boldsymbol{r}\| \leq r$ and the magnitude of its angular velocity $\|\boldsymbol{\omega}\|$ is less than a threshold, then it will start descending, leaving the sphere. During the descend and compression, the robot uses the attitude controller until the rebound starts. At that moment, the robot switches from attitude to position control and comes back to the original position $\boldsymbol{r}^d$. This iterative process continues repeating periodically.

V. Simulation and Experiments

We evaluate our PogoDrone model and design in simulations and actual robots[1]. We analyze the energy consumption in a time interval $[t_0, t_f]$. Since the current is proportional to the force generated by the spinning propellers, we use the rotors' force as a performance metric,

$$e(t_f) = \int_0^{t_f} \sum_{i=1}^{4} f_i(t)\, dt. \tag{6}$$

where f_i is the force generated by the rotor i at time t. The final time t_f defines the duration of the experiment with multiple jumps.

A. Simulations

We compare the PogoDrone in jumping mode versus hovering mode. We run the simulations in batch to evaluate the factors that affect the energy consumption of the PogoDrone: *a)* Noise level, *b)* Spring constant, *c)* Damping factor, *d)* Hover height.

We developed a 2D simulation environment in Python. We implement the dynamics of the PogoDrone and its interaction with the ground surface as in Section III using the Euler integration method with a time step of 1 millisecond. In the planar case, the PogoDrone has two propellers, making it an under-actuated system in 2-D, similar to a quadrotor in a 3-D environment. We first find a setting with the four factors where the PogoDrone saves energy with bouncing. Fig. 7 shows the 2-norm of rotor forces, the position of the PogoDrone during the bounces, the spring lengths, and its forces. The numerical values of the four factors used in the reference is shown in the caption. Note that the values stated in Fig. 7 are close to, but not precisely the values used in the real robot experiments. The spring constant and hover height are comparable. It is challenging, however, to choose a spring with a specific damping factor since it is difficult to measure. Through different simulations and real experiments, we approximated a damping factor into the simulation that

[1]The simulator and ros packages can be found at: `https://github.com/swarmslab/PogoDrone`

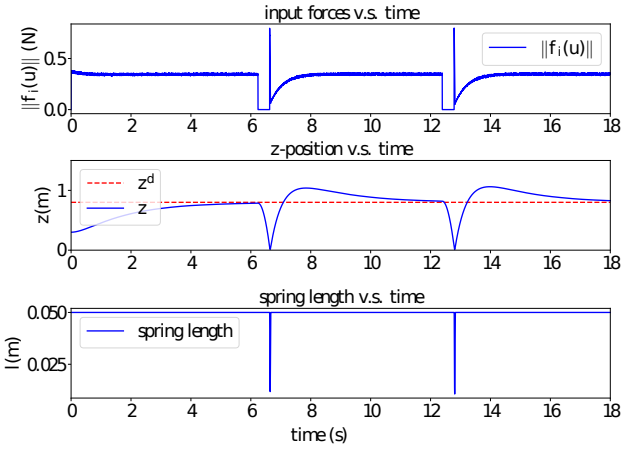

Fig. 7: The performance of the reference simulation, which saves 12% of the energy from hovering with two bounces. The numerical values of the four factors are: noise level = 0.2, spring constant = 400 N/m, damping factor = 1.0 Ns/m, hover height = 0.8 m.

closely reflected the behavior of our real spring. For each numerical value selection of a factor, we run the simulation 20 times for both normal hovering and bouncing mode while keeping all other factors invariant, and compare (6) from the two modes with $t_f = 18s$.

a) Noise level: In the experiments with real robots, we notice two unmodelled variants in Section III that keep the PogoDrone from achieving stability quickly. First, the uneven surface of the ground leads to the force received by the pogo having a perpendicular component to the pogo stick when touching the ground, causing large overshoots in xy-plane. Second, the force and torque generated by the propellers deviate from the desired input value f_q and $\boldsymbol{\tau}$ because of the imperfect motors, adding up to the loss of energy in achieving stability. As a result, the ascend phase lasts longer which keeps the motors actuating and increases the energy consumption. To emulate the errors from the input, we add Gaussian noise with standard deviation σ to the propeller forces $f_i(\boldsymbol{u})$. For the imperfect ground, we add a Gaussian noise with standard deviation 5σ to the direction of the spring force such that the spring force aligns with $\mathrm{Rot}_z(G)\,{}^W\boldsymbol{R}_B\boldsymbol{e}_3$, where $G \sim \mathcal{N}(0, (5\sigma)^2)$, instead of ${}^W\boldsymbol{R}_B\boldsymbol{e}_3$. We show in Fig. 8 that as the noise level increases, the energy consumption of both modes increases, and for the bouncing mode, it increases faster until catching up with the hovering mode. When the noise level is sufficiently high, the PogoDrone never achieves stability, making the propellers always actuated.

b) Spring constant: The spring constant determines the stiffness of the spring. In Fig. 9, we show the pattern of how the energy consumption changes as the spring constant gets higher. When the spring constant is low, the energy consumption decreases as the spring constant goes up because the spring can store an increasing amount of

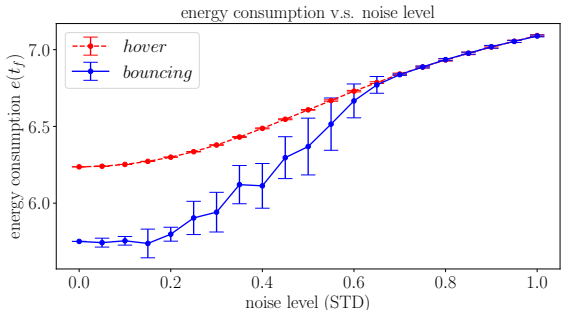

Fig. 8: Energy consumption v.s. noise level. A Gaussian noise is applied on the thrust forces of the propellers and the force direction of the pogo, emulating imperfect motors and ground surface.

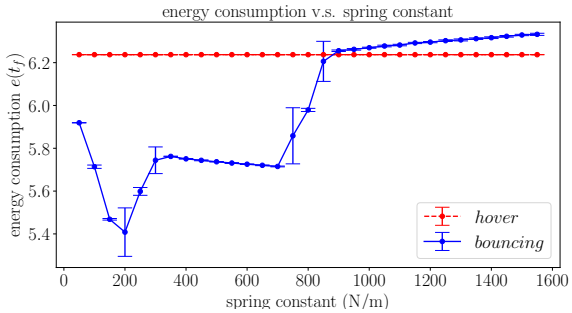

Fig. 9: Energy consumption v.s. spring constant. The higher spring constant, the more energy can be stored in the pogo and the stiffer the pogo is.

energy that would otherwise be lost during the hard impact. When the spring constant goes higher, the controller over-compensates the position in z, as shown in Fig. 7. From 200 to 300 N/m is a critical interval (same as 700 to 900 N/m) because the number of jumps in the time interval can change depending on the noise. Note that between 300 and 700 N/m the energy consumption remains stable because the bouncing behavior couples with the controller tuning. After 900, the energy consumption slowly increases due to the damping term that prevents the spring from rebounding faster. When the spring constant is sufficiently high, then energy spent on recovering stability surpasses the energy conserved by the spring, which supports our hypothesis. The energy consumption of hovering is not affected because the pogo does not make contact with the ground.

c) Damping factor: In Section III, we model the pogo as an imperfect spring, which induces energy loss from mechanical motion. The loss is in form of a damping term, which prevents the pogo from being compressed or rebounding quickly and therefore limits the maximum velocity of the PogoDrone after the rebound. Fig. 10 shows that as the damping factor increases, the energy consumption of the bouncing mode first decreases, then increases and stabilizes at a value lower than that of hovering. We notice that with a perfect spring whose damping factor is ignored or very small, the energy consumption becomes even higher than

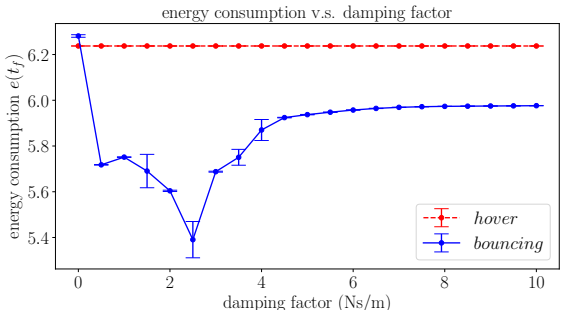

Fig. 10: Energy consumption v.s. damping factor.

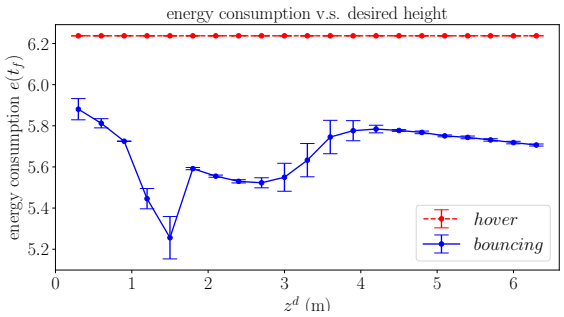

Fig. 11: Energy consumption v.s. desired height. The initial desired height determines the initial potential energy.

that of hovering. In theory, when a perfect spring is inserted in the pogo, to recover the same height before falling, the propellers do not need to actuate, since the elastic potential energy stored in the spring is equal to the gravitational energy before falling. However, as described in Section IV, the BHC activates the propellers to drive the drone to the desired height z^d upon entering rebound phase despite the state of the pogo. Thus, the redundant energy, provided by the propellers, creates an overshoot in the height, requiring more energy to compensate.

d) Desired height: The height of the desired position $\boldsymbol{r}^d$ determines the maximum amount of energy that needs to be stored in the pogo during a bounce. As shown in Fig. 11, when the desired height is below 1.5 m, the energy consumption of the PogoDrone is high because of the overshoots created by the controller. As the desired height goes higher, the energy consumption goes down to the low point at 1.5 m when the spring reaches maximum compression when storing all gravitational potential energy. The sudden change between 1.5 m and 2 m is caused by the energy loss due to hard impacts. After 2 m, the trend acts similar to Fig. 9. Our simulations verify that energy efficiency is strongly dependent on the spring constant, noise level, and damping coefficient. Fine tuning of the system achieves an optimal amount of energy saved but the jumps should perform under very specific conditions.

B. Experiments with an actual robot

We studied the physical system's efficiency and dynamic performance through five alterations of the desired height. In

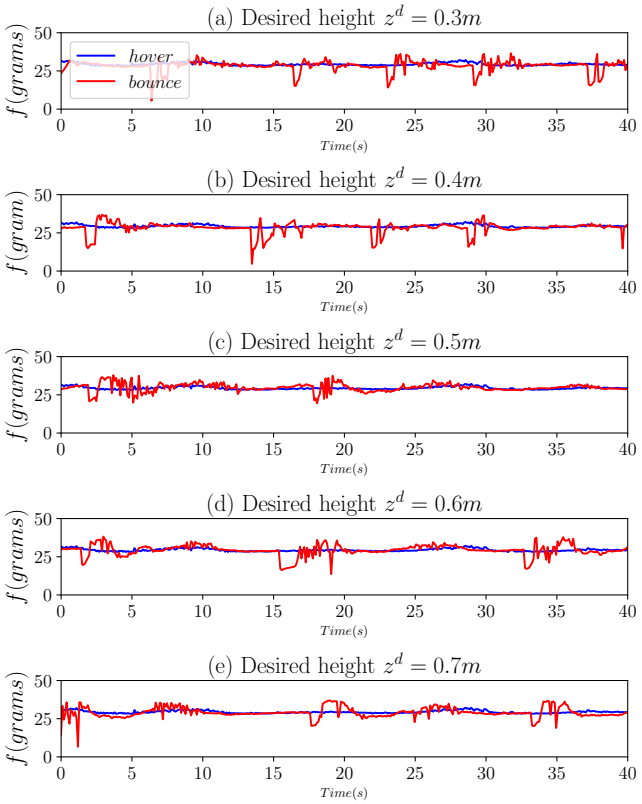

Fig. 12: Motor power of the system while hovering and bouncing at each tested height. The top graph shows 0.3 m and each graph below represents the height incremented by 0.1 m.

our testbed, we used our robot design, described in Sec. II, and the Crazyflie-ROS package to control the robot's actions. This package communicates with our Optitrack motion capture system, operating at 60 Hz. Our computer runs off-board position and attitude controls through ROS nodes, and these commands are then received by the PogoDrone via a 2.4 GHz radio. Fig. 12 shows the force of the motors over time for each height. We find that the optimal height for this bouncing system is at 0.3 m. At this height, energy is not lost to over-compression and the time of recovery in between the bouncing cycle is minimal. As we increase the height, we find that our spring is not able to store the entirety of the potential energy which leads to over-compression. Similar to the simulations, the hard impact with the ground requires the motors to compensate more due to the energy loss. Additionally, the system may generate a torque along the x-axis or y-axis. This leads to an imperfect bounce which further extends recovery time. At our inflection point of 0.4 m, we find that the culmination of these negative forces reward us with only a slight advantage in efficiency while bouncing. At 0.5 m, we find that bouncing performs worse than simply hovering. This is primarily due to over-compression, imperfect bounces, and significant recovery time. At 0.6 and 0.7 m, the efficiency performance is approximately equal for hovering and bouncing. This is caused by an extremely high time of recovery in-between bounces. Over the same time cycle, the PogoDrone spends most of the time recovering which leads to comparable energy performance. Similar to the previous experiments, these high recovery times are caused by over-compression and imperfect bounces along the z-axis. In addition, as the PogoDrone falls from higher heights, our attitude controller must compensate for more air resistance that wishes to create a torque on our system. We assumed a perfectly smooth and level surface to bounce on in our experiments. Our surface additionally is rigid which decreases the amount of energy loss per bounce due to damping or absorption.

VI. Conclusion and Future Work

This paper introduces PogoDrone, a robot that is able to jump and fly. We designed a prototype that works in actual environments. Using the modeled dynamics, we simulated and analyzed the behavior of the robot after changing different factors. We verified the concept of a dual locomotion quadrotor. Our robot is able to rapidly impact with the ground plane and return to a hover state, making it uniquely equipped to handle operations such as taking soil samples or planting seeds. Other multi-rotor vehicles would have to operate close to the ground for a prolonged period of time, making them susceptible to the ground effect. We found that the robot can quickly become energy inefficient when the spring constant or the actuators' noise is high. In conclusion, the robot has the potential to be considerably more efficient than a hovering robot (>20%), but it requires finding an optimal hardware setup and control gains for the actual robot.

Our robot has the potential to efficiently operate in low heights, so our future work is focused on examining the system's performance on imperfect surfaces which should be explored for real-world applications.

References

[1] Z. Zhang, J. Zhao, H. Chen, and D. Chen, "A survey of bioinspired jumping robot: Takeoff, air posture adjustment, and landing buffer," *Applied Bionics and Biomechanics*, vol. 2017, pp. 1–22, 09 2017.

[2] N. T. Truong, H. V. Phan, and H. C. Park, "Design and demonstration of a bio-inspired flapping-wing-assisted jumping robot," *Bioinspiration & Biomimetics*, vol. 14, no. 3, p. 036010, mar 2019.

[3] M. Noh, S.-W. Kim, S. An, J.-S. Koh, and K.-J. Cho, "Flea-inspired catapult mechanism for miniature jumping robots," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1007–1018, 2012.

[4] M. A. Woodward and M. Sitti, "Design of a miniature integrated multi-modal jumping and gliding robot," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 556–561.

[5] M. Kovac, M. Fuchs, A. Guignard, J.-C. Zufferey, and D. Floreano, "A miniature 7g jumping robot," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 373–378.

[6] M. Kovač, Wassim-Hraiz, O. Fauria, J.-C. Zufferey, and D. Floreano, "The epfl jumpglider: A hybrid jumping and gliding robot with rigid or folding wings," in *2011 IEEE International Conference on Robotics and Biomimetics*, 2011, pp. 1503–1508.

[7] Y. Naidoo, R. Stopforth, and G. Bright, "Development of an uav for search & rescue applications," in *IEEE Africon'11*. IEEE, 2011, pp. 1–6.

[8] R. Lagring, S. Degraer, G. de Montpellier, T. Jacques, W. Van Roy, and R. Schallier, "Twenty years of belgian north sea aerial surveillance: A quantitative analysis of results confirms effectiveness of international oil pollution legislation," *Marine Pollution Bulletin*, vol. 64, no. 3, pp. 644–652, 2012.

[9] G. Loianno and V. Kumar, "Cooperative transportation using small quadrotors using monocular vision and inertial sensing," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 680–687, 2017.
[10] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. Kumar, "Influence of aerodynamics and proximity effects in quadrotor flight," in *Experimental robotics*. Springer, 2013, pp. 289–302.
[11] M. N. Boukoberine, Z. Zhou, and M. Benbouzid, "A critical review on unmanned aerial vehicles power supply and energy management: Solutions, strategies, and prospects," *Applied Energy*, vol. 255, p. 113823, 2019.
[12] *Design of a Fuel Cell Powered Blended Wing Body UAV*, ser. ASME International Mechanical Engineering Congress and Exposition, vol. Volume 1: Advances in Aerospace Technology, 11 2012.
[13] J. Ouyang, Y. Che, J. Xu, and K. Wu, "Throughput maximization for laser-powered uav wireless communication systems," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2018, pp. 1–6.
[14] 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 *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 5166–5172.
[15] F. Muttin, "Umbilical deployment modeling for tethered uav detecting oil pollution from ship," *Applied Ocean Research*, vol. 33, no. 4, pp. 332–343, 2011.
[16] K. P. Jain and M. W. Mueller, "Flying batteries: In-flight battery switching to increase multirotor flight time," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3510–3516.
[17] D. W. Haldane, J. K. Yim, and R. S. Fearing, "Repetitive extreme-acceleration (14-g) spatial jumping with salto-1p," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3345–3351.
[18] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.
[19] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5420–5425.
[20] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2520–2525.