Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-3 (2023) 175–180

An Energy Efficient Jumping Drone - A Simple Projectile Motion Approach

Shraddha Barawkar [*] **Manish Kumar** [**]

[*] *Department of Mechanical and Materials Engineering, University of Cincinnati, Cincinnati, OH 45221 (e-mail: barawksd@mail.uc.edu).*
[**] *Faculty of Mechanical and Materials Engineering, University of Cincinnati, Cincinnati, OH 45221 (e-mail: manish.kumar@uc.edu)*

Abstract: Jumping robots are interesting devices that offer several advantages in terms of navigation about cluttered environments. However, they present unique challenges with respect to their design and control. In this paper, we propose a design that uses four planar thrusters/propellers on a jumping robot. Such a system provides better maneuverability due to agility provided by the propellers to guide the motion. From energy consumption perspective, we use the gravity for free fall and a spring mechanism to execute the jumping motion without loss of much energy on impact. The primary contribution of this paper is developing a novel navigation and control method for such jumping robots to go to the desired goal. In this paper, we present a projectile motion planning approach for the control of the proposed system. We propose proportional (P) and proportional-derivative (PD) controllers that compute the launch velocity required for the jumping drone after impact with the ground to follow a projectile motion in each jump to reach the goal position. The jumping drone bounces after impact with the ground, and the drone is then actuated for a certain time till it attains the required launch velocity after which it is made to move freely under the influence of gravity. Such a system shows significant reduction of energy consumption (by 81.35%) as compared to a normal drone navigating to the same goal location. Simulation results validate the effectiveness of the proposed approach.

Keywords: Jumping robot, projectile motion, energy efficient, better maneuverability

1. INTRODUCTION

Jumping robots are unique robots utilized for numerous applications such as search and rescue, surveillance, environmental monitoring, mobile sensor networks, and space exploration (Zhao et al. (2012)). Jumping is mostly employed for overcoming obstacles in a cluttered environment (Kovac et al. (2008)). Extensive research has been carried out on jumping robots in literature and various designs are proposed, for example, by the authors in references Klemm et al. (2019); Yu and Iida (2013); Grimes and Hurst (2012). Bio-inspired jumping robots are also presented in Zhang et al. (2017, 2013). However, the major focus in existing research is on the jumping mechanism design and control. Though steerable jumping robots exist, very little focus has been laid on maneuverability/agility and energy efficiency. These two aspects in a jumping robot can be especially relevant for certain applications such as space exploration. This paper proposes a simple system of using propellers (such as in drones) for guiding the jumping robots. Such design has potential to improve maneuverability and lower energy consumption. Using four rotors/thrusters in a planar configuration, on a jumping robot, provides more motion flexibility to the robot as the drone can make the robot roll, pitch or yaw unlike other robots where navigation to a goal location requires series of lateral and longitudinal jumps. Such a system can exploit the features of a drone such as higher agility Kaufmann et al. (2022). The following paragraph provides the state-of-the-art on existing jumping robots.
The jumping robot presented in Kovac et al. (2008) can jump more than 27 times its own size and can adjust its jumping force and take-off angle. It consists of elastic elements in a four bar linkage leg system. Similarly, Kovač et al. (2011) present a hybrid jumping and gliding robot that can glide using folding or rigid wings prolonging the jumps of jumping robot. This aspect can easily be provided by the proposed design of jumping quadrotor drone. Reference Yim and Fearing (2018) presents the Salto-1P jumping robot which can jump over heights above $1m$ and can hop over $2m$ horizontally using the simple Raibert controller. Thrusters and a tail are used to control the robots orientation. A two-wheeled jumping robot Ascento is presented in reference Klemm et al. (2019). Wheels of the robot enable it to move rapidly on flat terrain, while jumping is performed for overcoming obstacles. LQR controller has been utilized to stabilize the robot. Reference Zhao et al. (2012) presents a small steerable jumping robot actuated by a single motor. The development and design of a long jumping Grillo Mini Robot is presented in Scarfogliero et al. (2007). A torsional spring is loaded by an actuated eccentric cam to move the rear legs of the robot and a feedforward controller is employed.
In context to the proposed work, reference Zhu et al. (2022) presents a similar concept termed as PogoDrone where a drone is attached to a passive jumping mechanism. However this concept is different from the proposed work in terms of control approach and motion planning. The PogoDrone Zhu et al. (2022) has to land vertically on the ground keeping the orientation of the drone to be 0. No such restriction is assumed in the proposed work. It should be noted that, we assume the landing mechanism to be such that the drone can bounce/land at any orientation. Moreover, the PogoDrone Zhu et al. (2022) is not energy efficient.

10.1016/j.ifacol.2023.12.020

The proposed system consists of use of a drone or quadrotor attached to a jumping/bouncing mechanism which forms as a simple jumping robot. Though any jumping mechanism can be employed to provide jumping motion to the robot, we focus on a ball/sphere like bouncing mechanism, where the drone's orientation can be nonzero during impact. First the robot is made to fall freely on ground. The drone is required to follow projectile motion in jump. After the impact with the ground (impact phase), the drone is actuated for a certain amount of time such that it attains the projectile velocity required to reach a certain height and range in each jump. This initial velocity for projectile motion is calculated during each jump by simple proportional (P) and proportional-derivative (PD) controllers. After the drone attains the velocity required to complete the projectile motion, it is switched off (or allowed to run at idle speed) and allowed to fall freely under the influence of gravity. Intuitively this approach would save immense energy as compared to a drone navigating to the same goal location. The cycle of impact, drone flight and projectile phases repeats till the robot reaches the goal location.
Though the controllers are simple, the idea of actuating the drone for only for a short amount time and then allowing it to fall freely under the influence of gravity, following a projectile motion, makes up for the energy efficient jumping drone. Such a design shows significant reduction in energy consumption of 80% in simulations. The novel control approach, motion planning for improved energy efficiency and the idea of the proposed jumping quadrotor drone form the contributions of this work.

2. PROBLEM FORMULATION

We consider the jumping drone is required to reach the goal position on ground $r_g = (x_g, y_g, z_g = 0)^T$. The primary objective is planning and executing the motion of the jumping drone such that minimum energy is consumed. In order to navigate the drone to its goal location with less consumption of energy, the control is provided just after each impact for t_{flight} amount of time. After t_{flight} time post each impact, the control is switched off so that the drone moves freely following a projectile motion due to gravity. The control objective in each impact is to help the drone achieve a desired velocity in horizontal direction and a desired speed in vertical direction so that it reaches a desired height and covers a certain projectile range when control is switched off which happens after t_{flight} time after the impact. Refer to Fig. 2. Let v^i_{xy} be the desired velocity, in i^{th} jump, in $X_W Y_W$ plane to reach the goal location (x_g, y_g). The first problem consists of computing v^i_{xy} which will reduce with relative position of the robot from the goal location $e = r_g - r$. Where, $r = (r_x, r_y, r_z)^T$ is the position of the jumping drone. Thus,

$$v^i_{xy} = f(e) \tag{1}$$

Now, let H_1 be the desired height to be attained by the robot in first jump, H_2 be the desired height to be attained by the robot in second jump and so on. The second problem is to compute this desired jump/hop height which will reduce with relative position of the robot from the goal location e at the beginning of i_{th} jump. Thus,

$$H_i = f(e, \dot{e}) \tag{2}$$

Here i denotes the count of jumps. The vertical velocity is then obtained as: $v^i_z = \sqrt{2gH_i}$. The above gives us vertical (v^i_z) and horizontal components (v^i_{xy} in $X_W Y_W$ plane) of the launch velocity (v^i_{launch}), also and the angle (α^i) of the desired projectile motion. The next objective is that of controlling the drone to attain the launch velocity for each jump.

3. DYNAMICS

This section briefly presents the dynamics of the jumping quadrotor robot.

3.1 Drone Dynamics

The Quadrotor dynamics has been presented in detail in Barawkar et al. (2017) and is indicated here briefly for completeness of the paper. The linear and rotational equations of motion of the drone are,

$$m\ddot{r} = R\begin{bmatrix} 0 \\ 0 \\ (F_1 + F_2 + F_3 + F_4) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{3}$$

$$I\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I\begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4}$$

where m and g are the mass of the jumping drone and the acceleration due to gravity. F_1, F_2, F_3 and F_4 are the rotor thrust forces and M_1, M_2, M_3 and M_4 are the rotor moments generated by the drone. L is the distance between the drone and rotor centers. I is the moment of inertia of the drone. The angular velocity components of the drone in the body frame are p, q and r. The rotation matrix R from the body $X_B Y_B Z_B$ to the world frame $X_W Y_W Z_W$ is,

$$R = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + s\phi s\psi c\theta \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} \tag{5}$$

where ϕ, θ and ψ are the pitch, roll and yaw angles representing the drone's orientation. $c(.)$ and $s(.)$ represent the cosine and sine terms of the respective angles.

3.2 Jumper Dynamics

The impact dynamics of jumping robot is modeled as Spring-Mass-Damper system, which remains in effect till the robot maintains contact with the ground. The robot is assumed to have stiffness of K and damping of C. We assume an elastic spherical enclosure (or ball) around the jumping drone as shown in Fig. 2 acting as a spring-mass-damper system. We assume the quadrotor is fixed at the center of the bouncing ball, thus no torque is generated on the drone during impact. The acceleration of the jumper $\ddot{r}_{j,z}$ along vertical Z_W axis is then given by,

$$\ddot{r}_{j,z} = -g + \frac{K}{m}\Delta l + \frac{C}{m}\dot{\Delta l} \tag{6}$$

where, Δl is the corresponding deformation of the spring mass damper system.

4. APPROACH

We now describe our approach for the objectives listed in the Problem Formulation section. In order to navigate the energy efficient jumping drone to its goal location, the drone's controller is engaged after each impact for Δt amount of time, following which, the controller is switched off so that drone

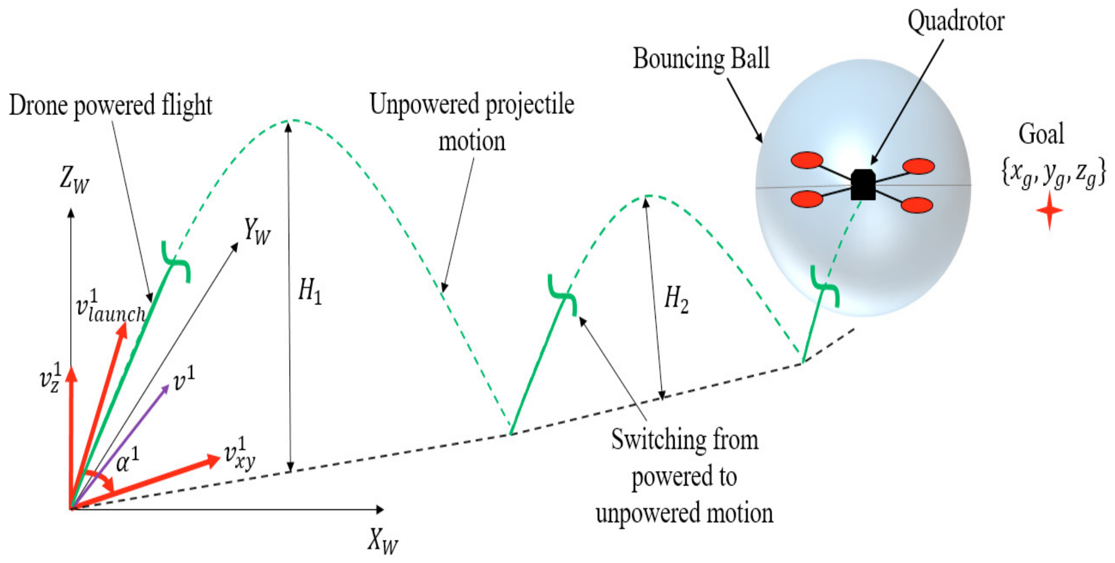

Fig. 1. Schematic diagram of the energy efficient system.

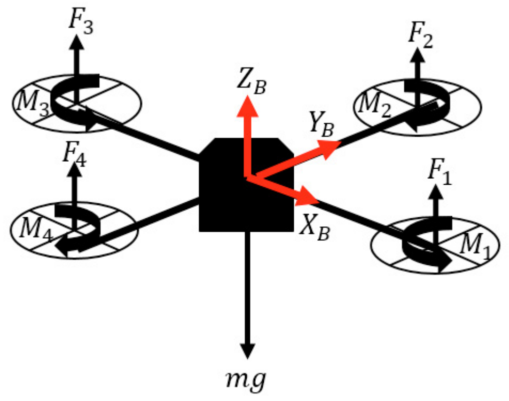

Fig. 2. Free body diagram of the drone.

moves freely following a projectile motion due to gravity. This switching on and off technique of the jumping drone saves immense energy. This section now deals with calculation of v_{launch} for the implementation of projectile motion during each jump of the energy efficient jumping robot.

4.1 Free fall, Compression and Expansion

Initially the robot is made to fall freely from a certain height. After impact the bouncing ball or the landing mechanism undergoes deformation (compression and expansion) according to the spring mass damper equation defined in **??**.

4.2 Projectile Motion Planning

The desired velocity of the system required to reach the goal location in $X_W Y_W$ plane v^i_{xy}, in each jump, is computed using a simple proportional controller as follows,

$$v^i_{xy} = k_{p,v} e_{xy} \tag{7}$$

where $k_{p,v}$ is a proportional constant and e_{xy} is given as,

$$e_{xy} = \begin{bmatrix} x_g & y_g \end{bmatrix}^T - r_{xy} \tag{8}$$

where $r_{xy} = \begin{bmatrix} r_x & r_y \end{bmatrix}^T$ is the 2D position of the robot. Next, the hop height during each jump is calculated as a function of the error e_{xy} as follows,

$$H_i = k_{p,h}||e_{xy}|| + k_{d,h}||\dot{r}_{xy}|| \tag{9}$$

where, $k_{p,h}$ and $k_{d,h}$ are the PD gains for the hop height controller. The desired vertical velocity v^i_z can now be calculated as follows from projectile motion kinematics:

$$v^i_z = \sqrt{2gH_i} \tag{10}$$

It should be noted that both v^i_{xy} and H_i reduce as the robot goes near the goal location. The required launch velocity and the projectile angle α^i are now given as:

$$v^i_{launch} = \begin{bmatrix} v^i_{xy} & v^i_z \end{bmatrix}^T \tag{11}$$

$$\alpha^i = tan^{-1}\frac{v^i_z}{v^i_{xy}} \tag{12}$$

4.3 Drone Powered Flight Control

Next, we describe our approach to control the robot or drone so that it achieves the launch velocity in δt amount of time. The drone control design is similar to that presented in Barawkar et al. (2018) with few modification. A simple proportional controller is utilized to implement velocity control to attain v^i_{launch} in each jump. The control law for the velocity control is,

$$\ddot{r}^{des} = k_{p1}(v^i_{launch} - v) \tag{13}$$

where k_{p1} is the P gain and v is the current velocity of the jumping drone. $\ddot{r}^{des}$ is the desired linear acceleration of the drone with subscripts 'x', 'y' and 'z' denoting its 3D components. The desired roll ϕ^{des} and pitch θ^{des} angles for the drone are then computed as,

$$\phi^{des} = \frac{1}{g}(\ddot{r}^{des}_x \sin\psi_T - \ddot{r}^{des}_y \cos\psi_T) \tag{14}$$

$$\theta^{des} = \frac{1}{g}(\ddot{r}^{des}_x \cos\psi_T + \ddot{r}^{des}_y \sin\psi_T) \tag{15}$$

where, ψ_T denotes the desired yaw angle at the goal location. $\Delta\omega_F$ is the change in rotor speeds required to produce acceleration along the Z_W axis. It is,

$$\Delta\omega_F = \frac{m\ddot{r}_z^{des}}{8k_F\omega_h} \tag{16}$$

$k_F = 2.2 \times 10^{-4}$ is the motor constant of the drone and ω_h is the hovering speed given by,

$$\omega_h = \sqrt{\frac{mg}{4k_F}} \tag{17}$$

The desired roll, pitch and $\Delta\omega_{Fl}$ are used to control the drone's position along X_W, Y_W and Z_W axes. It should be noted that the desired yaw angle ψ_{des} is assumed to be zero throughout this work. Deviations of angular speeds of the rotors from ω_h are used to navigate the drone along 3D axes. These deviations are $\Delta\omega_F$, $\Delta\omega_\phi$, $\Delta\omega_\theta$ and $\Delta\omega_\psi$. $\Delta\omega_\phi$, $\Delta\omega_\theta$ and $\Delta\omega_{\psi l}$ provide roll, pitch and yaw to the drone. $\Delta\omega_F$ provides motion along the Z_W axis. These deviations for the attitude control of the drone are computed as,

$$\Delta\omega_\phi = k_{p,\phi}(\phi^{des} - \phi) + k_{d,\phi}(p^{des} - p) \tag{18}$$

$$\Delta\omega_\theta = k_{p,\theta}(\theta^{des} - \theta) + k_{d,\theta}(q^{des} - q) \tag{19}$$

$$\Delta\omega_\psi = k_{p,\psi}(\psi^{des} - \psi) + k_{d,\psi}(r^{des} - r) \tag{20}$$

p^{des}, q^{des} and r^{des} denote the desired angular velocities of the drone. k_p and k_d with suffices ϕ, θ and ψ for roll, pitch and yaw angles, are the PD gains of the attitude controller of the drone. The desired rotor speeds of the drone are then calculated as,

$$\begin{bmatrix} \omega_1^{des} \\ \omega_2^{des} \\ \omega_3^{des} \\ \omega_4^{des} \end{bmatrix}_l = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \omega_h + \Delta\omega_F \\ \Delta\omega_\phi \\ \Delta\omega_\theta \\ \Delta\omega_\psi \end{bmatrix} \tag{21}$$

The individual rotor thrust forces and moments are,

$$F_i = k_F\omega_i^2 \tag{22}$$

$$M_i = k_M\omega_i^2 \tag{23}$$

The constant k_M was taken as 5.4×10^{-6}.

Refer to Algorithm 1 for further details of this section. r_z denotes the vertical position component of the drone. l denotes the length of the spring mass damper system and Δl_{max} is the maximum deformation of the bouncing ball. $Case_f$ and $Case_c$ denote the free fall and compression phases of the system. t_{flight} denotes the time for which the drone is actuated during each jump. ϵ is the minimum threshold for the norm of the error e_{xy} up to which the robot will be activated. $\Sigma F = F_1 + F_2 + F_3 + F_4$ is the summation of the rotor thrust forces of the drone.

5. RESULTS

MATLAB was utilized to simulate the system. The jumping drone is initially made to fall freely from a height of 3m. The parameters use in simulation are,
$r_g = \begin{bmatrix}1 & 2 & 0\end{bmatrix}^T$, $m = 1$kg, $L = 0.12$m, $l = 0.25$m, $k_{p1} = \begin{bmatrix}2 & 2 & 4\end{bmatrix}^T$, $k_{p,\phi} = k_{p,\theta} = 100$, $k_{p,\psi} = 40$, $k_{d,\phi} = k_{d,\theta} = k_{d,\psi} = 20$, $k_{p,h} = k_{d,h} = 0.1$, $k_{p,v} = \begin{bmatrix}0.6 & 0.4\end{bmatrix}^T$, $t_{flight} = 0.75$ secs, $\Delta l_{max} = 0.08$m, $\epsilon = 0.2$, $K = 100$N/m, $C = 19$N-sec/m.

Algorithm 1 Algorithm of Energy Efficient Jumping Drone

```
Require: $r_z >> l$
 1: $Case_f \leftarrow 1, Case_c \leftarrow 1, launch \leftarrow 1$
 2: if $Case_f = 1$ and $r_z > l$ then                              ▹ Free fall
 3:     $\ddot{r}_z \leftarrow -g$
 4:     Update position and orientation
 5: else if $Case_c = 1$ and $r_z < l$ then                         ▹ Compression
 6:     $\ddot{r}_z \leftarrow \ddot{r}_{j,z}, t \leftarrow 0$
 7:     Update position and orientation
 8:     if $r_z < l - \Delta l_{max}$ then
 9:         $Case_f \leftarrow 2, Case_c \leftarrow 2$
10:     end if
11: else
12:     Update $e_{xy}$ and $r_{xy}$
13:     if $r_z < l - \Delta l_{max}$ then                          ▹ velocity direction change
14:         $\dot{r}_z \leftarrow -\dot{r}_z$
15:     end if
16:     if $r_z < l$ then                                           ▹ Expansion
17:         $\ddot{r}_1 \leftarrow \begin{bmatrix}0 & 0 & \ddot{r}_{j,z}\end{bmatrix}^T$
18:     else
19:         $\ddot{r}_1 \leftarrow \begin{bmatrix}0 & 0 & -g\end{bmatrix}^T$
20:     end if
21:     if $t = 0$ then
22:         Update $v_{xy}^i$, $H_i$, $v_z^i$ and $v_{launch}^i$
23:     end if
24:     if $||e_{xy}|| > \epsilon$ and $t < t_{flight}$ and $launch = 1$ then ▹ Drone flight phase
25:         $\ddot{r}^{des} \leftarrow k_{p1}(v_{launch}^i - \dot{r})$
26:         Compute drone's linear acceleration $\ddot{r}$, then
27:         $\ddot{r} \leftarrow R\begin{bmatrix}0 & 0 & \frac{\Sigma F}{m}\end{bmatrix}^T + \ddot{r}_1$
28:         Compute drone's angular acceleration $(\dot{p}, \dot{q}, \dot{r})^T$
29:         Update position and orientation
30:     else
31:         $launch = 0$
32:         Update position and orientation
33:     end if
34:     if $||e_{xy}|| > \epsilon$ and $launch \leftarrow 0$ then   ▹ Projectile phase
35:         Update $r$ according to projectile motion
36:     end if
37:     if $||e_{xy}|| > \epsilon$ and $r_z < l$ then
38:         $Case_c \leftarrow 1, launch \leftarrow 1$
39:     end if
40:     Update time $t$
41: end if
```

Fig. 3 shows the position and orientation of the system. The jumping drone shows effective waypoint navigation. Fig. 4 shows that the drone (when it is activated) tracks the desired launch velocity. Fig. 5 shows the different phases of the robot such as free fall, compression, expansion, drone flight and projectile phases in vertical Z direction. Fig. 6 shows the three dimensional plot of the system showing the trajectory of the robot towards the goal location. Similar to Zhu et al. (2022) we calculate the energy consumption E of the jumping drone and a normal drone navigating to the same waypoint in the time interval $[0, t_f]$ as follows,

$$E = \int_0^{t_f} \sum_{i=1}^{4} F_i(t)dt \tag{24}$$

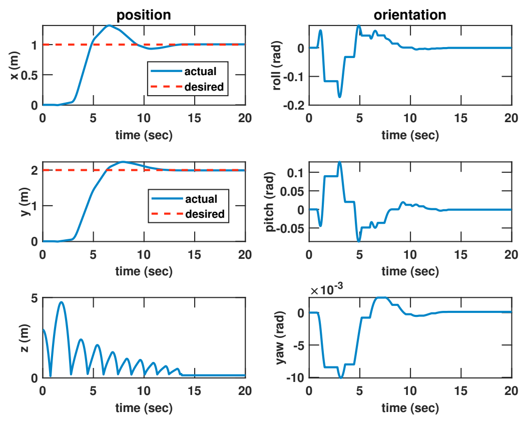

Fig. 3. Position and orientation of the Jumping Drone.

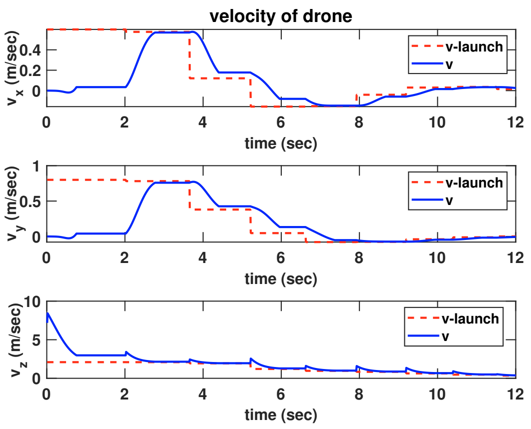

Fig. 4. Velocity of Jumping Drone during flight phase.

We use the above equation as performance metric to calculate energy consumption since the current is proportional to the propeller force. t_f denotes the time taken by the robots to reach the goal location. It was found that the energy consumption of the jumping drone was 5.49kN while that of a normal drone (without jumping mechanism) traveling to the same waypoint was 29.43kN. This shows that there is 81.35% energy consumption reduction using the proposed jumping quadrotor system as compared to a flying drone. This reduction is significant and it makes the system energy efficient.

6. CONCLUSION

In this paper, we propose a design of a jumping robot in which propellers are used to actuate a part of the motion. This paper primarily addresses the problem of navigation and control of such a robot. A projectile motion planning approach for the control of the proposed system is presented. Simple proportional (P) and proportional-derivative (PD) controllers are used to compute the projectile launch velocity required for the jumping drone to reach the goal position in each jump. The controller is designed in such a manner that the launch velocity reduces with the distance of the jumping drone from the goal location. Initially, the jumping drone is released to fall freely from a certain height. After impact with ground, the jumping drone is actuated for a certain time till it attains the required projectile launch velocity. Such a system shows significant

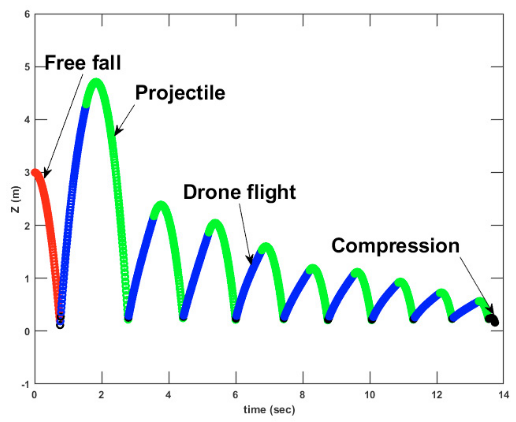

Fig. 5. Trajectory of the system along Z_W axis showing different phases such as free fall (red), compression (black), drone powered flight (blue) and unpowered projectile motion (green).

improvement in reduction of energy consumption (by 81.35%) as compared to a normal drone navigating to the same goal location for a typical goal-point navigation case. Simulation results validate the effectiveness of the proposed approach in both navigation as well as energy conservation.

REFERENCES

Barawkar, S., Radmanesh, M., Kumar, M., and Cohen, K. (2017). Admittance based force control for collaborative transportation of a common payload using two uavs. In *Dynamic Systems and Control Conference*, volume 58295, V003T39A007. American Society of Mechanical Engineers.

Barawkar, S., Radmanesh, M., Kumar, M., and Cohen, K. (2018). Fuzzy logic based variable damping admittance control for multi-uav collaborative transportation. In *2018 Annual American Control Conference (ACC)*, 2084–2089. IEEE.

Grimes, J.A. and Hurst, J.W. (2012). The design of atrias 1.0 a unique monopod, hopping robot. In *Adaptive Mobile Robotics*, 548–554. World Scientific.

Kaufmann, E., Bauersfeld, L., and Scaramuzza, D. (2022). A benchmark comparison of learned control policies for agile quadrotor flight. *arXiv preprint arXiv:2202.10796*.

Klemm, V., Morra, A., Salzmann, C., Tschopp, F., Bodie, K., Gulich, L., Küng, N., Mannhart, D., Pfister, C., Vierneisel, M., et al. (2019). Ascento: A two-wheeled jumping robot. In *2019 International Conference on Robotics and Automation (ICRA)*, 7515–7521. IEEE.

Kovač, M., Fauria, O., Zufferey, J.C., Floreano, D., et al. (2011). The epfl jumpglider: A hybrid jumping and gliding robot with rigid or folding wings. In *2011 IEEE International Conference on Robotics and Biomimetics*, 1503–1508. IEEE.

Kovac, M., Fuchs, M., Guignard, A., Zufferey, J.C., and Floreano, D. (2008). A miniature 7g jumping robot. In *2008 IEEE international conference on robotics and automation*, 373–378. IEEE.

Scarfogliero, U., Stefanini, C., and Dario, P. (2007). Design and development of the long-jumping" grillo" mini robot. In *Proceedings 2007 IEEE International conference on robotics and automation*, 467–472. IEEE.

Yim, J.K. and Fearing, R.S. (2018). Precision jumping limits from flight-phase control in salto-1p. In *2018 IEEE/RSJ*

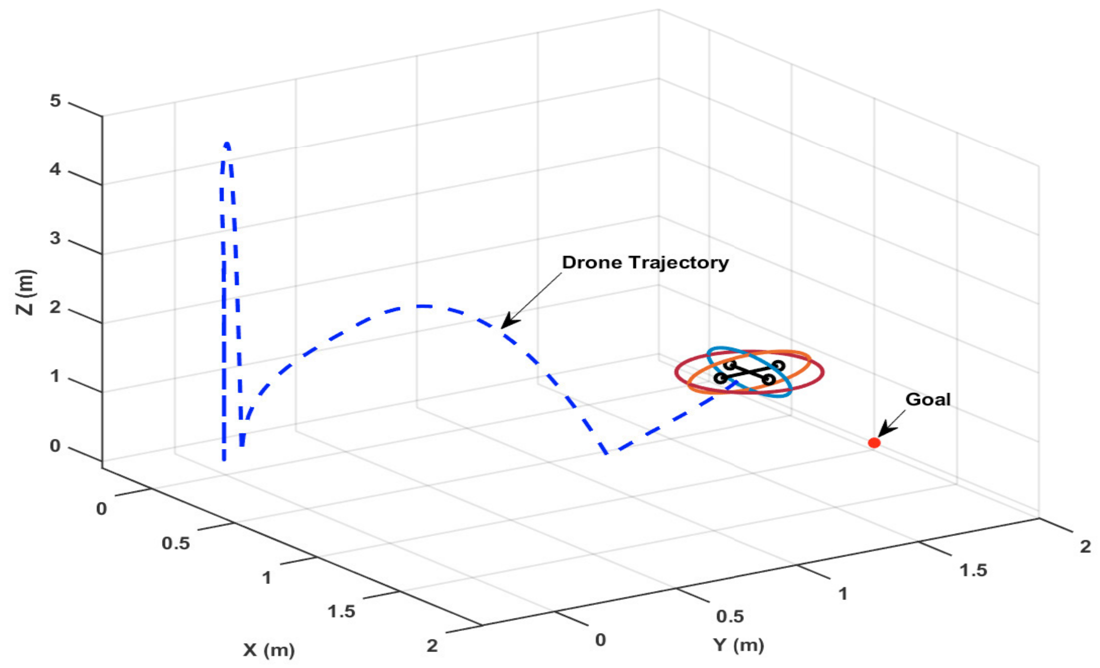

Fig. 6. 3D plot of the system showing the trajectory.

international conference on intelligent robots and systems (IROS), 2229–2236. IEEE.

Yu, X. and Iida, F. (2013). Minimalistic models of an energy-efficient vertical-hopping robot. *IEEE Transactions on Industrial Electronics*, 61(2), 1053–1062.

Zhang, J., Song, G., Li, Y., Qiao, G., Song, A., and Wang, A. (2013). A bio-inspired jumping robot: Modeling, simulation, design, and experimental results. *Mechatronics*, 23(8), 1123–1140.

Zhang, Z., Zhao, J., Chen, H., and Chen, D. (2017). A survey of bioinspired jumping robot: takeoff, air posture adjustment, and landing buffer. *Applied bionics and biomechanics*, 2017.

Zhao, J., Xi, N., Cintrón, F.J., Mutka, M.W., and Xiao, L. (2012). A single motor actuated miniature steerable jumping robot. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4274–4275. IEEE.

Zhu, B., Xu, J., Charway, A., and Saldaña, D. (2022). Pogodrone: Design, model, and control of a jumping quadrotor. In *2022 International Conference on Robotics and Automation (ICRA)*, 2031–2037. IEEE.