



OPEN ACCESS

EDITED BY

Egidio Falotico,
Sant'Anna School of Advanced Studies,
Italy

REVIEWED BY

Yugang Liu,
Royal Military College of Canada
(RMCC), Canada
Erik Prada,
Technical University of Košice, Slovakia

*CORRESPONDENCE

Tianqi Ma,
mtq19@mails.tsinghua.edu.cn

SPECIALTY SECTION

This article was submitted to
Bio-Inspired Robotics,
a section of the journal
Frontiers in Robotics and AI

RECEIVED 29 August 2022

ACCEPTED 28 October 2022

PUBLISHED 24 November 2022

CITATION

Ma T, Zhang T and Ma O (2022), On the
dynamics and control of a squirrel
locking its head/eyes toward a fixed
spot for safe landing while its body is
tumbling in air.

Front. Robot. AI 9:1030601.

doi: 10.3389/frobt.2022.1030601

COPYRIGHT

© 2022 Ma, Zhang and Ma. This is an
open-access article distributed under
the terms of the [Creative Commons
Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use,
distribution or reproduction in other
forums is permitted, provided the
original author(s) and the copyright
owner(s) are credited and that the
original publication in this journal is
cited, in accordance with accepted
academic practice. No use, distribution
or reproduction is permitted which does
not comply with these terms.

On the dynamics and control of a squirrel locking its head/eyes toward a fixed spot for safe landing while its body is tumbling in air

Tianqi Ma^{1*}, Tao Zhang¹ and Ou Ma²

¹Department of Automation, Tsinghua University, Beijing, China, ²Department of Aerospace and Engineering Mechanics, University of Cincinnati, Cincinnati, OH, United States

An arboreal mammal such as a squirrel can amazingly lock its head (and thus eyes) toward a fixed spot for safe landing while its body is tumbling in air after unexpectedly being thrown into air. Such an impressive ability of body motion control of squirrels has been shown in a recent YouTube video, which has amazed public with over 100 million views. In the video, a squirrel attracted to food crawled onto an ejection device and was unknowingly ejected into air by the device. During the resulting projectile flight, the squirrel managed to quickly turn its head (eyes) toward and then keeps staring at the landing spot until it safely landed on feet. Understanding the underline dynamics and how the squirrel does this behavior can inspire robotics researchers to develop bio-inspired control strategies for challenging robotic operations such as hopping/jumping robots operating in an unstructured environment. To study this problem, we implemented a 2D multibody dynamics model, which simulated the dynamic motion behavior of the main body segments of a squirrel in a vertical motion plane. The inevitable physical contact between the body segments is also modeled and simulated. Then, we introduced two motion control methods aiming at locking the body representing the head of the squirrel toward a globally fixed spot while the other body segments of the squirrel were undergoing a general 2D rotation and translation. One of the control methods is a conventional proportional-derivative (PD) controller, and the other is a reinforcement learning (RL)-based controller. Our simulation-based experiment shows that both controllers can achieve the intended control goal, quickly turning and then locking the head toward a globally fixed spot under any feasible initial motion conditions. In comparison, the RL-based method is more robust against random noise in sensor data and also more robust under unexpected initial conditions.

KEYWORDS

squirrel flight behavior, righting reflex, body tumbling, attitude control, reinforcement learning

1 Introduction

The righting reflex of animals is the ability to correct their body posture (orientation) from an abnormal posture on emergency. A well-known example is that when a cat falls from a high position, it can always manage to change its body orientation and land right on its feet even if it falls from an upside-down posture (Kane and Scher, 1969). A similar phenomenon has also been observed on many other species, such as lizards (Sinervo and Losos, 1991; Schlesinger et al., 1993) and rats (Laouris et al., 1990). Such a mid-air body posture control ability helps animals reduce the risk of body injury from a free fall. This kind of amazing body posture control (attitude control) behavior of cats and other animals is difficult to understand due to the conservation of momentum, which tells us that in a free fall, nobody (regardless a human or animal) can change their angular momentum by moving body segments (head, limbs, tail, etc.) because all the efforts generate internal forces/moments only. Therefore, the well-known cat free-fall problem has motivated many research efforts in the past, and the findings have been well documented (Fernandes et al., 1993; Arabyan and Tsai, 1998; Weng and Nishimura, 2000; Xu et al., 2012). Arboreal mammals such as squirrels face higher risk of unexpected falls than cats (Young and Chadwell, 2020) due to their natural habitats and aggressive activities. Squirrels are hardly injured by falls from height because of their impressive self-control ability of their body posture. In a highly popular

YouTube video (Rober, 2020), squirrels were lured by food onto an ejection device and they were suddenly ejected into air (as shown in Figure 1). Obviously, the ejection caused their bodies to tumble (general 3D rotation) and free fall in air. Amazingly, an ejected squirrel could quickly (within 0.1 s) turn its head toward where it would land and then lock its head (thus eyes) to that direction for staring at the landing spot, while the rest of its body continued tumbling in air. At the end, the squirrel always landed at its stared landing spot with its feet touching down first. The squirrels' such amazing behavior of locking head toward a fixed spot understandingly is for better situation awareness and safe landing. However, the question of how they can achieve that capability from dynamics and control perspective has not been studied in the literature. Our research is to address this problem.

Understanding how animals control their body postures in the absence of external forces (except gravity) is not only driven by our scientific curiosity but also highly motivated by engineering needs, especially robotics. Understandably, the principles of animals controlling their body postures using their body part movements can be applied to robots for achieving similar or even better behavior. In this regard, research works have been conducted (Shuster et al., 1993; Tsiotras, 1996). For example, controlling body posture without external force is necessary in some special task environments such as space, disaster scene (Gonzalez et al., 2020), and hopping robots (Yim et al., 2020). Damage by falls



FIGURE 1 Squirrel is able to lock its head (and eyes) toward its landing spot while the rest of its body is tumbling after unknowingly ejected into air by the ejection platform [snapshots from the video (Rober, 2020)].

(Cameron and Arkin, 1992) could also be prevented to extend servicing and lifespan of robots (Bingham et al., 2014).

The research inspired by animal motions to control body postures can be divided into the following three aspects:

- 1) Using tails to control their body orientation: This kind of research particularly focuses on imitating lizards (Jusufi et al., 2010; Libby et al., 2012; Clark et al., 2021) and geckos (Jusufi et al., 2008; Siddall et al., 2021). These kinds of arboreal animals often have a tail with large moment of inertia and a body without obvious internal motions. Therefore, the lizards/geckos can reorient or stabilize the body by moving the tail (Wenger et al., 2016) and they can be simply modeled as two linked rigid bodies. The tail is known to take a role in controlling the pitch angle of the lizards/geckos in the air for landing (Libby et al., 2012; Siddall et al., 2021), air-righting like a cat (Jusufi et al., 2010, 2008). For gliding lizards, tails also help adjusting the angle of attack to improve both glide distance and stability (Clark et al., 2021). Similar work on squirrels (Fukushima et al., 2021) also explained the stability using the motion of tails in unexpected falls. Special tails were added to help robot jumping (Zhao et al., 2013), insect-sized robot achieving more rapid orientation (Singh et al., 2019), and more complicated tails (e.g., a three-segment prototype (Liu and Ben-Tzvi, 2020)) and soft tails (Butt et al., 2021) were designed to perform as real tails, which is not rigid in reality.
- 2) Using the whole body to reorient body posture (or attitude) (Kane and Scher, 1969; Sadati and Meghdari, 2017; Liu et al., 2020; Yim et al., 2020): In this kind of research, models do not need a special part that absorbs extra angular momentum, instead, the control strategy is to redistribute the angular momentum to different parts of the body to achieve different goals. For example, cats do not necessarily need their tails to change their body orientation in air (Fredrickson, 1989). Hence, the control of a cat-like robot needs to consider all the joints of its body (Arabyan and Tsai, 1998). Flying snakes slide with aerial undulation to increase performance and could be another inspiration for dynamic flying robots (Yeaton et al., 2020).
- 3) Using the aerodynamic effect on the special structure of the body (Li et al., 2016; Norby et al., 2021): Since the motion is in air, robots can imitate the gliding principle of animals, such as flying squirrels. Using the aerodynamic effect wisely can control body attitude and reduce the energy consumption.

The squirrel behavior shown in the study proposed by Rober (2020) inspired this work. We focus on the control of turning and locking the head of a squirrel toward the landing spot when it is initially thrown into air and remains tumbling in air. We study the possibility of using joint motion control (like a robot) to regulate the head orientation. For easy understanding of the dynamics, we use planar multibody dynamics (in the pitch plane) to model a squirrel. Such a

2D modeling approach has been widely applied by other researchers for other bio-inspired robotics problems, such as unmanned aerial vehicle (Bouabdallah and Siegwart, 2007; Yilmaz et al., 2019; Zamora et al., 2020), and other tasks, such as trajectory tracking (Castillo-Zamora et al., 2019), grasping objects (Kobilarov, 2014; Zhao et al., 2018), and sports (table tennis) (Shi et al., 2019).

In the squirrel problem, there are three features in the motion of sudden ejection of the squirrel:

- 1) The ejection occurs suddenly so that there is no preparation time
- 2) The initial posture (initial motion conditions) is random
- 3) The flight duration is short, and thus, a timely response is required

These factors contribute to the challenge of the investigated motion control. In our study, we apply reinforcement learning (RL) technology to train a neural network-based control algorithm to deal with the non-linear dynamics and control problem just as those having been tried in different bio-inspired robots (Dooraki and Lee, 2019; Li et al., 2019; Kamali et al., 2020). At the same time, we also apply a traditional PD controller along with an optimal trajectory planning approach as an alternative and baseline for comparison.

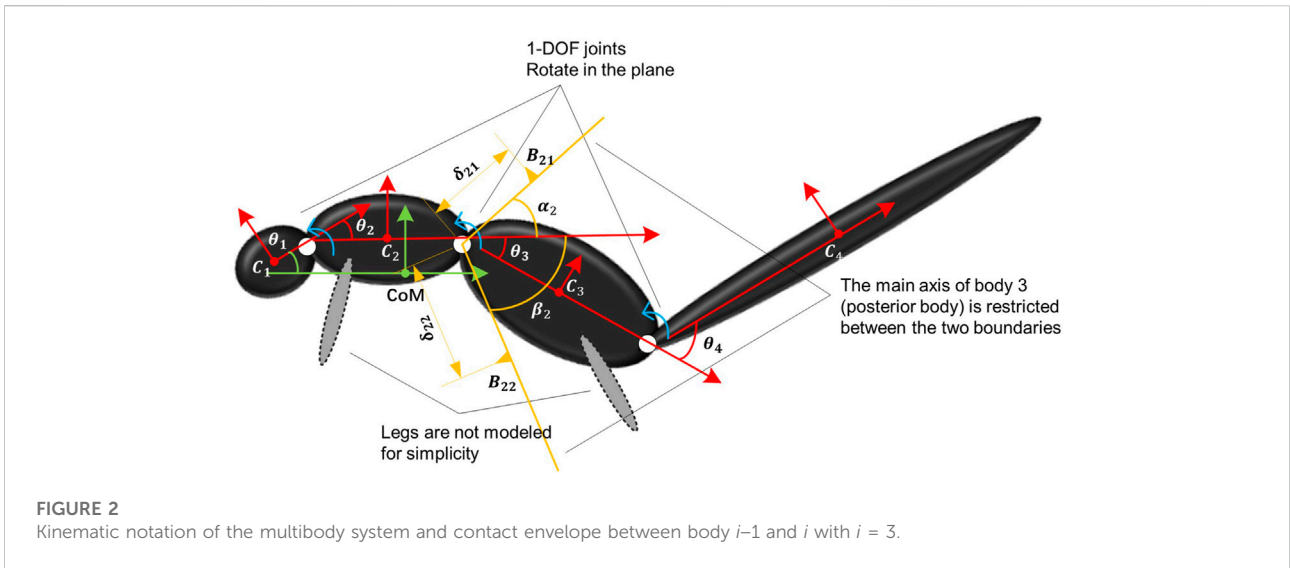
We believe that aerodynamics also plays a role in a real squirrel's flight behavior because its furry tail can cause enough air drag (external force) to change its body momentum. However, in this study we are investigating the dynamics and control only in the case of conservation of momentum without external force. We know that animals can achieve their desired motion behavior by only controlling the relative motions of their body segments and limbs without air drag on tails. For example, even a tailless cat can land on its four feet after it is dropped from an upside-down configuration (Fredrickson, 1989). We leave the investigation of the complex aerodynamic modeling and analysis of the furry tails to future research.

The rest of this article is organized as follows: In Section 2, the dynamics formulation of the model is introduced. Section 3 describes the RL-based method and PD controller. The simulation environment and results are in Section 4. Finally, the article is concluded with a discussion in Section 5.

2 Dynamics of the 2D multilink model

2.1 Dynamics

In general, we can model the squirrel with N rigid bodies connected with each other. O_b , O_C , and O_W are the origins of the coordinate frames attached to the i th body, the center of mass (CoM) of the multibody system, and the inertial frame, respectively. These symbols are also used to represent their



associated coordinate frames. Let θ_1 be the angle between the major axis (defined as the y axis) of the first body and the y axis of frame O_C . Then, θ_i ($i = 2, \dots, N$) is defined as the angle between the major axis of the i th and $i - 1$ th body. The configuration of the system can be determined by $\theta = [\theta_1, \theta_2, \dots, \theta_N]^T \in \mathbb{R}^N$. A multibody rigid model of a squirrel with $N = 4$ is shown in Figure 2. Next, we can derive the dynamics of the multibody system (Featherstone, 1984) in frame O_C , following Lagrange's equations of the second kind. The dynamic equation of the system is

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} = [0, \tau + \tau_c]^T, \quad (1)$$

where $M(\theta)$ is the positive-definite inertia matrix, which is determined by the configuration θ , $C(\theta, \dot{\theta})$ is the Coriolis matrix, which is determined by the configuration θ and its derivative $\dot{\theta}$. We omit the gravity force term in the left-hand side since Eq. 1 is derived in the frame O_C . In fact, gravity only has influence on the position, velocity of CoM, and thus the duration of the flight, but when we focus on the configuration θ only, the gravity does not have influence. In simulation, we can calculate θ and the position of CoM, respectively and separately, as shown in Section 4.1.

Since we could only control the extra inner wrench on each joint, the first element of the right-hand side of Eq. 1 should be zero and the rest of the elements $\tau \in \mathbb{R}^{N-1}$ are the torques applied to the $N - 1$ joints, i.e., the joint control torques. τ_c is the vector of the generalized torques caused by collision between body segments and it is non-zero only if such a collision occurs. Since we assume that body segment collisions are intermittent, in our simulation we calculate the velocities right before and after a collision rather than the specific τ_c , which we discuss in detail in Section 2.2. Considering that the CoM of the

system moves only under the action of gravity, the motion of the system can also be determined in the frame O_W .

The configuration $\theta(t)$ is determined by three different factors: 1) the joint control torques; 2) the coupling effect caused by the joint motion; and 3) the initial motion conditions. The second term is the main source of non-linearity. The third term determines the total linear and angular momentums of the system, which cannot be changed during the entire flight phase. In this study, we set $\dot{\theta}_1(0) = \pi/2$ rad/s and $\dot{\theta}_i(0) = 0$ for $i = 2, \dots, N$ (meaning that the squirrel rotates at $\pi/2$ rad/s at the beginning with respect to where it stands/sits). In 2D simulation, $\dot{\theta}_1(0)$ cannot be too large, otherwise, factor 3 mentioned previously determines the whole flight and make it very difficult to control. In this study, we particularly focus on joint control of the system to achieve the goal that the head (body 1) must be quickly turned to facing the landing spot after ejection and then lock toward the landing spot throughout the rest of the projectile flight.

2.2 Intermittent contact between the bodies

When a squirrel (or another animal) is unintentionally thrown into air leading to tumbling in flight, some of its body segments will intermittently contact with each other causing impact and bouncing of the contacting bodies during the flight. This physical phenomenon should be captured in the modeling for realistic simulation. To simulate this body segment contact behavior, we add a contact modeling constraint. In general, we consider the contact model between bodies $i - 1$ and i and show an example with $i = 3$ in Figure 2. The orange lines are the two

physical boundaries of body $i - 1$ with angles α_i and β_i , respectively. In this way, body i is restricted between the two boundaries, i.e., $\theta_i \in [-\beta_i, \alpha_i]$, $i = 2, \dots, N$.

To calculate the new motion state immediately after contact, we assume two contact points B_{i1} and B_{i2} on the two boundaries to determine the contact location. Let $|B_{i1}| = \delta_{i1}$, $|B_{i2}| = \delta_{i2}$, and $\theta_i = \alpha_i$ when the impact occurs. The relative angular velocity of body i relative to $i - 1$ is $\dot{\theta}_i$ and the relative velocity of B_{i1} on body i relative to $i - 1$ right before impact is

$$\Delta \mathbf{v}_{B_{i1}} = [-\dot{\theta}_i \delta_{i1} \sin(\theta_i), \dot{\theta}_i \delta_{i1} \cos(\theta_i)]^T. \quad (2)$$

Then, the projected relative velocity of B_{i1} on body i relative to $i - 1$ in the impact direction is

$$\Delta \mathbf{v}_{B_{i1}}^p = \mathbf{R} \left(\sum_{j=1}^i \theta_j \right) \Delta \mathbf{v}_{B_{i1}} = [0, \dot{\theta}_i \delta_{i1}]^T, \quad (3)$$

where $\mathbf{R}(\cdot)$ represents the 2D rotation matrix. Right after the impact, the relative angular velocity changes to $\dot{\theta}'_i$, and the separation velocity at B_{i1} is

$$\Delta \mathbf{v}_{B_{i1}}^{p'} = [0, \dot{\theta}'_i \delta_{i1}]^T. \quad (4)$$

Assuming the coefficient of restitution of the impact is k ,

$$-k \dot{\theta}_i \delta_{i1} = \dot{\theta}'_i \delta_{i1}. \quad (5)$$

It is easy to figure out that δ_{i1} (i.e., the position of each contact point) has no influence on the angular velocity after impact, i.e.,

$$-k \dot{\theta}_i = \dot{\theta}'_i. \quad (6)$$

In addition to Eq. 6, the conservation law of angular momentum in the frame O_C also needs to be satisfied. In frame O_C , the angular momentum of the system is

$$H = \sum_{i=1}^N J_i \dot{\theta}_i + y_{ci} m_i \dot{z}_{ci} - z_{ci} m_i \dot{y}_{ci}, \quad (7)$$

where m_i , J_i , y_{ci} , and z_{ci} are the mass, momentum of inertia, and y - and z -axis coordinates of the CoM of the i th body, respectively. Since the coordinates of each CoM are determined by the configuration θ , H can be rewritten as a function of θ and $\dot{\theta}$ and the function is linear with respect to $\dot{\theta}$, namely, $H = \tilde{\mathbf{J}}(\theta)^T \dot{\theta}$, where $\tilde{\mathbf{J}}$ can be regarded as an “equivalent” inertia tensor corresponding to $\dot{\theta}$. The conservation law of angular momentum on the impact between bodies $i - 1$ and i is

$$\sum_{j=i-1}^{i+1} \tilde{J}_j \sum_{k=i-1}^j \dot{\theta}_k = \sum_{j=i-1}^{i+1} \tilde{J}_j \sum_{k=i-1}^j \dot{\theta}'_k. \quad (8)$$

Additionally, the relative angular velocity of body $i + 1$ with respect to that of body i should be the same right before and after the impact, namely,

$$\sum_{j=i-1}^{i+1} \dot{\theta}_j = \sum_{j=i-1}^{i+1} \dot{\theta}'_j. \quad (9)$$

To summarize, Eqs 6, 8, 9 form the equations relating the angular velocities right before and after the impact. If more than two bodies simultaneously collide, we can use similar equations to calculate. For example, if bodies $i - 1$, i , and $i + 1$ collide at the same time, then the equations

$$\begin{aligned} \sum_{j=i-1}^{i+2} \tilde{J}_j \sum_{k=i-1}^j \dot{\theta}_k &= \sum_{j=i-1}^{i+2} \tilde{J}_j \sum_{k=i-1}^j \dot{\theta}'_k \\ \sum_{j=i-1}^{i+2} \dot{\theta}_j &= \sum_{j=i-1}^{i+2} \dot{\theta}'_j \\ -k \dot{\theta}_i &= \dot{\theta}'_i \\ -k \dot{\theta}_{i+1} &= \dot{\theta}'_{i+1} \end{aligned} \quad (10)$$

can be used to determine the motion state right after the impact.

3 Control method

In this section, we develop the control methodology of locking the first body of the multibody system (i.e., the head of the squirrel) toward the landing point during the flight phase by applying joint control efforts. We apply two different control methods to achieve the control goal: 1) trajectory planning in addition to a PD feedback control law and 2) an RL-based control policy. The first method is a traditional control approach while the second is a machine learning-based approach.

3.1 Trajectory planning for a PD controller

A PD controller needs a pre-planned motion trajectory as a reference to determine the position and velocity errors as the input to the PD control law. Therefore, we need to calculate a reference trajectory that represents the squirrel’s motion behavior. It is clear that the CoM of the squirrel follows the projectile motion trajectory determined by the gravitation and squirrel’s initial conditions. However, the relative motion of the individual bodies or joints of the squirrel will be governed by the multibody dynamics and conservation of angular momentum. As shown in Section 2, the dynamics of the system is non-linear, which makes it difficult to derive an analytic solution of the dynamic system. Hence, we choose to solve a set of optimization problems for the trajectory planning. Assume D is the fixed landing point on the ground, R is the moving looking point, H is the head point, H_p is on the ground, and HH_p is parallel to the z axis of the frame O_W , as shown in Figure 3. To ensure the head (Body 1) to look toward the landing point D at $(y_b, 0)$ in frame O_W , θ_1 should be equal to γ in Figure 3, which is the angle between HD (the line between the head point and landing point)

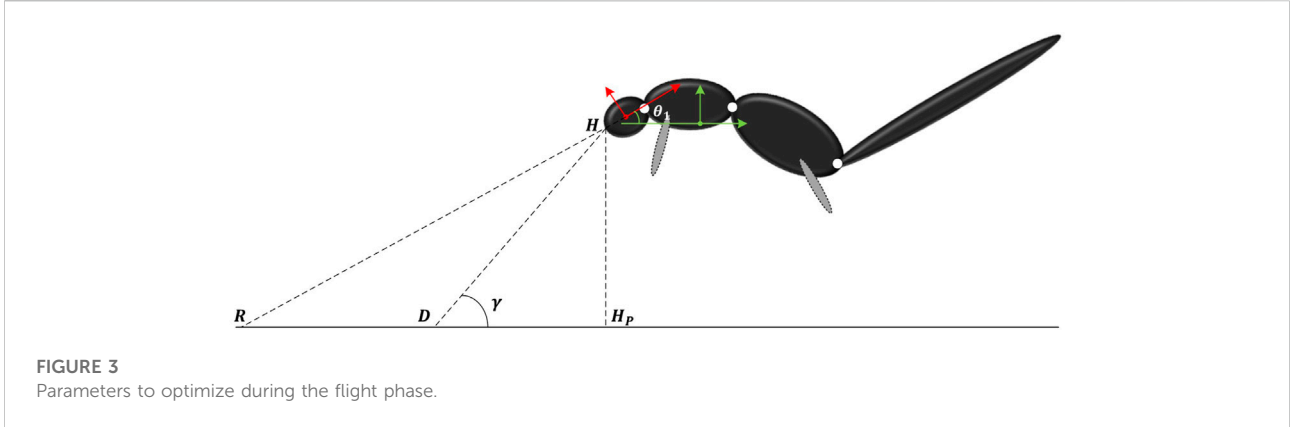


FIGURE 3
Parameters to optimize during the flight phase.

and the y axis of the frame O_W . Then, the optimization objective is

$$\min_{\theta} \frac{1}{2} (\tan \gamma - \tan \theta_1)^2. \quad (11)$$

To avoid singularity when $\theta_1 = \pi/2$ and $|DH_p| = 0$, we use the following optimization objective instead:

$$\min_{\theta} \frac{1}{2} (|HH_p| \cos \theta_1 - |DH_p| \sin \theta_1)^2. \quad (12)$$

During the flight, the total angular momentum of the system in the frame O_C in Eq. 7 should be constant. Assuming that the initial angular momentum is H_0 and the configuration of the system at time t is $\theta(t)$, then from time s to time $s + \delta s$, where δs is a short time period

$$\int_s^{s+\delta s} \tilde{\mathbf{J}}(\theta(t))^T \dot{\theta}(t) dt = H_0 \delta s. \quad (13)$$

If $\tilde{\mathbf{J}}(\theta(t))$ remains almost constant, i.e., $\tilde{\mathbf{J}}(\theta(s))$, during δs , then

$$\tilde{\mathbf{J}}(\theta(s))^T (\theta(s + \delta s) - \theta(s)) \approx H_0 \delta s. \quad (14)$$

However, during simulation, we find that if we set D as the landing point from the beginning, the configuration solution of the system at time $t = 0$ will be quite different from the initial configuration, and Eq. 14 is no longer satisfied. To solve this problem, we introduce a moving point D_f on the ground as a pseudo landing point at $(y_{pl}(t), 0)$, whose y position is defined as

$$y_{pl}(t) = \max(y_l, y_{ml}(t)), \quad (15)$$

where

$$y_{ml}(t) = y_{R0} - \frac{g(y_{R0} - y_l)}{0.2 \times (\dot{y}_{CoM} + \sqrt{\dot{y}_{CoM}^2 + 2gz_{CoM}})} t, \quad (16)$$

where y_{R0} is the initial y coordinate of R in Figure 3 and $g = 9.81 \text{ m/s}^2$ is the gravity acceleration. In this way, D_f moves from the initial position of R to the target point D in the process, and thus, we have Algorithm 1 for calculating the reference trajectory.

Given θ_{last} , $\tilde{\mathbf{J}}$, (y_{CoM}, z_{CoM}) , $(\dot{y}_{CoM}, \dot{z}_{CoM})$, y_l , Δt , H_0
 Initialize $y_{pl} = y_{R0}$, $y_{ml} = y_{R0}$
 While not landing:
 Update $y_{CoM} \leftarrow y_{CoM} + \dot{y}_{CoM} \Delta t$
 Update $z_{CoM} \leftarrow z_{CoM} + \dot{z}_{CoM} \Delta t - g \Delta t^2 / 2$
 Update $\dot{z}_{CoM} \leftarrow \dot{z}_{CoM} - g \Delta t$
 Update $y_{ml} \leftarrow y_{ml} - \frac{g \Delta t (y_{R0} - y_l)}{0.2 \times (\dot{y}_{CoM} + \sqrt{\dot{y}_{CoM}^2 + 2gz_{CoM}})}$
 Update $y_{pl} \leftarrow \max(y_l, y_{ml})$
 Solve the optimization problem
 $\min_{\theta} (|HH_p| \cos \theta_1 - |DH_p| \sin \theta_1)^2 / 2$
 s.t. $\tilde{\mathbf{J}}^T \theta = \tilde{\mathbf{J}}^T \theta_{last} + H_0 \Delta t$, $-\beta_i \leq \theta_i \leq \alpha_i$, $i = 2, \dots, N$
 Update $\theta_{last} \leftarrow \theta$ and $\tilde{\mathbf{J}}$
 Return $\theta(t)$

Algorithm 1. Calculation of the reference trajectory.

Suppose a reference trajectory $\theta_r(t)$ is obtained from Algorithm 1, then we propose a PD controller to allow the system to track the reference trajectory in the way, as shown in Figure 4. Suppose $\tilde{\theta} = [\theta_2, \dots, \theta_N]^T$ and $\tilde{\theta}_r = [\theta_{2r}, \dots, \theta_{Nr}]^T$, the tracking error is calculated as

$$\epsilon = 2(\tilde{\theta}_r - \tilde{\theta}) / \pi. \quad (17)$$

We choose the control gains of the PD controller on each joint separately. Let joint i be the joint between body $i - 1$ and i and K_{pi} , K_{di} be the corresponding control gains, then the output of the PD control law is defined as

$$\tau = \mathbf{K}_p \epsilon + \mathbf{K}_d \frac{d\epsilon}{dt} \approx \mathbf{K}_p \epsilon + \mathbf{K}_d \frac{\epsilon - \epsilon_{last}}{\Delta t}, \quad (18)$$

where $\mathbf{K}_p = \text{diag}(K_{p2}, K_{p3}, \dots, K_{pN})$ and $\mathbf{K}_d = \text{diag}(K_{d2}, K_{d3}, \dots, K_{dN})$. Physically, τ is a vector whose components are the required joint control torques.

3.2 Reinforcement learning-based method

Conventional control methods, as the one described in Section 3.1, usually require trajectory planning, which is a difficult problem if one does not fully understand how a

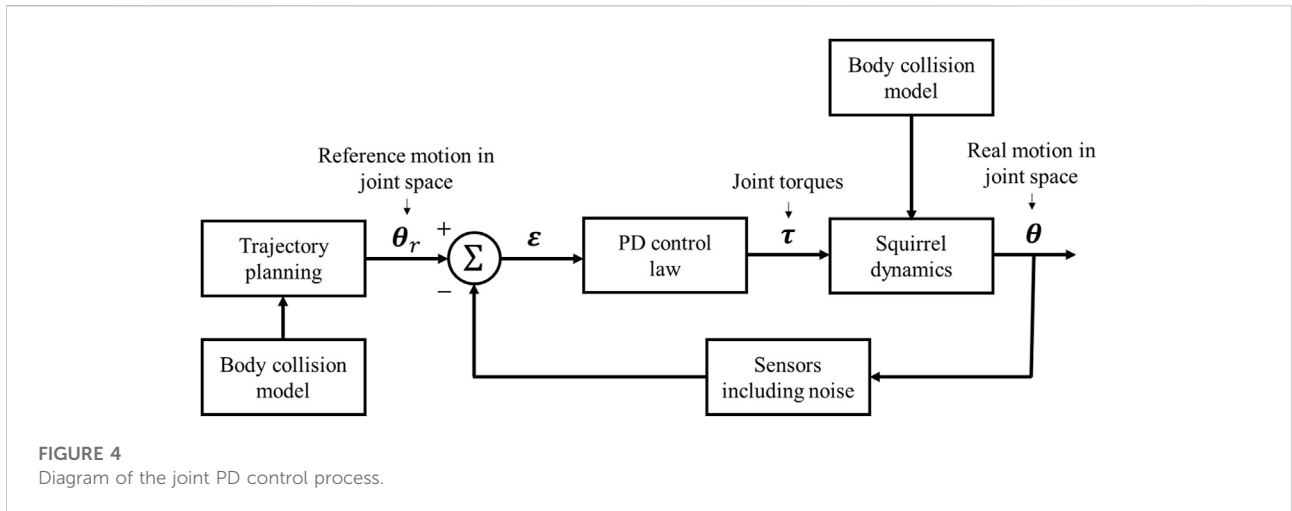


FIGURE 4
Diagram of the joint PD control process.

squirrel controls its body segments (i.e., joint motions) for a target motion behavior. In fact, trajectory planning can be avoided by applying reinforcement learning technology in the control solution for the squirrel to achieve an expected behavior. In this section, we apply the proximal policy optimization (PPO) algorithm (Schulman et al., 2017) as the RL method to train a control policy for the squirrel to achieve the desired behavior.

PPO has an actor-critic architecture and works for both discrete and continuous action domains. It trains a policy to obtain the maximum expectation of total reward which we set before a training. Therefore, if we set the reward properly, the policy trained by PPO will implement specific functions. To access the desired policy π_{Θ} using the PPO2 algorithm (Schulman et al., 2017), the optimization objective \mathcal{L} to maximize is defined as

$$\mathcal{L}^{\Theta'}(\theta) = E_{(s_t, a_t) \sim \pi_{\Theta'}} \min \left[\frac{p_{\Theta}(a_t|s_t)}{p_{\Theta'}(a_t|s_t)} A^{\Theta'}(s_t, a_t), \text{clip} \left(\frac{p_{\Theta}(a_t|s_t)}{p_{\Theta'}(a_t|s_t)}, 1 - \epsilon, 1 + \epsilon \right) A^{\Theta'}(s_t, a_t) \right], \quad (19)$$

where (s_t, a_t) represent the state and action of the agent at time = t ; $\pi_{\Theta'}$ represents another policy with parameter Θ' (usually similar to π_{Θ} by applying minimization and clip function); $p_{\Theta}/p_{\Theta'}$ represents corresponding probability; and $A^{\Theta'}(s_t, a_t)$ represents the advantage function under state s_t and action a_t of policy π_{Θ} . Usually, for an RL trajectory τ with length T (defined as a series of state-action pair $(s_0, a_1, s_1, \dots, a_T, s_T)$), we set

$$A^{\Theta}(s_t, a_t) = \sum_{t'=t}^T \gamma^{t'-t} r_{t'} - E[R(\tau)], \quad (20)$$

where $r_{t'}$ denotes the one-step reward at t' in an RL trajectory τ ; $\gamma \in [0, 1)$ denotes the discount factor; and $E[R(\tau)]$

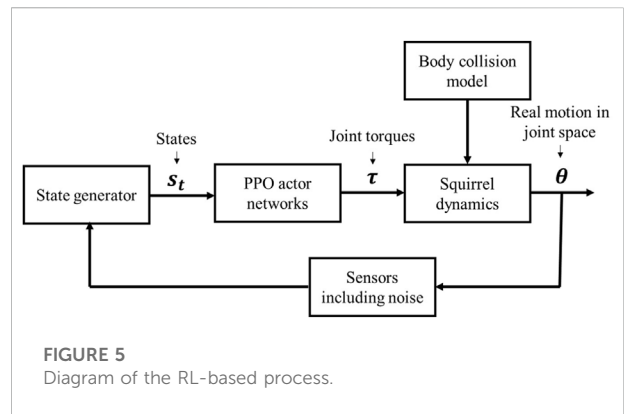
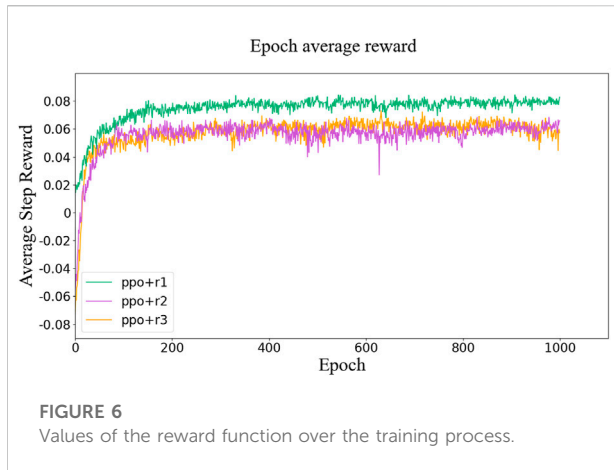


FIGURE 5
Diagram of the RL-based process.

(approximated by a neural network) is the expectation of trajectory reward $R(\tau)$. The control method is based on the PPO2 algorithm, as shown in Figure 5. The vector s_t describes the current status of the system and it consists of the head coordinate in the frame O_W , θ , $\dot{\theta}$, and the velocity of the CoM in the frame O_W . The vector τ is the control torque applied on each joint and is sampled from a trainable normal distribution. For one-step reward, since a desired policy should 1) minimize the distance between point R and the landing point (or the pseudo landing point); 2) avoid high angular velocities; and 3) avoid high joint torques. The first objective is to lock the head (body 1) toward the landing point, while the other two are to reduce the kinetic energy. Thus, there are three different settings.

$$\begin{aligned} r_1 &= 0.1 \exp(-|y_R - y_l|) + 0.01 \exp(-0.1 \|\dot{\theta}\|) + 0.001 \exp(-\|\tau\|) \\ r_2 &= 0.1 \exp(-|y_R - y_l|) - 0.001 \|\dot{\theta}\| - 0.001 \|\tau\| \\ r_3 &= 0.1 \exp(-|y_R - y_{pl}|) - 0.001 \|\dot{\theta}\| - 0.001 \|\tau\| \end{aligned} \quad (21)$$



These three reward settings aim at minimizing the distance between point D (or D_f) and R , the l_2 -norm of $\dot{\theta}$ and τ , respectively. The difference among the reward settings also focuses on a smoother configuration trajectory from two aspects: 1) training with D_f instead of D and 2) different ways of penalty on $\dot{\theta}$ and τ .

We train control policies with the three rewards for 1,000 epochs with 2,048 timesteps in each epoch. The values of the reward function over the training process are shown in Figure 6, where the rewards all tended to saturate near end and little gain can be obtained from further training.

4 Simulation results and discussion

4.1 Simulation environment

The simulation environment was established on the Ubuntu 20.04 system with a 16 GB RAM and an Intel Core i9-9900kf CPU. We simulated the motion behavior by calculating and recording the motion for each time step using Python. The process is described in Algorithm 2. It used the Newmark-beta method (Newmark, 1959), as the method of numerical integration to solve Eq. 1 with time step $\Delta t = 0.001$ s.

```

Initialize model configuration  $\theta, \dot{\theta}, (y_{CoM}, z_{CoM})$ , and  $(\dot{y}_{CoM}, \dot{z}_{CoM})$ 
While not landing:
  Get joint control torques  $\tau$ 
  Calculate  $\ddot{\theta}$  using Eq.(1)
  Update  $\theta \leftarrow \theta + \dot{\theta}\Delta t + \ddot{\theta}\Delta t^2/2$ 
  Update  $\dot{\theta} \leftarrow \dot{\theta} + \ddot{\theta}\Delta t$ 
  If body impact:
    Update  $\theta$  and  $\dot{\theta}$  using Eq.(8) and (9)
  Update  $y_{CoM} \leftarrow y_{CoM} + \dot{y}_{CoM}\Delta t$ 
  Update  $z_{CoM} \leftarrow z_{CoM} + \dot{z}_{CoM}\Delta t - g\Delta t^2/2$ 
  Update  $\dot{z}_{CoM} \leftarrow \dot{z}_{CoM} - g\Delta t$ 
Return  $\theta(t), (y_{CoM}, z_{CoM})(t)$ 

```

Algorithm 2. The process of the simulation.

In the simulation, we set $N = 4$ and the four bodies are the head, upper body (chest portion of the body trunk), lower body (abdomen portion of the body trunk), and tail. The model parameters and the selected PD control gains are shown in Table 1. The platform ejecting the squirrel has an initial angular velocity of 4π rad/s, and thus the velocity of the CoM is determined by where the squirrel is initially located on the platform. However, once the squirrel detects the sudden movement of the platform, it will quickly move its body to adjust its body posture and thus its initial angular velocity can be different from that of the platform. In the simulation, we set the joint motion for the squirrel model as $\pi/2$ rad/s, as shown in Table 1. The reference trajectory generated based on Algorithm 1 is shown in Figure 7.

In the following results, the simulation has been repeated 100 times with or without random noise in the feedback information. The noise is defined as a Gaussian distribution of 0 mean and $0.05A$ standard deviation, where A is the maximal measuring range of an input variable (e.g., $\theta_4 \in [-0.75\pi, 0.75\pi]$ then $A = 0.75\pi$). Therefore, both the nominal motion and noised motion (with the standard deviation of the error distribution) of the squirrel are shown in the following sections.

4.2 Control of the configuration

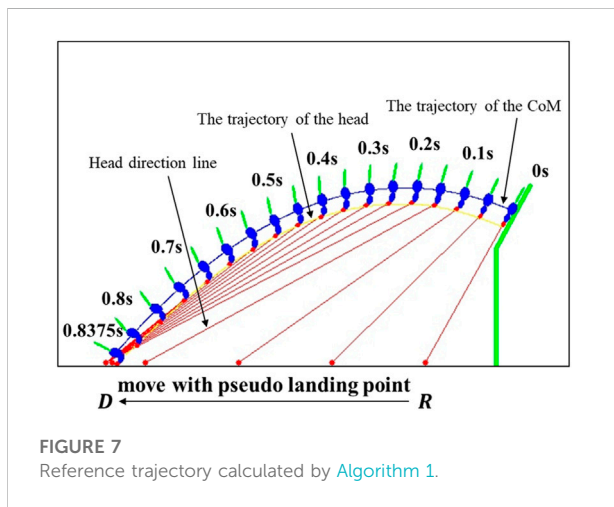
In this section, we discuss the simulation results obtained from different controlling methods. First, θ_1 is a key parameter of the behavior since it is the squirrel's head direction and also directly connected with the distance of DR in the control algorithm.

Simulated squirrel's head orientation (θ_1 values) using different control methods are shown in Figure 8. The reference value of θ_1 (black line) has the following features. The head motion θ_1 can be divided into three parts: 1) quickly turns the head to the orientation to be able to see the landing point (0 s—0.2 s in Figure 7). In this part, the squirrel adjusts its configuration from its initial configuration to the one such that it faces and can observe the landing point. 2) A slow increase in time (0.2 s—0.7 s in Figure 7), and this is the main part of the flight in which the squirrel locks its head toward the landing spot to maintain its visibility. The slow rotation of the head is due to its motion along the projectile trajectory. 3) Rapid decrease in time near landing (0.7 s—0.8375 s in Figure 7). In this part, the squirrel needs to quickly adjust its head orientation near the landing point to maintain its head locking toward the landing point near the end of the projectile trajectory.

The PD controller provides almost the same motion trajectory of θ_1 as the pre-planned reference, but θ_1 from PPO with r_1 and r_2 has a shorter part 1 (namely, reaching the locking posture much faster) and a longer part 2, that is because at part 2, the distance of DR is close to zero, and thus the agent gets a higher reward. However, a shorter part 1 may result in a more

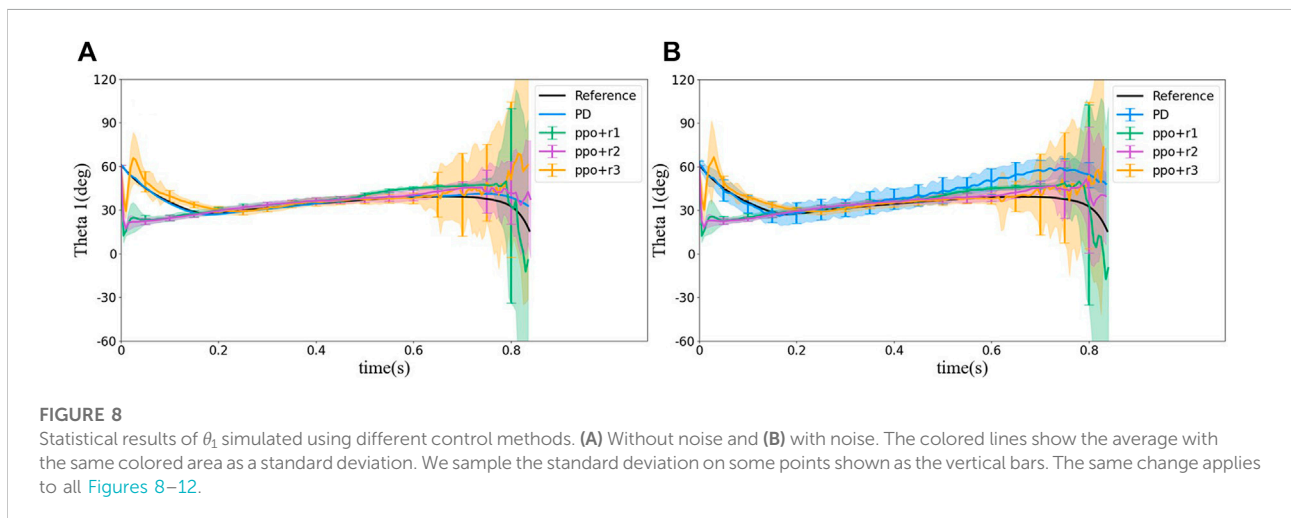
TABLE 1 Parameters of the model, kinematics, and PD controller.

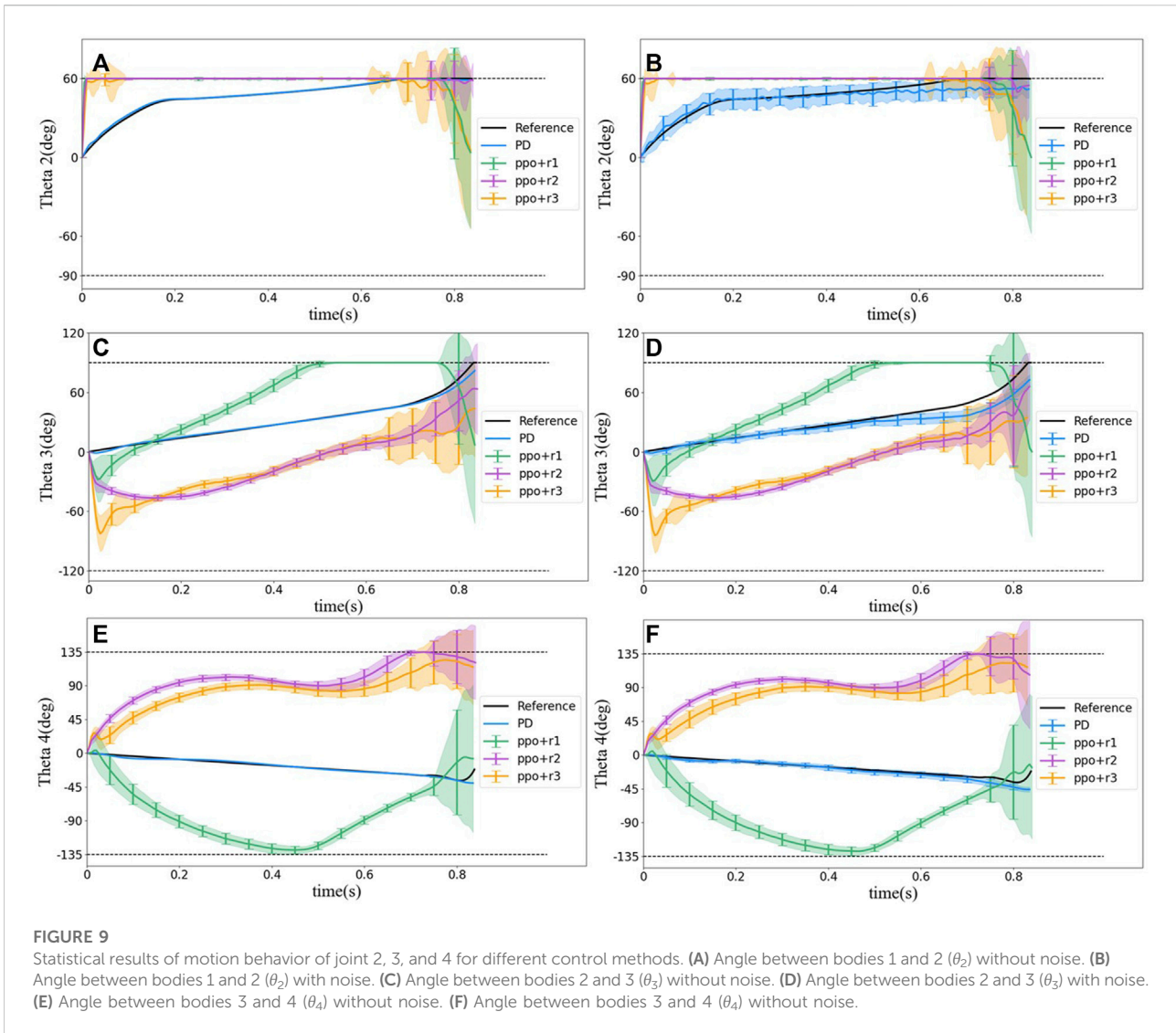
Data item	Value	Data item	Value	Data item	Value	Data item	Value
m_1	0.05 kg	l_4	0.2 m	k	0	$K_p(\text{Joint 1})$	0.66
m_2	0.35 kg	b_1	0.05 m	α_2	60°	$K_p(\text{Joint 2})$	0.55
m_3	0.4 kg	b_2	0.07 m	α_3	90°	$K_p(\text{Joint 3})$	0.19
m_4	0.2 kg	b_3	0.1 m	α_4	135°	$K_d(\text{Joint 1})$	0.009
l_1	0.06 m	b_4	0.02 m	β_2	90°	$K_d(\text{Joint 2})$	0.011
l_2	0.1 m	$\theta(0)$	[61°, 0°, 0°, 0°]	β_3	120°	$K_d(\text{Joint 3})$	0.009
l_3	0.14 m	$\dot{\theta}(0)$	[90,0,0,0] deg/s	β_4	135°		



rapid change of the configuration, for example, a higher angular acceleration in each joint, which may exceed the system’s power limit. The reward r_3 is set to extend part 1, but as shown in Figure 8, θ_1 does not follow the reference at the beginning,

instead, it decreases quickly as in PPO with reward r_1 and r_2 and increases to target θ_1 soon. Therefore, r_3 did not limit the sudden decrease at the beginning. On the contrary, extra motion is introduced. It should be emphasized that the PD controller tracked the reference trajectory better because it is designed to track the reference, while the RL control does not have any knowledge of the reference. From overall behavior perspective, both controllers achieved the control goal, namely, to quickly turn the head (eyes) toward the landing point after ejected into air and then remained looking at the landing point during the flight until landing. The motion histories of the other joints during the flight are shown in Figure 9. The figure demonstrates that all the joint motions from RL methods are very different from each other and from those of the reference trajectory. For example, in the reference trajectory, the angle between bodies 1 and 2 (i.e., θ_2) move slowly to 60° (i.e., α_1) at about 0.6 s, whereas in RL methods, θ_2 reaches 60° quickly and maintain at the angle for all the time. However, θ_1 values from all methods including the reference trajectory are the same. This phenomenon emphasizes that the reference trajectory may not be a naturally optimal solution although it was calculated from an optimization problem.





4.3 Control of the distance of DR

The distance between D and R is an important and straightforward index to evaluate a control strategy of locking the head toward the landing point. Considering that the distance values of DR ranges from 0 to infinity, any distance values over 10 m were recorded as 10 m in our simulation. The value of the DR distance for different control strategies is shown in Figure 11. We divide the flight period into three phases as defined in Section 4.2. The first phase lasts about 0.2 s, and the DR distance decreases quickly to near zero, while the second phase lasts about 0.5 s, and the main purpose of the strategy transforms to maintaining the DR distance near zero. The third phase lasts about 0.13 s, and the main purpose is the same as the second phase but requiring a more rapid rotation of head. In Figure 10,

we use red, green, and yellow background to distinguish the three phases.

Figures 10A, B demonstrate some results about the accuracy of the different strategies. First, the PD controller performs better than the RL method under ideal situation (i.e., without noise) but becomes difficult after introducing noise. When random noise is applied as in the reality, the convergence speed of the DR distance using the PD controller obviously decreases, while the standard deviation increases, suggesting that the pose at the same time in different ejections varies from each other due to the random noise. However, if the standard deviation of the Gaussian distribution of the noise decreases from $0.05A$ to $0.02A$, where A is the maximal measuring range of an input variable, the PD controller could still maintain the performance better than the RL method, as shown in Figure 10C, suggesting that the PD

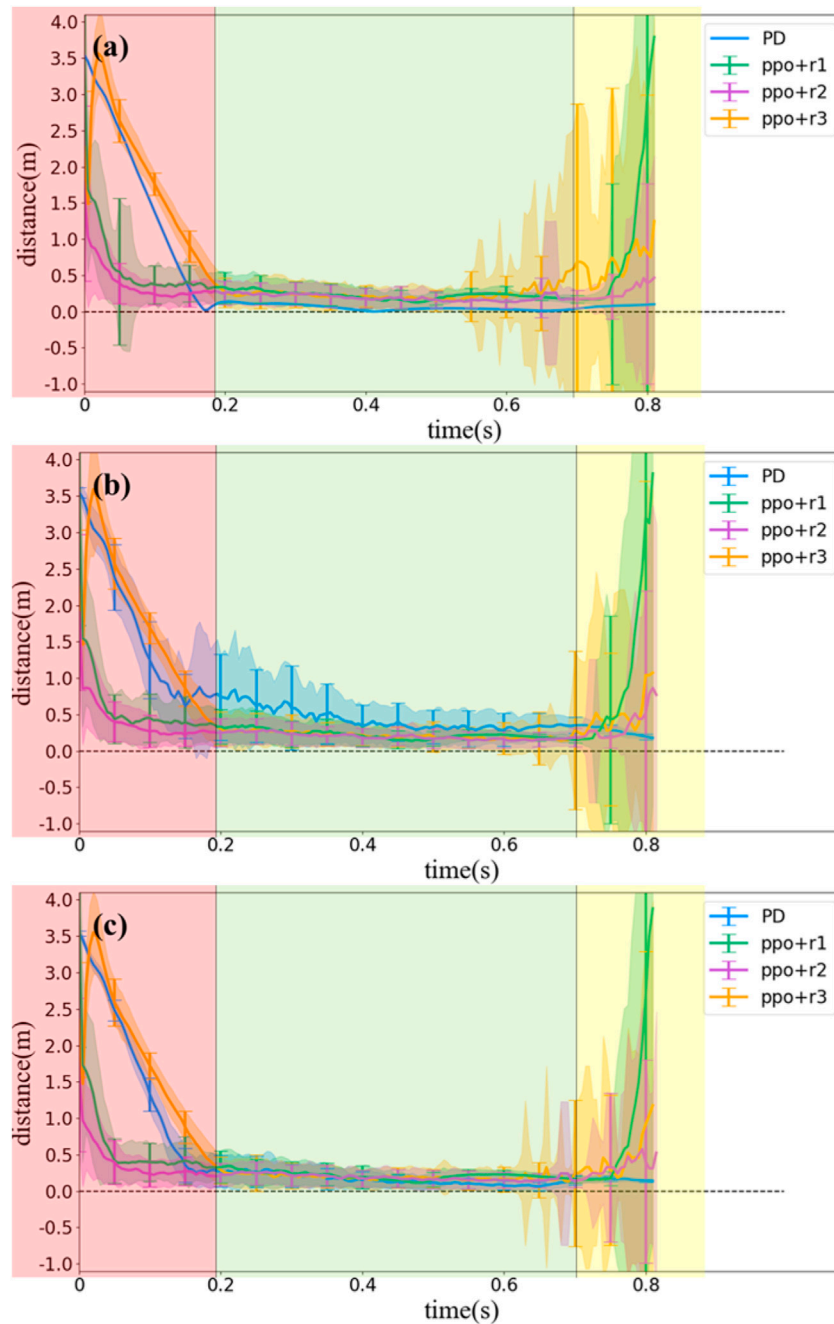


FIGURE 10
 Statistical results of *DR* distance values simulated with different control methods. **(A)** Without noise, **(B)** with noise (0.05A), and **(C)** with noise (0.02A) (A is the maximal measuring range of an input variable).

controller also has certain robustness to noise, but not as robust as the RL method.

Second, all the three RL methods are not so sensitive to the noise as the PD controller, but their variation (with respect to random noise) is more obvious near the landing time. We

infer the reason would be that, as the system moves near the landing time, the motion is in the part 3 region, as we have discussed in Section 4.2. However, as the vast majority of the flight is in part 2, the RL method may have learned the strategy that weighs part 2 performance much more than the part

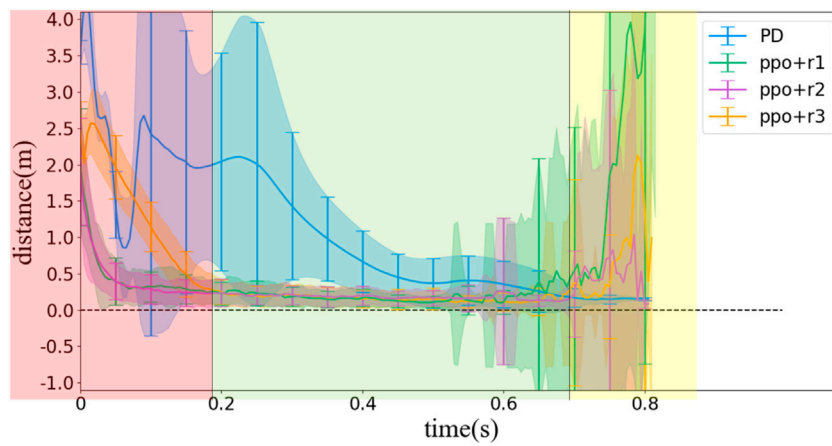


FIGURE 11
Statistical results of DR distance values simulated with different control methods and random initial poses (without noise).

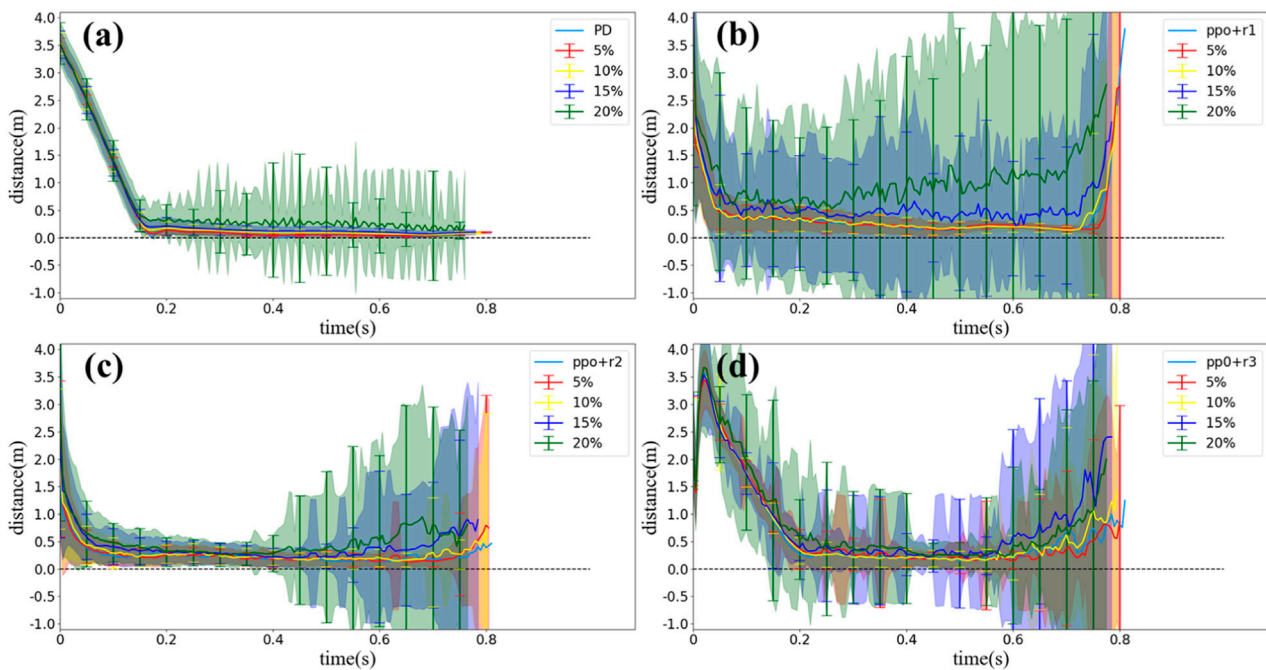


FIGURE 12
Statistical results of DR distance values simulated with different control methods and model errors of 5%–20% (without noise). (A) PD controller, (B) PPO method with r_1 , (C) PPO method with r_2 , and (D) PPO method with r_3 .

3 performance and is more sensitive when the system changes more frequently.

Third, among the three one-step reward designs, the PPO method with r_2 is the most robust one. There are two aspects for the reason. The first is that the l_2 -norm provides more penalty

than the exponent of the l_2 -norm when $\dot{\theta}$ and τ become too large. The second is that as the set of D_f in the reward cannot smooth the motion, as shown in Figures 7, 8, the direct reward of the DR distance can lead to a longer part 2, which may contribute to the less joint motion variations.

4.4 Control performance with random initial poses

For unexpectedly ejected squirrels, the initial pose is unpredictable, thus the motion should be random in joint space. In this section, we also test the performance of different methods with random initial poses which satisfy: 1) $\theta_1 = 61^\circ$; 2) $-\beta_i \leq \theta_i \leq \alpha_i$ for $i = 2, 3, 4$; and 3) all bodies should stay above the platform. Since the joint initial motions are random, we only record the *DR* distance to evaluate the performances. Figure 11 illustrates the results. Compared with Figure 10A, we find that the PD controller cannot deal with the random initial conditions well, and that is because the ejection is unexpected and it is impossible to calculate a reference trajectory for each random initial pose, thus we can only use the same reference trajectory, which influences the calculation of error ε for the PD controller. On the other hand, the PPO method keeps almost the same performance with random initial poses, which suggests the robustness of the reinforcement learning-based control strategy.

4.5 Control performance with slightly changed models

In reality, the physical properties of a squirrel (i.e., mass and size parameters) are changing over time, but the squirrel can always lock its head/eyes toward the landing point while body tumbling in air. Therefore, it is also important to test the robustness of the strategy provided by the PD controller/RL method. In this section, we randomly changed the parameters of our multilink model, i.e., mass/length/width of each body, by $\pm 5\%/10\%/15\%/20\%$ without tuning the PD control law or re-training the RL control policy and then observed the corresponding performance of the controllers.

The results are shown in Figure 12A. For the PD controller, Figures 12A illustrates that the PD controller is robust to model errors by 15%. When the model errors reach 20%, the standard deviation of *DR* distance grows significantly, which means that the motion varies in each simulation. For the RL method, similar results are shown in Figures 12B–D; 10% of model errors does not influence the *DR* distance, but 15% and 20% of model errors can cause uncertainty in different stage of motions. However, in the sense of average, both the PD controller and RL method are robust to the model error even by 20% (except for the PPO method with r_1 , but it can still maintain the average under 15% errors).

5 Conclusion and future work

In this work, we studied the dynamics and control of a squirrel's amazing capability of maintaining its head facing

the landing spot, while its other body segments tumbling in air after it is unexpectedly ejected into air from any initial pose. To understand the dynamics and explain the observed real squirrel's behavior, we developed a simplified 2D multibody dynamics model with body segment collision constraints of a squirrel and applied two very different control methods to reproduce the observed squirrel behavior. The first control method is to plan a reference motion trajectory first, representing the squirrel motion behavior and then apply a PD feedback controller to track the planned reference trajectory. The second control method is to use a reinforcement learning process to train a deep neuron-network-based control policy to achieve the squirrel motion behavior. In both control methods, random noise (white noise) is added to the sensed feedback motion data to make the simulated situation closer to the reality. Simulation results demonstrated that both methods successfully achieved the expected control goal of quickly turning the head toward the landing point and then locking the head toward the landing spot during the flight phase. Comparing the two control methods, the RL method performs better in terms of closer to expected behavior and robustness against sensor errors. However, the RL method shows more variant joint motions with respect to noisy input data near the landing time, but these variations are all acceptable because they all achieved the targeted head motion behavior. Another main advantage of the RL method is that we do not need to plan a reference trajectory first, and thus the method would suit more to the natural environment and lead to more natural outcome.

The future work especially focuses on the following aspects: 1) further development of the one-step reward setting for the RL process. Current reward settings still converge to local optimal and cannot stretch the body enough. 2) Expand the dynamics model to the 3D space. To achieve this expansion, the segmentation of the motion and multi-layer control are necessary. Another future research direction of 3D motion is gait analysis, which is to reveal repetitive motion pattern of body segments including legs and tail. The research about locomotion and gait analysis has been explored and studied in legged or snake-like animals or robots (Ostrowski and Burdick, 1998; Guo et al., 2018). Specific body relative motion gaits may exist in squirrel locomotion while in the air, especially the tail motion (Fukushima et al., 2021), but we need to further explore the locomotion in real squirrels and apply the gaits in a more rapid duration. 3) We should try establishing a larger dataset of real squirrel motion behavior. This will support us to obtain more scientific understanding of the observed squirrel behavior and develop better control design for squirrel-like challenging operations of bio-inspired robots or other autonomous systems.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material; further inquiries can be directed to the corresponding author.

Author contributions

TM: conceptualization, investigation, methodology, coding and simulation, and writing. TZ: conceptualization, supervision, and writing (review and editing). OM: conceptualization, supervision, and writing (review and editing).

References

- Arabyan, A., and Tsai, D. (1998). A distributed control model for the air-righting reflex of a cat. *Biol. Cybern.* 79, 393–401. doi:10.1007/s004220050488
- Bingham, J. T., Lee, J., Haksar, R. N., Ueda, J., and Liu, C. K. (2014). "Orienting in mid-air through configuration changes to achieve a rolling landing for reducing impact after a fall," in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Illinois, United States, September 14–18, 2014 (IEEE), 3610.
- Bouabdallah, S., and Siegwart, R. (2007). "Full control of a quadrotor," in 2007 IEEE/RSJ international conference on intelligent robots and systems, California, United States, October 29–November 02, 2007 (IEEE).
- Butt, J. M., Chu, X., Zheng, H., Wang, X., Kwok, K.-W., and Au, K. S. (2021). Modeling and control of soft robotic tail based aerial maneuvering (stam) system: Towards agile self-righting with a soft tail. *IEEE*, 531–538. 2021 20th International Conference on Advanced Robotics (ICAR).
- Cameron, J. M., and Arkin, R. C. (1992). "Survival of falling robots," in *Mobile robots VI (SPIE)*, 1613, 91.
- Castillo-Zamora, J. J., Escareño, J., Alvarez, J., Stephant, J., and Boussaada, I. (2019). "Disturbances and coupling compensation for trajectory tracking of a multi-link aerial robot," in 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, April 23–26, 2019 (IEEE), 738.
- Clark, J., Clark, C., and Higham, T. E. (2021). Tail control enhances gliding in arboreal lizards: An integrative study using a 3D geometric model and numerical simulation. *Integr. Comp. Biol.* 61, 579–588. doi:10.1093/icb/icab073
- Dooraki, A. R., and Lee, D.-J. (2019). "Multi-rotor robot learning to fly in a bio-inspired way using reinforcement learning," in 2019 16th International Conference on Ubiquitous Robots (UR), Jeju, South Korea, June 24–27, 2019 (IEEE), 118–123.
- Featherstone, R. (1984). Robot dynamics algorithms," in *Annexe thesis digitisation project 2016 block 5*. New York: Springer.
- Fernandes, C., Gurvits, L., and Li, Z. (1993). "Optimal nonholonomic motion planning for a falling cat," in *Nonholonomic motion planning* (Springer), 379.
- Fredrickson, J. (1989). The tail-less cat in free-fall. *Phys. Teach.* 27, 620–624. doi:10.1119/1.2342893
- Fukushima, T., Siddall, R., Schwab, F., Toussaint, S. L., Byrnes, G., Nyakatura, J. A., et al. (2021). Inertial tail effects during righting of squirrels in unexpected falls: From behavior to robotics. *Integr. Comp. Biol.* 61, 589–602. doi:10.1093/icb/icab023
- Gonzalez, D. J., Lesak, M. C., Rodriguez, A. H., Cymerman, J. A., and Korpela, C. M. (2020). "Dynamics and aerial attitude control for rapid emergency deployment of the agile ground robot agro," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nevada, United States, October 25–29, 2020 (IEEE), 2577.
- Guo, X., Ma, S., Li, B., and Fang, Y. (2018). A novel serpentine gait generation method for snakelike robots based on geometry mechanics. *Ieee. ASME. Trans. Mechatron.* 23, 1249–1258. doi:10.1109/tmech.2018.2809786
- Jusufo, A., Goldman, D. I., Revzen, S., and Full, R. J. (2008). Active tails enhance arboreal acrobatics in geckos. *Proc. Natl. Acad. Sci. U. S. A.* 105, 4215–4219. doi:10.1073/pnas.0711944105
- Jusufo, A., Kawano, D. T., Libby, T., and Full, R. J. (2010). Righting and turning in mid-air using appendage inertia: Reptile tails, analytical models and bio-inspired robots. *Bioinspir. Biomim.* 5, 045001. doi:10.1088/1748-3182/5/4/045001
- Kamali, K., Bonev, I. A., and Desrosiers, C. (2020). "Real-time motion planning for robotic teleoperation using dynamic-goal deep reinforcement learning," in 2020 17th Conference on Computer and Robot Vision (CRV), Ottawa, Canada, May 13–15, 2020 (IEEE).
- Kane, T., and Scher, M. (1969). A dynamical explanation of the falling cat phenomenon. *Int. J. solids Struct.* 5, 663–670. doi:10.1016/0020-7683(69)90086-9
- Kobilarov, M. (2014). Nonlinear trajectory control of multi-body aerial manipulators. *J. Intell. Robot. Syst.* 73, 679–692. doi:10.1007/s10846-013-9934-3
- Laouris, Y., Kalli-Laouri, J., and Schwartz, P. (1990). The postnatal development of the air-righting reaction in albino rats. quantitative analysis of normal development and the effect of preventing neck-torso and torso-pelvis rotations. *Behav. Brain Res.* 37, 37–44. doi:10.1016/0166-4328(90)90070-u
- Li, T., Geyer, H., Atkeson, C. G., and Rai, A. (2019). "Using deep reinforcement learning to learn high-level policies on the atris biped," in 2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada, May 20–24, 2019 (IEEE), 263.
- Li, X., Wang, W., Wu, S., Zhu, P., and Wang, L. (2016). "A research on air posture adjustment of flying squirrel inspired gliding robot," in 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), Qingdao, China, December 03–07, 2016 (IEEE).
- Libby, T., Moore, T. Y., Chang-Siu, E., Li, D., Cohen, D. J., Jusufi, A., et al. (2012). Tail-assisted pitch control in lizards, robots and dinosaurs. *Nature* 481, 181–184. doi:10.1038/nature10710
- Liu, P., Geng, J., Li, Y., Cao, Y., Bayiz, Y. E., Langelaan, J. W., et al. (2020). "Bio-inspired inverted landing strategy in a small aerial robot using policy gradient," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nevada, United States, October 25–29, 2020 (IEEE).
- Liu, Y., and Ben-Tzvi, P. (2020). Design, analysis, and integration of a new two-degree-of-freedom articulated multi-link robotic tail mechanism. *J. Mech. Robotics* 12, 021101. doi:10.1115/1.4045842
- Newmark, N. M. (1959). A method of computation for structural dynamics. *J. Engrg. Mech. Div.* 85, 67–94. doi:10.1061/jmcea3.0000098
- Norby, J., Li, J. Y., Selby, C., Patel, A., and Johnson, A. M. (2021). Enabling dynamic behaviors with aerodynamic drag in lightweight tails. *IEEE Trans. Robot.* 37, 1144–1153. doi:10.1109/tro.2020.3045644
- Ostrowski, J., and Burdick, J. (1998). The geometric mechanics of undulatory robotic locomotion. *Int. J. robotics Res.* 17, 683–701. doi:10.1177/027836499801700701
- Rober, M. (2020). *Backyard squirrel maze 1.0- ninja warrior course*.
- Sadati, S. H., and Meghdari, A. (2017). "Singularity-free planning for a robot cat free-fall with control delay: Role of limbs and tail," in 2017 8th International Conference on Mechanical and Aerospace Engineering, Prague, Czech Republic, July 22–25, 2017 (ICMAE IEEE), 215.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Schlesinger, W. H., Knops, J. M., and Nash, T. H. (1993). Arboreal sprint failure: Lizardfall in a California oak woodland. *Ecology* 74, 2465–2467. doi:10.2307/1939598
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). *Proximal policy optimization algorithms*. arXiv preprint arXiv:1707.06347.
- Shi, F., Zhao, M., Anzai, T., Ito, K., Chen, X., Nozawa, S., et al. (2019). Multi-rigid-body dynamics and online model predictive control for transformable multi-links aerial robot. *Adv. Robot.* 33, 971–984. doi:10.1080/01691864.2019.1660710
- Shuster, M. D., et al. (1993). A survey of attitude representations. *Navigation* 8, 439.
- Siddall, R., Ibanez, V., Byrnes, G., Full, R. J., and Jusufi, A. (2021). Mechanisms for mid-air reorientation using tail rotation in gliding geckos. *Integr. Comp. Biol.* 61, 478–490. doi:10.1093/icb/icab132
- Sinervo, B., and Losos, J. B. (1991). Walking the tight rope: Arboreal sprint performance among sceloporus occidentalis lizard populations. *Ecology* 72, 1225–1233. doi:10.2307/1941096
- Singh, A., Libby, T., and Fuller, S. B. (2019). “Rapid inertial reorientation of an aerial insect-sized robot using a piezo-actuated tail,” in 2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada, May 20–24, 2019 (IEEE), 4154.
- Tsiotras, P. (1996). Stabilization and optimality results for the attitude control problem. *J. Guid. control, Dyn.* 19, 772–779. doi:10.2514/3.21698
- Weng, Z., and Nishimura, H. (2000). “Final-state control of a two-link cat robot by feedforward torque inputs,” in 6th International Workshop on Advanced Motion Control. Proceedings, Nagoya, Japan, March 30–April 01, 2000 (IEEE), 264. (Cat. No. 00TH8494).
- Wenger, G., De, A., and Koditschek, D. E. (2016). “Frontal plane stabilization and hopping with a 2dof tail,” in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, South Korea, October 09–14, 2016 (IEEE), 567.
- Xu, Z., Yamamoto, M., and Furukawa, M. (2012). “A study on falling cat landing problem using composite neuroevolution,” in Proceedings of JSPE Semestrial Meeting 2012 JSPE Autumn Conference, Kitakyushu, Japan, September 14–16, 2012 (The Japan Society for Precision Engineering), 451.
- Yeaton, I. J., Ross, S. D., Baumgardner, G. A., and Socha, J. J. (2020). Undulation enables gliding in flying snakes. *Nat. Phys.* 16, 974–982. doi:10.1038/s41567-020-0935-4
- Yilmaz, E., Zaki, H., and Unel, M. (2019). “Nonlinear adaptive control of an aerial manipulation system,” in 2019 18th European control conference (ECC), Naples, Italy, June 25–28, 2019 (IEEE), 3916.
- Yim, J. K., Singh, B. R. P., Wang, E. K., Featherstone, R., and Fearing, R. S. (2020). Precision robotic leaping and landing using stance-phase balance. *IEEE Robot. Autom. Lett.* 5, 3422–3429. doi:10.1109/lra.2020.2976597
- Young, J. W., and Chadwell, B. A. (2020). Not all fine-branch locomotion is equal: Grasping morphology determines locomotor performance on narrow supports. *J. Hum. Evol.* 142, 102767. doi:10.1016/j.jhevol.2020.102767
- Zamora, J. J. C., Escareno, J., Boussaada, I., Stephant, J., and Labbani-Igbida, O. (2020). Nonlinear control of a multilink aerial system and asekf-based disturbances compensation. *IEEE Trans. Aerosp. Electron. Syst.* 57, 907. doi:10.1109/TAES.2020.3034010
- Zhao, J., Zhao, T., Xi, N., Cintrón, F. J., Mutka, M. W., and Xiao, L. (2013). “Controlling aerial maneuvering of a miniature jumping robot using its tail,” in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 03–05, 2013 (IEEE), 3802.
- Zhao, M., Kawasaki, K., Anzai, T., Chen, X., Noda, S., Shi, F., et al. (2018). Transformable multirotor with two-dimensional multilinks: Modeling, control, and whole-body aerial manipulation. *Int. J. Robotics Res.* 37, 1085–1112. doi:10.1177/0278364918801639