MSU Tailbot: Controlling Aerial Maneuver of a Miniature Tailed Jumping Robot

Jianguo Zhao, *Student Member, IEEE*, Tianyu Zhao, *Student Member, IEEE*, Ning Xi, *Fellow, IEEE*, Matt W. Mutka, *Fellow, IEEE*, and Li Xiao, *Senior Member, IEEE*

***Abstract*—Inspired by the aerial maneuvering ability of lizards, we present the design and control of MSU (Michigan State University) tailbot—a miniature tailed jumping robot. The robot can not only wheel on the ground but also jump up to overcome obstacles. Moreover, once leaping into the air, it can control its body angle using an active tail to dynamically maneuver in mid-air for safe landings. We derive the mid-air dynamics equation and design controllers, such as a sliding mode controller, to stabilize the body at desired angles. To the best of our knowledge, this is the first miniature (maximum size 7.5 cm) and lightweight (26.5 g) robot that can wheel on the ground, jump to overcome obstacles, and maneuver in mid-air. Furthermore, tailbot is equipped with on-board energy, sensing, control, and wireless communication capabilities, enabling tetherless or autonomous operations. The robot in this paper exemplifies the integration of mechanical design, embedded system, and advanced control methods that will inspire the next-generation agile robots mimicking their biological counterparts. Moreover, it can serve as mobile sensor platforms for wireless sensor networks with many field applications.**

***Index Terms*—jumping robot, microrobotics, aerial maneuvering, biologically inspired robot, multi-modal locomotion, sliding mode control**

I. Introduction

MANY small animals or insects use jumping to locomote in environments with obstacles [1]. With the jumping ability, they can easily clear obstacles much larger than their sizes. For instance, a froghopper can jump up to 700 mm—more than one hundred times its size (about 6.1 mm) [2].

Inspired by animals' jumping abilities, we have built a series of centimeter scale jumping robots [3], [4], [5]. Actuated by a single actuator, the latest generation in [6] can jump into the air, land on the ground, steer to a desired direction, and self-right for the next jump. Moreover, it can jump more than 87 cm in height and 90 cm in distance with a mass of 23.5 grams.

Although the robot we have built can jump repetitively with good performances, it cannot control its mid-air orientation once leaping into the air. Orientation control, however, can ensure a safe landing posture for the robot to protect it from damages, especially for landing on hard surfaces. Meanwhile, if the robot is used as a sensing platform, it is desirable to control its mid-air orientation to communicate towards a desired direction [7]. Based on these two reasons, it is critical the robot's mid-air orientation can be controlled effectively.

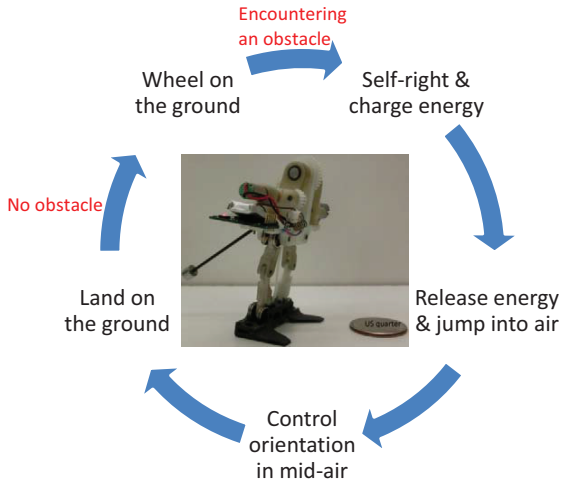


Fig. 1. Tailbot's motion cycle with the prototype in the center.

To control the mid-air orientation, we adopt biological inspirations found in nature where tails are employed for aerial maneuvering. In fact, biologists discover that lizards use their tails to dynamically maneuver in mid-air for successful landing [8]. As described in [8], they let a lizard jump from a horizontal flat surface to a vertical wall. If the surface is non-slippery, the lizard can jump to the wall without using its tail. If the surface is slippery, however, the lizard swings its tail up to control its body's orientation to still successfully land on the wall. With such an inspiration, we append an active tail to the jumping robot to control its body's mid-air orientation.

In addition to aerial maneuvering, the tailbot presented in this paper can also use two wheels to move forward on the ground, while our previous jumping robot can only use one wheel to steer: change the jumping direction [6]. The wheeling ability consumes less energy on flat grounds compared with jumping [9] and enables a faster steering speed.

With aerial maneuvering and wheeling ability, the tailbot can accomplish the motion cycle shown in Fig. 1. It can jump up to 80 cm with a mass of 26.5 grams and a maximum body size 7.5 cm. Moreover, the tailbot has onboard energy,

This work is partially supported by the National Science Foundation under Grant CNS-1320561, the U. S. Army Research Laboratory, and the U. S. Army Research Office under the grant W911NF-14-1-0327.

Jianguo Zhao and Ning Xi are with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, 48824 USA (e-mail: zhaojia1@msu.edu; xin@egr.msu.edu).

Tianyu Zhao was with the Department of Mechanical Engineering, Michigan State University, East Lansing, MI, 48824 USA. He is now with the Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213, USA.

Matt W. Mutka and Li Xiao are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, 48824 USA (e-mail: mutka@cse.msu.edu; lxiao@cse.msu.edu).

sensing and control, and wireless communication abilities, which make it an ideal mobile sensing platform for wireless sensor networks [7].

Two challenges exist for developing such a robotic platform. First, it is difficult to perform real time onboard control for aerial maneuvering since the jumping time only lasts for less than one second and the robot needs to perform real time sensing, computation, and control. Second, it is challenging to design a robot with a small size and a light weight yet having all the jumping, wheeling, and aerial maneuvering capabilities.

The research presented in this paper involves three areas: jumping robots, hybrid jumping and wheeling robots, and robots with tails for dynamic maneuvering. Relevant research in each area will be reviewed in the following.

Many jumping robots have been developed in recent years and a comprehensive review can be found in [6]. Several recent jumping robots include the frogbot [10], the EPFL jumper V1 [11], V2 [12], and V3 [13], the Grillo [14], the surveillance robot [15], the Jollbot [16], and the flea [17]. Although some of these robots can jump to a height much larger than their sizes, none of them can control their mid-air orientations.

Researchers have also built hybrid jumping and wheeling robots since wheeled locomotion is the most energy efficient method when no obstacle exists [9]. Examples include the scout robot [18], the mini-whegs [19], the rescue robot [20], the scoutbot [21], the stair climbing robot [22], and the recent sand flea robot [23]. Nevertheless, these robots have weights ranging from 0.2 kilograms for the mini-wheg to 5 kilograms for the sand flea. In contrast, our robot is designed to be less than 30 grams. With a light weight, the robot consumes less energy for the same jumping height; moreover, the robot is less susceptible to the damage from landing impacts.

In recent years, inspired by the tail's functions in animals [24], [8], researchers developed robots to investigate the merits of tails for dynamic and rapid maneuvering. Chang-Siu *et al.* added a tail to a wheeled robot to control the robot's pitch angle during free fall [25]. Johnson *et al.* also appended a tail to a legged robot to control the robot's pitch angle for safe landings from some height [26]. Demir *et al.* found that an appendage added to a quadcopter could enhance the flight stabilization [27]. Briggs *et al.* added a tail to a cheetah robot for rapid running and disturbance rejection [28]. Kohut *et al.* studied the dynamic turning of a miniature legged robot using a tail on the ground [29]. Casarez *et al.* also performed similar studies for small legged robots [30]. Although various studies exist, none of them tries to append a tail to miniature jumping robots for aerial maneuvering.

The major contributions of this paper can be summarized into two aspects. First, based on our previous robot with a tail [31], we present the design and control of a miniature robot that can wheel on the ground, jump to clear obstacles, and maneuver in mid-air. To the best of our knowledge, it is the first miniature, lightweight, and tetherless robot that has all the three capabilities. Second, we present the detailed dynamic analysis and controller design for aerial maneuvering using an active tail. Although the dynamics model has been obtained before [8], [25], only the PD controller is adopted for the stabilization control. In this paper, however, we transform the dynamics model into the standard nonlinear form and design advanced controllers such as sliding mode controllers.

The rest of this paper is organized as follows. First, the detailed robot design is presented in section II. Then, we elaborate the dynamics modeling for aerial maneuvering and optimize the design of the tail to obtain the best dynamic performance in section III. Based on the dynamics model, controllers are designed including a sliding mode controller and a PD controller in section IV. Finally, we present simulation and experimental results for aerial maneuvering and demonstrate the multi-modal locomotion abilities of the tailbot.

II. ROBOT DESIGN

The robot design, including mechanical and electrical design, will be elaborated in this section.

A. Mechanical Design

The solid model and the working principle of the tailbot are illustrated in Fig. 2. The whole robot can be divided into two parts: the tail part and the body part as shown in Fig. 2(a).

For the tail part, we implement a tail with one degree of freedom to control the body's pitch angle. Fig. 2(c) illustrates the detailed design of the tail part. The tail and the body are connected by a revolute joint actuated by a DC motor—labeled as tail motor in the figure. A motor gear is directly attached to the shaft of the motor, and a tail gear meshes with the motor gear. A carbon fiber rod with a steel block shown in Fig. 2(a) is fixed to the tail gear. Note that only part of the rod is shown in Fig. 2(c). Two teeth of the tail gear are removed to avoid the interference between the tail and the body at limit positions. This is also useful for the wheeling part that will be discussed later.

The tail can also be utilized for mode transition: from wheeling mode to jumping mode and vice versa. On one hand, when the robot wheels on the ground, the tail can push the ground to have the robot stand up and be ready for the next jump. On the other hand, when the robot stands for jumping, the tail can also push the ground to make the robot lie down for wheeling. The detailed process can be found in the experimental part for mode transition.

The design for the tail part should ensure rapid control of the body's orientation since the jumping process lasts for a short time (less than one second for a one meter jumping height). Therefore, an optimal design will be performed to obtain the optimal parameters for the tail based on the dynamics model in the next section.

The body part comes from our previous jumping robot design [6], [32]. It consists of two major components: a jumping mechanism to convert potential energy stored in the robot to kinetic energy for takeoff and an energy mechanism to store and release energy. The working principles for these two mechanisms are illustrated in Fig. 2(e) and (f), respectively. Since the detailed design can be found in [6], we only briefly describe it here.

The jumping mechanism is a symmetric six bar mechanism. As shown in Fig. 2(b) and (e), two identical kinematic chains connect the foot and the platform via revolute joints. Each

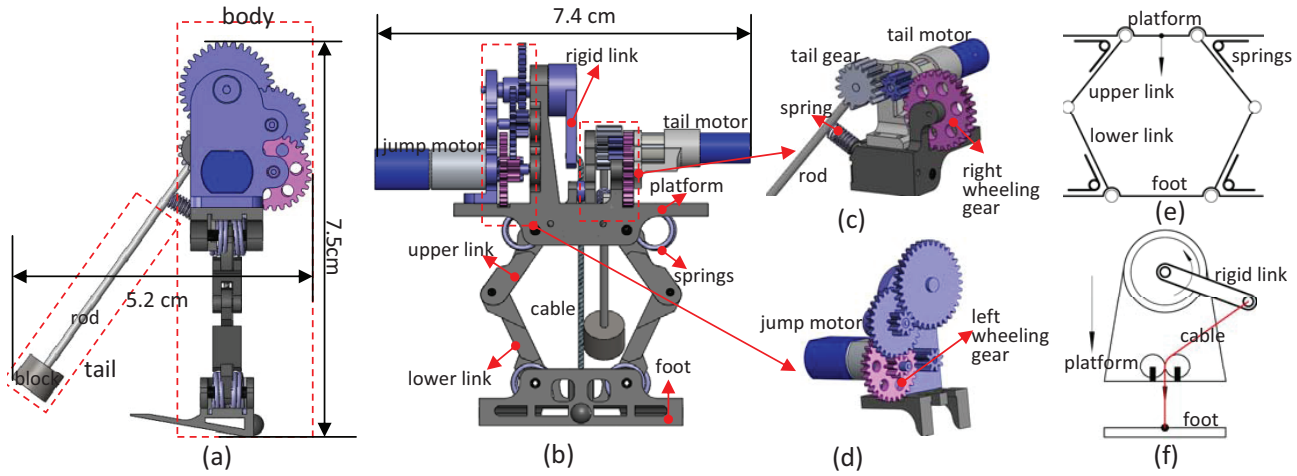


Fig. 2. Mechanical design of the tailbot: (a) Left view with height and width, and the robot is divided into the body and tail part encircled by two rectangles; (b) Front view with length, and two major parts encircled by two rectangles are shown in (c) and (d), respectively; (c) Section view of the tail part; (d) Section view of the gear train part for the energy mechanism; (e) Working principle of the jumping mechanism; (f) Working principle of the energy mechanism.

kinematic chain contains an upper link and a lower link connected by a revolute joint as well. Torsion springs are placed at four joints of the foot or the platform. In this way, if a vertical downward force is applied at the center of the platform, the platform will move towards the foot, and the energy will be stored in the springs. The stored energy is released once the force is removed. The foot has a tilted angle of 15° to make it take off at an angle of 75°.

The force can be applied and removed using an energy mechanism shown in Fig. 2(b) and (f). The gear train, a main component as shown in Fig. 2(d), is actuated by a DC motor—labeled as jump motor, and the output of the gear train is connected to a rigid link via a one way bearing, which is not shown in the figure since it is inside the rigid link. A polyester cable connects the distal end of the rigid link to the foot. As the link rotates, the cable drags the platform to move towards the foot. Once the link passes a critical position, the energy is released due to the one-way bearing [6].

In addition to the jumping mechanism and energy mechanism, the robot has two wheeling gears as shown in Fig. 2(c) and (d), which are employed for differential drive on the ground. Note that the robot's wheeling posture, once it lands on the ground, can be guaranteed. In fact, the robot has a rectangular shape with two sides much larger than the other four; therefore, it will land on the ground with one of the two large sides. It can wheel if it lands with the side having the two wheeling gears on the ground. If it lands with the other large side, the tail can rotate to turn the robot upside down so that the robot can still wheel on the ground.

The wheeling part does not require extra actuations. The left wheeling gear is part of the gear train for the energy mechanism as shown in Fig. 2(d). If the jump motor rotates in one direction, the rigid link will rotate to charge and release the energy. But if the motor rotates in the opposite direction, the rigid link will stay still due to the one-way bearing. In this case, the left wheeling gear is used for the wheeling motion.

The right wheeling gear is actuated by the tail motor as shown in Fig. 2(c). Since the tail gear has two teeth removed, once the tail reaches the left limit position shown in Fig. 2(c), the clockwise rotation of the motor gear cannot actuate the tail. In this case, the right wheeling gear can be used for the wheeling motion. To switch from wheeling mode to jumping mode, a small extension spring placed perpendicular to the tail at the left limit position can re-engage the tail gear and motor gear if the tail motor rotates counterclockwise.

The turning motion is realized by actuating one wheeling gear while keeping the other one still. Therefore, the robot can turn in both counterclockwise and clockwise directions.

B. Electrical Design

A miniature embedded system is designed to enable tetherless or autonomous operation of the tailbot. It is implemented by a printed circuit board with a dimension of $2.5cm \times 2.5cm$ and a mass of 1.3 g. The system has four parts: the central processing unit, the sensing unit, the actuation unit, and the power supply unit. Fig. 3 illustrates the architecture of the system.

A microcontroller (ATmega128RFA1 from Atmel) serves as the central processing unit, which has an integrated 2.4GHz Zigbee transceiver. It enables the two-way data transmission between a computer and the robot. Moreover, multiple robots are able to communicate with each other to form a mobile sensor network.

The sensing elements contain a tri-axis accelerometer, a tri-axis gyroscope, and a tri-axis magnetic compass. We use a single chip for the former two sensors (MPU-6050 from Invensense) and another chip for the compass (HMC5883L from Honeywell). The accelerometer can detect the free fall, while the gyroscope can feedback the body's angle and angular velocity to the microcontroller. The compass can feedback the heading direction when the robot wheels on the ground.

The actuation unit is a dual H-Bridge motor driver with pulse width modulation ability (MC34933 from Freescale) to control both the jump motor and tail motor. A FullRiver 50mAh LiPo battery—after being regulated to 3.3 V—powers the whole robotic system. The charging time for such a battery

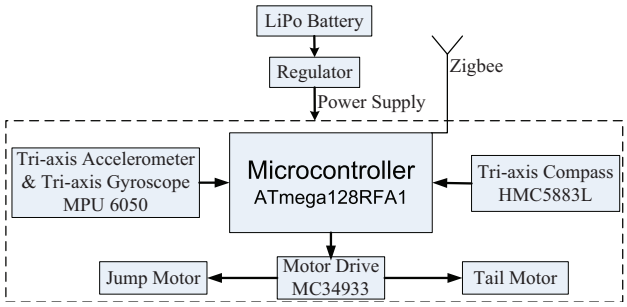


Fig. 3. The architecture of the embedded control system.

is less than 15 minutes. However, the battery can power the robot's jump for more than 50 times.

The embedded system and the battery are sandwiched between the platform and the tail motor (Fig. 2(b)). With such a system, the robot can perform thetherless operations. Commands can be sent to the robot through Zigbee to control its working mode such as jumping or wheeling. Moreover, the robot can also perform autonomous aerial maneuvering once it leaps into the air. The details will be discussed in the experimental section.

III. DYNAMICS MODEL AND TAIL OPTIMIZATION

A. Dynamics Model

Successful aerial maneuvering requires the robot's mid-air dynamics model, which belongs to the dynamics for coupled rigid bodies during free fall. In the last decade, researchers designed controllers for two coupled rigid bodies during free-fall. Mather and Yim investigated the controlled fall for modular robots [33]. Yang *et al.* [34] modeled two rigid bodies connected by a spherical joint in three dimensional space and designed controllers using input-output linearization. Later, they studied the control of two rigid bodies connected by a universal joint [35]. Agrawal and Zhang utilized differential flatness to control two rigid bodies connected by a revolute joint [36]. Most recently, Chang-Siu *et al.* [37] studied the nonlinear control of a two link tailed robot with two degree-of-freedom actuation.

The tailbot shown in Fig. 2(a) can be abstracted as shown in Fig. 4, where the body is connected to the tail via an actuated revolute joint at point C. Suppose the center of mass for the tail, body, and whole robot be at point A, B, and O, respectively. We attach a frame $OXYZ$ to the robot's center of mass O with X axis along the horizontal direction, Y axis along the vertical direction, and Z axis determined by the right hand rule. The body's three orientation angles—roll, pitch, and yaw—are shown at the lower right part of Fig. 4. Note that the angles are defined with respect to frame $OXYZ$ that will move with the robot. By actively swinging the tail, the body's pitch angle can be controlled. In this section, we obtain the system's dynamics equation and transform it into a standard nonlinear system.

For the system in Fig. 4, we use the parameters listed in Table I in the following discussions. Denote the coordinates

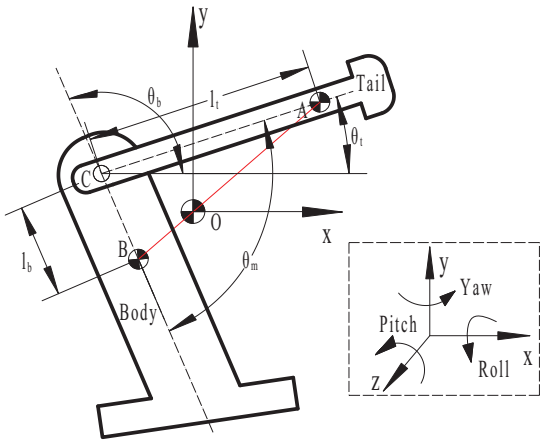


Fig. 4. The schematic of the tailbot in mid-air for dynamics modeling, where the body and the tail are connected by a revolute joint at point C (Fig. 2(a) shows the tail and body part with solid models).

TABLE I
LIST OF PARAMETERS FOR DYNAMICS MODELING

m_b	body mass
m_t	tail mass
l_b	length of link BC
l_t	length of link AC
θ_b	body angle with respect to the horizontal line
θ_t	tail angle with respect to the horizontal line
I_b	body moment of inertial with respect to C
I_t	tail moment of inertial with respect to C

for point A and point B in frame $OXYZ$ as $\vec{A}$ and $\vec{B}$, respectively. They can be obtained using the relation $|AO|/|BO| = m_b/m_t$ and trigonometric equations in $\triangle ABC$.

The Euler-Lagrange method is used to obtain the dynamic equations. For this method, the Lagrangian for the system should be first derived. Because frame $OXYZ$ is fixed with the robot, its translational motion decouples from the rotational motion once it jumps into air [38]. Since the translational motion is a simple projectile motion [6], we only consider the rotational motion for aerial maneuvering. Without the translational motion, the robot's potential energy is zero. Therefore, the Lagrangian is just the system's kinetic energy:

$$\mathcal{L} = \frac{1}{2}I_t\dot{\theta}_t^2 + \frac{1}{2}m_t||\dot{\vec{A}}||_2^2 + \frac{1}{2}I_b\dot{\theta}_b^2 + \frac{1}{2}m_b||\dot{\vec{B}}||_2^2$$
$$= \frac{1}{2}[I_t\dot{\theta}_t^2 + I_b\dot{\theta}_b^2 + \frac{m_t m_b(l_t^2\dot{\theta}_t^2 + l_b^2\dot{\theta}_b^2 - 2l_t l_b\dot{\theta}_t\dot{\theta}_b\cos\theta_m)}{m_t + m_b}]$$

where $\theta_m = \pi + \theta_t - \theta_b$, shown in Fig. 4, is the actuator's rotation angle. Neglecting the air resistance and applying the Euler-Lagrange method, we obtain the dynamics equation as:

$$M\ddot{\theta}_t - L\cos\theta_m\ddot{\theta}_b - L\sin\theta_m\dot{\theta}_b^2 = \tau \quad (1)$$
$$N\ddot{\theta}_b - L\cos\theta_m\ddot{\theta}_t + L\sin\theta_m\dot{\theta}_t^2 = -\tau \quad (2)$$

where

$$M = I_t + \frac{m_t m_b l_t^2}{m_t + m_b}, \quad N = I_b + \frac{m_t m_b l_b^2}{m_t + m_b}, \quad L = \frac{m_t m_b l_t l_b}{m_t + m_b}$$

τ is the actuation torque from the motor. Note that we only have one τ for external forces since only one actuator is used at the revolute joint.

For the system described by Eqs. (1) and (2), if both θ_t and θ_b should be controlled to desired values, then the system is underactuated since only one input τ exists. In this paper, however, we only care about the robot's body angle θ_b. To control θ_b, Eqs. (1) and (2) should be transformed into a single equation by eliminating θ_t, but this is impossible due to the nonlinear coupling between θ_t and θ_b. Nevertheless, Eqs. (1) and (2) can be converted to a new equation with θ_m and $\dot{\theta}_m$ as the state variable using the following steps.

First, we solve $\ddot{\theta}_t$ and $\ddot{\theta}_b$ from Eqs. (1) and (2):

$$\ddot{\theta}_t = \frac{-SL\dot{\theta}_t^2 \cos\theta_m + SN\dot{\theta}_b^2 + R\tau}{T} \tag{3}$$

$$\ddot{\theta}_b = \frac{-SM\dot{\theta}_t^2 + SL\dot{\theta}_b^2 \cos\theta_m - Q\tau}{T} \tag{4}$$

where $Q = M - L\cos\theta_m$, $R = N - L\cos\theta_m$, $S = L\sin\theta_m$, and $T = MN - L^2\cos^2\theta_m$. Since $T \geq MN - L^2 > 0$, there is no singularity for using T as the denominator in Eqs. (3) and (4). From (4) − (3) and $\ddot{\theta}_m = \ddot{\theta}_t - \ddot{\theta}_b$, we have:

$$\ddot{\theta}_m = \frac{SQ\dot{\theta}_t^2 + SR\dot{\theta}_b^2}{T} + \frac{Q+R}{T}\tau \tag{5}$$

Second, we utilize the conservation of angular momentum to eliminate both $\dot{\theta}_t$ and $\dot{\theta}_b$ in Eq. (5) by expressing them as a function of $\dot{\theta}_m$. In fact, the angular momentum for the total system can be obtained as:

$$H_0 = (M - L\cos\theta_m)\dot{\theta}_t + (N - L\cos\theta_m)\dot{\theta}_b = Q\dot{\theta}_t + R\dot{\theta}_b$$

If the air resistance is negligible, then H_0 is a constant. Since $\dot{\theta}_m = \dot{\theta}_t - \dot{\theta}_b$, $\dot{\theta}_t$ and $\dot{\theta}_b$ can be solved as follows:

$$\dot{\theta}_t = \frac{R\dot{\theta}_m}{Q+R} + \frac{H_0}{Q+R} \tag{6}$$

$$\dot{\theta}_b = \frac{-Q\dot{\theta}_m}{Q+R} + \frac{H_0}{Q+R} \tag{7}$$

Finally, plugging Eqs. (6) and (7) into (5), we obtain:

$$\ddot{\theta}_m = \frac{QRS\dot{\theta}_m^2 + SH_0^2}{T(Q+R)} + \frac{Q+R}{T}\tau \tag{8}$$

Let $x = [\theta_m, \dot{\theta}_m]^T$ and $u = \tau$. Since we want to control the body's angle θ_b, let the output $y = \theta_b$. Then from Eqs. (8) and (7), the system can be written as:

$$\dot{x} = f(x) + g(x)u \tag{9}$$

$$y = h(x) \tag{10}$$

with

$$f(x) = \begin{bmatrix} x_2 \\ \dfrac{QRSx_2^2 + SH_0^2}{T(Q+R)} \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 \\ \dfrac{Q+R}{T} \end{bmatrix}, \tag{11}$$

$$h(x) = \int \frac{-Qx_2 + H_0}{Q+R} + \theta_b(0) \tag{12}$$

where $\theta_b(0)$ is the initial angle for θ_b.

With the dynamics equation, the control goal is to design a state or output feedback controller to stabilize the output at some desired constant value $y(t) = \theta_b^*$. Since the initial angular momentum is difficult to measure, the case of zero angular momentum ($H_0 = 0$) will be considered in this paper.

The same dynamics in Eqs. (1) and (2) has been obtained before using Newtonian mechanics [8], [25], [26]. Nevertheless, none of them tries to formulate the problem in the standard form shown in Eqs. (9) and (10), which facilities the controller design in the next section.

B. Tail Optimization

With the dynamics model, the tail can be optimally designed to achieve the best dynamic performance in mid-air. The performance can be measured by indices such as rising time, settling time, or steady state error, etc. Such indices depend on the designed controller and the robot's parameters such as those listed in Table I. Therefore, the general optimization problem can be formulated as:

$$\max \text{ or } \min \quad \mathcal{M} = f(u(t), p)$$

where $\mathcal{M}$ is a specific performance index, $u(t)$ is the control input, and p represents the robot's parameters.

This optimization represents a new problem in optimal design since it blends both the controller and the robot's parameters, and we need to solve them together. As our first step, however, we focus on the mechanism optimization by solving a simplified version without the controller. Specifically, we aim to obtain an optimal tail design that maximizes the change of body angle θ_b for a given short time with the tail motor actuated by a constant rated voltage supply.

With a constant rated voltage supply, the torque τ is related to its angular speed $\dot{\theta}_m$ by $\tau = \tau_s(1 - \dot{\theta}_m/\omega_0)$, where τ_s is the motor's stall torque and ω_0 is the no-load angular speed. In this case, Eq. (8) becomes a second order ordinary differential equation for θ_m:

$$\ddot{\theta}_m = \frac{QRS\dot{\theta}_m^2}{T(Q+R)} + \frac{Q+R}{T}\tau_s(1 - \frac{\dot{\theta}_m}{\omega_0}) \tag{13}$$

From Eq. (13), we can solve $\theta_m(t)$. With $\theta_m(t)$, the body angle's trajectory can be derived from Eq. (12).

Using Eq. (13), we perform the optimal design in four steps: identifying optimization variables, formulating the objective function, obtaining the constraints, and solving the constraint optimization problem.

The parameters for the tail part design include m_t, l_t, I_t, and the speed reduction ratio r between the motor gear and the tail gear (Fig. 2(c)). Since the carbon fiber rod in the tail part has a negligible mass compared with the steel block, the tail's moment of inertial I_t can be approximated as $I_t = m_t l_t^2$. Therefore, the optimization variables are m_t, l_t, and r. Note that m_b, l_b, and I_b are known based on our previous jumping robot [6]. To simplify the design, we choose the tail motor empirically (GH6123S from Gizmoszone) by estimating the required torque and speed.

The objective function for the optimization problem is chosen to be the change of θ_b. Specifically, an optimal design is one that maximizes the change of θ_b for a small fixed time period (0.1 s) under a given constant rated voltage. To achieve this goal, we consider θ_b as a function of m_t, l_t, and r from Eq. (13), and denote this function as $\theta_b(t) = f(m_t, l_t, r)$.

The optimization constraints are derived as follows. Since a large weight decreases the jumping performance and increases the landing impact, we let $m_t \leq 0.15m_b$. On the other hand, since m_t cannot be too small to perform effective aerial maneuvering, we constrain $m_t \geq 0.05m_b$. With similar practical reasons, let $0.75L_b \leq l_t \leq 1.5L_b$ and $0.1 \leq r \leq 10$ with $L_b = 7.5$ cm being the maximum body length (L_b is different from l_b in Fig. 4).

Based on the previous discussions, the optimal design problem is formulated as:

$$\begin{aligned} \max \quad & \theta_b(0.1) = f(m_t, l_t, r) \\ \text{subject to} \quad & 0.05m_b \leq m_t \leq 0.15m_b, \qquad 0.1 \leq r \leq 10 \\ & 0.75L_b \leq l_t \leq 1.5L_b, \qquad \theta_b(0) = constant \end{aligned}$$

The optimization problem is solved using the Optimization toolbox in Matlab. Specifically, the *fmincon* function with the interior point algorithm is adopted. To avoid local minima, we run the optimization algorithm 50 times with random initial points. For all of them, the optimal result is $m_t = 1.3$ g, $l_t = 6.8$ cm, and $r = 2.2$. With the optimal parameters, θ_b can change 80 degrees in 0.1 s. To accommodate the available off-the-shelf gears, we choose $r = 2$. In this case, the value of $\theta_b(0.1)$ only decreases about 0.1%.

Based on the optimal design, the tail's parameters are chosen as follows. The length of the carbon fiber rod is $l_t = 6.8$ cm, and the mass for the steel block is $m_t = 1.3$ g. For the gear train that actuates both the tail and the right wheeling gear (Fig. 2(c)), the motor gear has 8 teeth, the tail gear has 16 teeth since $r = 2$, and the right wheeling gear has 28 teeth which is the same with the left wheeling gear.

IV. CONTROLLER DESIGN

Based on the dynamics model, controllers are designed to stabilize the body angle θ_b at a desired constant angle θ_b^*. Since the embedded control system attached to the robot body can only feedback body's angular velocity $\dot{\theta}_b$ and consequently the output $y = \theta_b$, the state $x = [\theta_m, \dot{\theta}_m]^T$ is unavailable. Therefore, we cannot design state feedback controllers for the system. In this section, we first transform the system to a normal form with $\dot{\theta}_b$ and θ_b as the transformed states. Based on the normal form, a sliding mode controller is designed to regulate θ_b to θ_b^*. Additionally, we also discuss a special case to design a PD controller.

A. Normal Form

We follow the process in [39] to transform the system described by Eqs. (9) and (10) to the normal form. First, the relative degree for the system is obtained by differentiating the output y with respect to time until the control input u appears. Direct differentiating y with respect to time, we have $\dot{y} = L_f h(x) + L_g h(x)u$ with $L_f h(x) = -Qx_2/(Q+R)$ and $L_g h(x) = 0$ the Lie Derivative of h along f and g, respectively. Continuing the differentiation, we have $\ddot{y} = L_f^2 h(x) + L_g L_f h(x)u$ with

$$L_f^2 h(x) = -\frac{Q^2RS + ST(R-Q)}{T(Q+R)^2}x_2^2, \quad L_g L_f h(x) = -\frac{Q}{T}$$

Since $L_g h(x) = 0$ and $L_g L_f h(x) \neq 0$, the system has a relative degree of two, which is equal to the dimension of the system. Therefore, we can transform the state equation to the normal form using the following map [39]:

$$z = \begin{bmatrix} h(x) \\ L_f h(x) \end{bmatrix} = \begin{bmatrix} -\int \dfrac{Qx_2}{Q+R} + \theta_b(0) \\ -\dfrac{Qx_2}{Q+R} \end{bmatrix} = \begin{bmatrix} \theta_b \\ \dot{\theta}_b \end{bmatrix} \tag{14}$$

The transformed new system in normal form is thus:

$$\dot{z} = A_c z + B_c[L_g L_f h(x)\, u + L_f^2 h(x)] \tag{15}$$

$$y = C_c z \tag{16}$$

where

$$A_c = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_c = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

If we want to stabilize the output—the body angle θ_b—at a constant angle θ_b^*, which requires $z^* = [\theta_b^*, 0]^T$. Let $\bar{z} = z - z^*$. Since $A_c z^* = [0, 0]^T$, Eq. (15) can be rewritten as

$$\dot{\bar{z}} = A_c \bar{z} + B_c[L_g L_f h(x)\, u + L_f^2 h(x)] \tag{17}$$

Therefore, regulating θ_b to θ_b^* is equivalent to stabilize the system in Eq. (17) to the origin. If $\bar{z} = [\bar{z}_1, \bar{z}_2]^T$, Eq. (17) can be written out as

$$\dot{\bar{z}}_1 = \bar{z}_2, \quad \dot{\bar{z}}_2 = pu + q \tag{18}$$

where

$$p = -\frac{Q}{T}, \quad q = -\frac{Q^2RS + ST(R-Q)}{TQ^2}\bar{z}_2^2$$

Note that p and q are functions of $\bar{z}$ and $x_1 = \theta_m$, although explicit forms are not written out.

B. Sliding Mode Controller

Sliding mode control attains the control goal in two steps. First, the system's trajectory undergoes a reaching phase to a sliding surface. Second, the trajectory is constrained to the sliding surface to approach the equilibrium during the sliding phase [39]. Let the sliding surface be $s = a\bar{z}_1 + \bar{z}_2$ with $a > 0$ a constant which determines the convergent speed during the sliding phase.

As shown in the following, we have $Q > 0$ and the following two items are upper bounded with the bounds denoted by k_1 and k_2, respectively:

$$\left|\frac{T}{Q}\right| \leq k_1, \quad \left|\frac{Q^2RS + ST(R-Q)}{Q^3}\right| \leq k_2$$

First we show that $Q > 0$. In fact,

$$Q = M - L\cos\theta_m \geq M - L = I_t + \frac{m_t m_b l_t (l_t - l_b)}{m_t + m_b} > 0$$

since l_t is much larger than l_b in our implementations. The bound for the first item $|T/Q|$ can be obtained as:

$$\left|\frac{T}{Q}\right| = \frac{MN - L^2\cos^2\theta_m}{M - L\cos\theta_m} \leq \frac{MN}{M-L} := k_1$$

The bound for the second item is:

$$\left|\frac{Q^2RS+ST(R-Q)}{Q^3}\right| \le \frac{Q^2|R||S|+T|S||R-Q|}{Q^3}$$
$$\le \frac{(N+L)L}{M-L}+\frac{LMN|N-M|}{(M-L)^3} := k_2$$

Based on the above results, we have the following theorem:

Theorem 1. *The following controller can asymptotically stabilize the system represented by Eq.* (17) *at the origin:*

$$u = (ak_1|\bar{z}_2| + k_2\bar{z}_2^2 + k_3)\ sgn(a\bar{z}_1+\bar{z}_2) \tag{19}$$

where $k_3 > 0$ is a constant and $sgn(\cdot)$ is the sign function.

Proof. We first show that the system will reach the sliding surface. Define a Lyapunov function as $V = s^2/2$, then:

$$\dot{V} = s\dot{s} = s(a\dot{\bar{z}}_1+\dot{\bar{z}}_2) = s(a\bar{z}_2+q)+pus$$

Since

$$\left|\frac{a\bar{z}_2+q}{p}\right| = \left|-\frac{aT}{Q}\bar{z}_2+\frac{Q^2RS+ST(R-Q)}{Q^3}\bar{z}_2^2\right|$$
$$\le ak_1|\bar{z}_2|+k_2\bar{z}_2^2$$

we have

$$\dot{V} \le |s||a\bar{z}_2+q|+pus$$
$$\le -p(ak_1|\bar{z}_2|+k_2\bar{z}_2^2)|s|+pus = k_3p|s|$$

Since $p = -Q/T < 0$, we have $\dot{V} < 0$ for any $s \neq 0$. Therefore, the system will reach the sliding surface in finite time. After that, it will be governed by $\dot{\bar{z}}_1 = -a\bar{z}_1$ to make the state approach the origin. □

The sliding mode controller in Eq. (19) only needs the feedback of transformed state $z_1 = \theta_b$ and $z_2 = \dot{\theta}_b$, which are available from the gyroscope. Moreover, during the implementation, we can use a simplified controller

$$u = k\ \text{sgn}(a\bar{z}_1+\bar{z}_2) \tag{20}$$

with $k > ak_1|\bar{z}_2| + k_2\bar{z}_2^2 + k_3$. The stability for the system using this controller can also be verified by showing $\dot{V} < 0$.

C. Proportional-Derivative (PD) Controller

Besides the sliding mode controller, we examine the controller design for a special case when $l_b = 0$, which is detailed in [26]. This special case approximates the situation when the robot has a very small l_b such as the robots in [25] and [26]. In this case, since $L = m_tm_bl_tl_b/(m_t+m_b) = 0$, the dynamics equation (2) is simplified to $N\ddot{\theta}_b = -\tau$. Since $N = I_b$, the simplified system in state space form is thus: $\dot{\bar{z}} = A_c\bar{z} - B_cu/I_b$. For this system, a PD controller can be designed as:

$$u = I_bK\bar{z} \tag{21}$$

where $K = [K_p,\ K_d]$ is designed to make $A_c - B_cK$ Hurwitz.

Similar to the sliding mode controller, the PD controller only needs the feedback of the transformed state. Therefore, we will include experimental results for the PD controller in the next section since l_b for our robot is close to zero.

V. TESTING RESULTS

With the designed robot, we first simulate the designed controllers and then conduct experiments for aerial maneuvering. In addition, we also perform experiments for mode transitions.

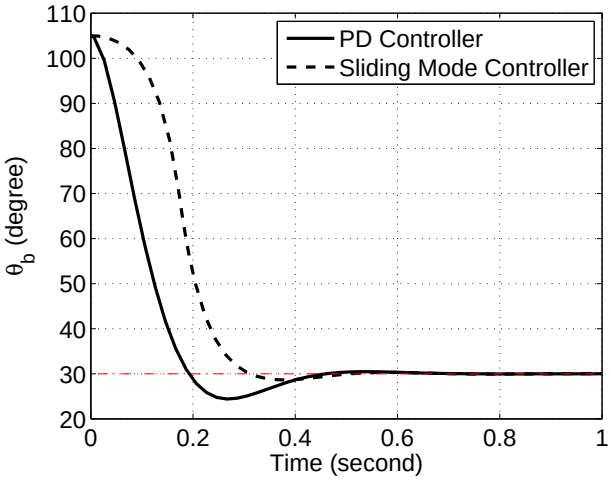


Fig. 5. Simulation results for the PD controller and the sliding mode controller. Each curve shows the trajectory for θ_b with respect to time for each controller.

A. Simulation Results for Aerial Maneuvering

We simulate the aerial maneuvering using Matlab/Simulink to validate the dynamics model and controller design. Based on our robot design, the following parameter values are chosen for the simulation: $m_t = 1.3\times10^{-3}kg$, $m_b = 25.2\times10^{-3}kg$, $l_t = 6.8\times10^{-2}m$, $l_b = 1.0\times10^{-2}m$, $I_t = 5.8\times10^{-6}kg\cdot m^2$, and $I_b = 1.0\times10^{-5}kg\cdot m^2$.

The initial value for θ_b is chosen as $105°$, which means the robot takes off at an angle of $75°$. The initial value for θ_m is chosen as $180°$, which corresponds to the setup in experiments. We let the desired body angle $\theta_b^* = 30°$ to make the robot land on ground with one of the two largest sides.

For the PD controller, we choose $K = [240,\ 20]^T$ to make the system approach θ_b^* within 0.2 second. With all the parameters and the dynamics model, the trajectory of θ_b with respect to time is obtained and shown by the solid line in Fig. 5.

For the sliding mode controller, we first calculate the bounds for the two items as: $k_1 = 2.2\times10^{-5}$ and $k_2 = 1.96\times10^{-6}$. The controller with saturation function is employed to reduce the chattering

$$u = (ak_1|\bar{z}_2| + k_2\bar{z}_2^2 + k_3)\ \text{sat}(\frac{a\bar{z}_1+\bar{z}_2}{\mu}) \tag{22}$$

where $\text{sat}(\cdot)$ is the saturation function and μ a positive constant [39]. Let $a = 10$, $k_3 = 5\times10^{-4}$, and $\mu = 1$, which are chosen to make the response have a similar profile compared with the PD controller. The trajectory of θ_b for the sliding mode controller is shown as the dash line in Fig. 5.

Based on the simulation results, performances of the two controllers are compared and major metrics are listed in Table

II. The sliding mode controller has a larger rising time 0.31s compared with 0.21s for the PD controller. However, if we consider the settling time as when θ_b gets and stays within 10% of θ_b^*, then the sliding mode controller settles faster. The steady state errors for both controllers are zero, which verifies the stability of the system. In addition, the PD controller has a larger percent of overshoot compared with the sliding mode controller.

TABLE II
SIMULATION RESULTS COMPARISON FOR THE TWO CONTROLLERS

Metric	PD controller	Sliding mode controller
Rising time (s)	0.21	0.31
Settling time (s)	0.37	0.26
Steady state error (degree)	0	0
Percent of overshoot (%)	18.66	4.56

B. Experimental Results for Aerial Maneuvering

Based on the design presented in section II, the robot prototype is built. With the prototype and designed controllers, we conduct aerial maneuvering experiments to test the robot's performance.

The experiments are set up as follows. We let the robot jump in front of a white board with height marks. To minimize the slippage during takeoff, the robot is placed on a sand paper to provide sufficient friction. Furthermore, the initial position of the tail is carefully adjusted onto the top of the body so that the initial angular momentum can be minimized at takeoff. Note that, however, the initial angular momentum cannot be eliminated as will be explained in the experimental results.

All the sensing, computation, and control are implemented by C programming in the embedded system discussed in section II-B. The microcontroller, ATmega128RFA1, runs at a speed of 8MHz. In each control loop, the microcontroller first samples raw angular velocities from the gyroscope. Then it obtains the current roll, pitch, and yaw angle by numerically integrating the angular velocity using the Runge-Kutta method. Based on the current pitch angle, the control input is computed and the tail motor is actuated using the computed control command. The time for one control loop is three millisecond, and the system can achieve a bandwidth of 333Hz, which is sufficient for feedback control as will be seen in the experimental results. Note that we neglect the drift problem for the gyroscope since the control time for our experiment is less than one second.

Three sets of experiments are conducted. First, to compare with results under active tail control, five jumping experiments with the tail disabled are performed. Second, five experiments with active tail control using the PD controller are conducted. Third, another five experiments using the sliding mode controller are carried out.

During each experiment, we use a Casio Exilim EX-ZR400 high-speed camera with a frame rate of 240 frames/s to record the entire motion. Meanwhile, the body's instantaneous pitch, roll, and yaw angles—defined in Fig. 4—are recorded by the embedded control system, which are sent back to a computer wirelessly after landing. The initial pitch angle is obtained using the accelerometer. Since the roll and yaw angle cannot be controlled, we only measure the change for each angle starting from an initial zero angle.

1) Jumping without Tail Actuation: For the five experiments without tail actuation, Fig. 6(a) shows the robot's moving trajectory for one of them (jump 2). Referring to the height marked on the left side of the white board, we visualize that the robot jumps 32 inches (81.3 cm). Without actuating the tail, the robot's pitch angle only changes slightly due to a small initial angular momentum. The landing posture of the robot is bad since landing on its feet may cause the robot to bounce severely on the ground, increasing the probability of damage.

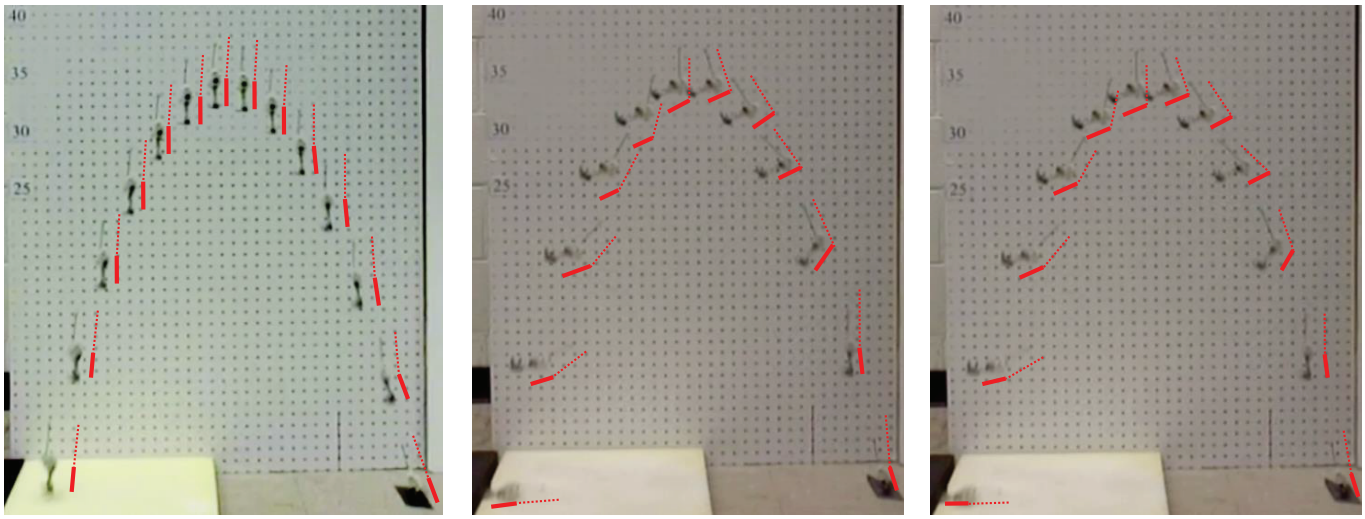
For jumping with the tail disabled, if there is no initial angular momentum and the air resistance is neglected, the pitch, roll, and yaw angles should be constants all the time. Fig. 7 shows the robot's mid-air orientation for the five jumps, where the pitch, roll, and yaw angles are plotted with respect to time. For all the jumps, the pitch angles do not change too much since only a small initial angular momentum affects the pitch angles. In this case, the pitch angles should change linearly; however, this is not the case in Fig. 7 because the tail can slightly rotate even though it is not actuated due to the backlash between the tail gear and the motor gear. From plot (b) and (c) in Fig. 7, the roll and yaw angles change almost linearly in random manners. These imply that rotations about other axes are affected by random initial angular momentums that the robot might have after takeoff. Nevertheless, the changes in roll and yaw are relatively small, and they will not cause the robot to flip around in mid-air to affect its landing posture significantly.

2) Aerial Maneuvering with the PD Controller: Experiments for tail actuated jumping are conducted as follows. Initially, the robot stands on the ground, and the accelerometer obtains the initial pitch angle. Once the robot jumps up, the accelerometer detects the free fall and triggers the controller to regulate the body's pitch angle to a desired value by swinging the tail. In mid-air, the gyroscope measures the body's instantaneous angular velocity, and the microcontroller calculates the body's current angle by numerical integrations. With the current angle, current angular velocity, and the desired angle, the controller computes a voltage output for the tail motor to perform controlled aerial maneuvering.

For the PD controller, we tune the values for K_p and K_d by letting the robot fall from a given height since the dynamics is the same as the case when the robot jumps. Using the Ziegler-Nichols tuning method, we obtain values for the controller as $K_p = 40$ and $K_d = 0.3$.

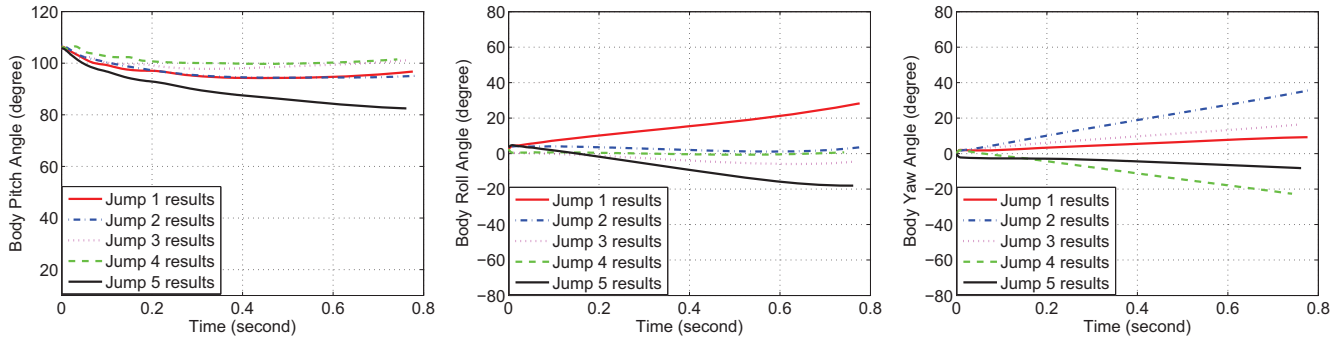
Fig. 6(b) shows one of the motion trajectories (jump 1) for the robot using the PD controller. Same to the simulations, the desired angle of θ_b is set to be 30°. From the trajectory, the robot rotates its tail to regulate the pitch angle to 30°. Eventually, the robot lands safely on a sponge pad with one of its two large sides.

Fig. 8 shows the robot's pitch, roll, and yaw angles with respect to time. Combining Fig. 6(b) with Fig. 8(a), we observe that the tail rotates counter-clockwisely to make the pitch angle θ_b approach 30°, which is the horizontal yellow solid line in



(a) (b) (c)

Fig. 6. Aerial maneuvering results from video frames show the robot trajectory in a single image for three cases. A schematic view for each robot in all the three images is added for illustration purposes. The dashed lines represent the tail, while the solid lines represent the body. (a) the tail is not actuated; (b) the tail is controlled by the PD controller; (c) the tail is controlled by the sliding mode controller. Note that the robot jumps from right to left in the figure.



(a) (b) (c)

Fig. 7. Experimental results for aerial maneuvering when the tail is not actuated: (a) the body's pitch angle with respect to time; (b) the body's roll angle with respect to time; (c) the body's yaw angle with respect to time.

Fig. 8(a). The system has a rising time about 0.2s, i.e., the pitch angle reaches 30° in 0.2s. After about 0.3s, it reaches a steady-state value about 28°, although a small oscillation exists. The reason for this oscillation is that the smallest voltage applied to the tail motor is set to be the minimum voltage which the motor overcomes its static friction. This means the motor will not stop running even if the body has reached the desired angle. Additionally, the roll and yaw plots in Fig. 8(b) and (c) provide information about rotations in uncontrolled axes. Unlike the case when the tail is disabled, some experiments have nonlinear curves such as the yaw angle for jump 3. The reason is that although the tail is only used to control the pitch angle, manufacturing or assembly error may induce the cross coupling between the tail and yaw or roll angle. Therefore, the rotation of the tail may slightly influence the change of yaw and roll angle.

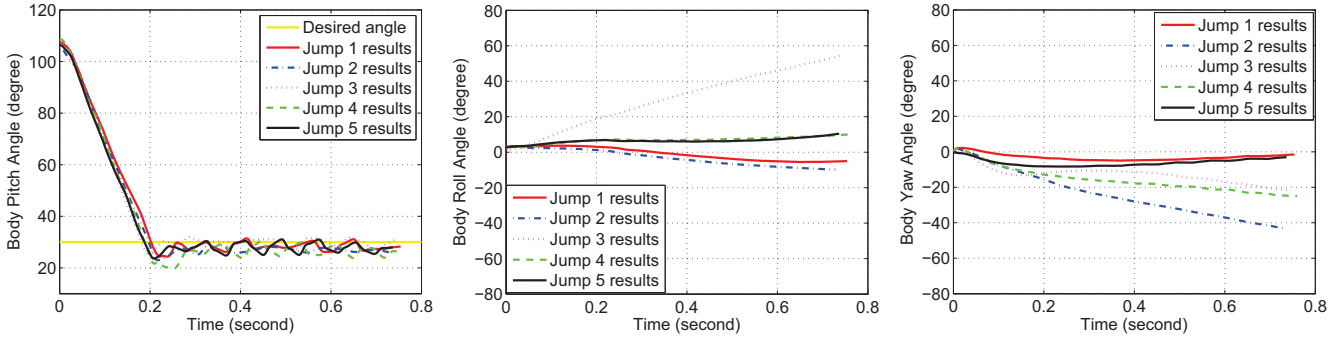
3) Aerial Maneuvering with the Sliding Mode Controller: Aerial maneuvering with the sliding mode controller is conducted similar to the one using the PD controller. To simplify the parameter tuning process, we employ the simplified controller in Eq. (20) with a saturation function $\text{sat}(\cdot)$

$$u = k\,\text{sat}(\frac{a\bar{z}_1 + \bar{z}_2}{\mu}) \tag{23}$$

where μ is a positive constant. Similar to the PD controller, we tune the parameters to obtain good performances with the free-fall experiment, and the final parameters are chosen as $k = 255$, $\mu = 600$, and $a = 50$.

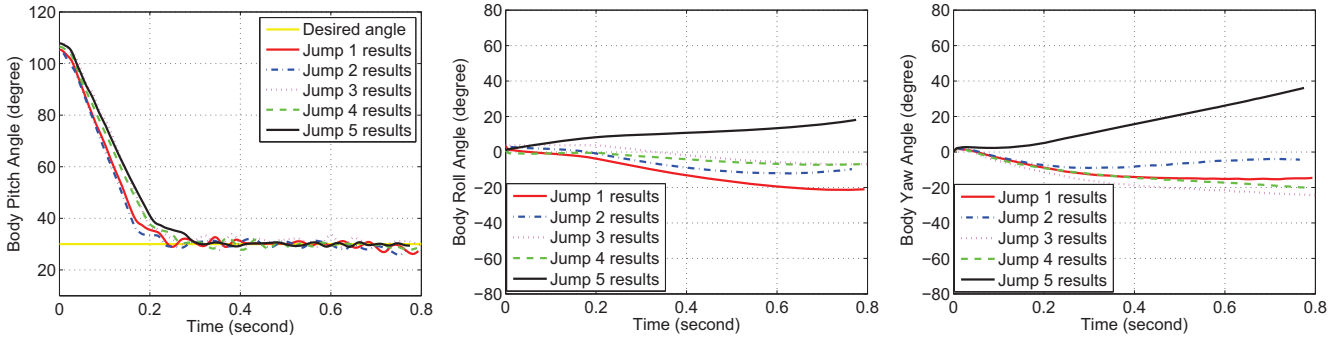
Five experiments are performed, and one (jump 4) of the robot's trajectory is shown in Fig. 6(c). As seen from the trajectory, the robot lands on the ground safely with one of its two large sides. The three body angles for five experiments are plotted in Fig. 9. From Fig. 9(a), the system has a rising time about 0.3s, and the pitch angle θ_b oscillates around 30° thereafter. Compared with results using the PD controller in Fig. 8, although the rising time is larger than the PD controller, the sliding mode controller has a smaller overshoot and steady state error. Furthermore, the oscillation after 0.3s is also smaller.

The robot also rotates in roll and yaw as can be seen from Fig. 9(b) and (c). Similar to the reason for the PD controller,



(a) (b) (c)

Fig. 8. Experimental results for aerial maneuvering when the tail is controlled by the PD controller: (a) the body's pitch angle with respect to time; (b) the body's roll angle with respect to time; (c) the body's yaw angle with respect to time.



(a) (b) (c)

Fig. 9. Experimental results for aerial maneuvering when the tail is controlled by the sliding mode controller: (a) the body's pitch angle with respect to time; (b) the body's roll angle with respect to time; (c) the body's yaw angle with respect to time.

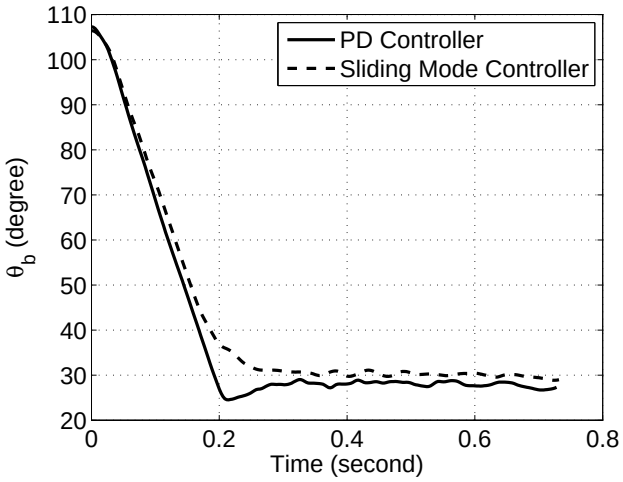


Fig. 10. Experimental results comparison for the PD controller and the sliding mode controller. The two curves are obtained by averaging the trajectories of five jumps for each controller.

some of the curves for roll and yaw angle are nonlinear. Nevertheless, these rotations do not significantly affect the landing posture of the robot.

4) Comparison of the Two Controllers: To compare the results of the two controllers, we average the results of the five jumps for each controller and plot the averaged result in Fig. 10. The data for important metrics are also listed in Table III. For the rising time, the PD controller reaches the desired angle faster, which is the same to the simulation results shown in Table II. The settling time for both controllers, using the same definition for the simulation, is almost the same. However, different from the simulation results, the PD controller has a larger steady state error, which might be due to the discrete implementation of the controller. The percent of overshoot is similar to the simulation results with the sliding mode controller having a smaller overshoot.

TABLE III
EXPERIMENTAL RESULTS COMPARISON FOR THE TWO CONTROLLERS

Metric	PD controller	Sliding mode controller
Rising time (s)	0.19	0.39
Settling time (s)	0.27	0.24
Steady state error (degree)	2.00	0.22
Percent of overshoot (%)	18.25	4.13

C. Tail Assisted Mode Transition

Besides aerial maneuvering, the tail is also designed for transition between wheeling and jumping during the locomotion. But different from aerial maneuvering, the control of mode transitions is achieved by wireless open loop control. Various commands can be sent from a desktop computer wirelessly to the embedded system, which actuates the tail

motor to rotate the tail in different directions with specified speeds. Meanwhile, we visually check the robot's status to make sure it can finish the mode transition successfully.

As explained in the tail design part, the robot can utilize its tail to stand up for jumping from the wheeling mode, and to lie down for wheeling from the jumping mode. Experimental results for these two cases are elaborated in the following.

The first case is to let the robot stand up for jumping when the robot lies on the ground with its side. One experiment for this case is shown in Fig. 11, where four frames from a self-righting video are extracted with time labeled on the top right corner. In Fig. 11, the robot is first in the wheeling mode with its wheels on the ground. Then, the robot's tail rotates towards the ground to make the robot's wheels away from the ground. After that, the robot starts to store energy and lower its center of mass in order to obtain a stabilizing standing position later. At 12s, the robot is fully charged for jumping and the tail rotates to push the ground such that the robot can complete the transition to a standing position and be ready for taking off. Note that Fig. 11 only shows the self-righting from the side with wheels. If the robot lands on the ground with the other side, the tail can rotate in the opposite direction to perform self-righting as well.

The second case is transforming from the jumping mode to the wheeling mode. Fig. 12 shows one experimental result of such a transformation. Four frames are extracted from a video that records the process. As shown in the figure, the lying down process is successfully achieved by swinging the tail forward to push the ground, and the robot transforms to the wheeling mode.

Tail assisted mode transition enables the robot to perform multi-modal locomotion. The entire experiments for modal transition can be found in the accompanied video submission. The wheeling and turning experimental results can be found in the video as well.

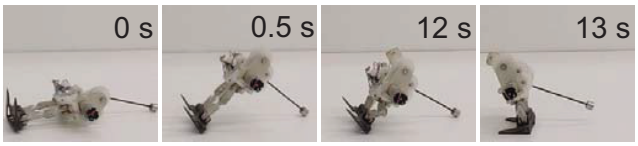


Fig. 11. Experimental results for transition from wheeling mode to jumping mode.

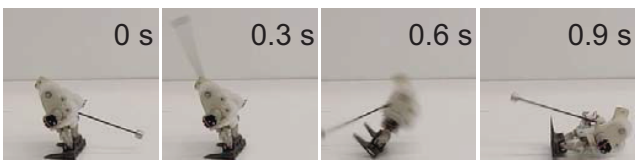


Fig. 12. Experimental results for transition from jumping mode to wheeling mode.

VI. Conclusion

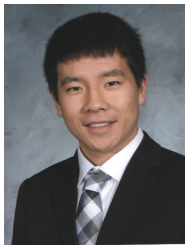
In this paper, we have presented the design, analysis, and experimentation of a miniature tailed jumping robot (MSU tailbot) that uses an active tail to maneuver in mid-air for safe landings. Additionally, the tailbot can wheel on the ground when no obstacle exists. The mechanical and electrical designs of the tailbot are detailed. The dynamics model for the tailbot in mid-air is established. Based on the dynamics model, the tail is optimized, and a PD controller and a sliding mode controller are designed. Both simulation and experimental results demonstrate successful aerial maneuvering using the two controllers. From the results, the two controllers have comparable performances with the sliding mode controller having a smaller steady state error in experiments. To the best of our knowledge, the tailbot is the first centimeter scale robot that has all the three capabilities: jumping, wheeling, and aerial maneuvering. With its small size and multi-mode locomotion ability, the tailbot can perform energy efficient locomotion in environments with obstacles, which has many applications such as mobile sensor networks, military surveillance, and environmental monitoring.

References

[1] R. M. Alexander, *Principles of Animal Locomotion*. Princeton University Press, 2003.
[2] M. Burrows, "Biomechanics: froghopper insects leap to new heights," *Nature*, vol. 424, no. 6948, pp. 509–509, 2003.
[3] J. Zhao, R. Yang, N. Xi, B. Gao, X. Fan, M. W. Mutka, and L. Xiao, "Development of a self-stabilization miniature jumping robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, 2009, pp. 2217–2222.
[4] J. Zhao, N. Xi, B. Gao, M. W. Mutka, and L. Xiao, "Design and testing of a controllable miniature jumping robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Taipei, Taiwan, 2010, pp. 3346–3351.
[5] ——, "Development of a controllable and continuous jumping robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, 2011, pp. 4614–4619.
[6] J. Zhao, J. Xu, B. Gao, N. Xi, F. J. Cintron, M. W. Mutka, and L. Xiao, "MSU jumper: A single-motor-actuated miniature steerable jumping robot," *IEEE Trans. Robotics*, vol. 29, no. 3, pp. 602–614, 2013.
[7] F. Cintrón, K. Pongaliur, M. W. Mutka, L. Xiao, J. Zhao, and N. Xi, "Leveraging height in a jumping sensor network to extend network coverage," *IEEE Trans. on Wireless Communications*, vol. 11, no. 5, pp. 1840–1849, 2012.
[8] T. Libby, T. Moore, E. Chang-Siu, D. Li, D. Cohen, A. Jusufi, and R. Full, "Tail-assisted pitch control in lizards, robots and dinosaurs," *Nature*, vol. 481, no. 7380, pp. 181–184, 2012.
[9] Y. Pei, F. Cintrón, M. Mutka, J. Zhao, and N. Xi, "Hopping sensor relocation in rugged terrains," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, 2009, pp. 3856–3861.
[10] J. Burdick and P. Fiorini, "Minimalist jumping robots for celestial exploration," *Int. J. Robot. Res.*, vol. 22, no. 7, pp. 653–674, 2003.
[11] M. Kovac, M. Fuchs, A. Guignard, J. Zufferey, and D. Floreano, "A miniature 7g jumping robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, USA, 2008, pp. 373–378.
[12] M. Kovac, M. Schlegel, J. Zufferey, and D. Floreano, "A miniature jumping robot with self-recovery capabilities," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, 2009, pp. 583–588.
[13] ——, "Steerable miniature jumping robot," *Auton. Robots*, vol. 28, no. 3, pp. 295–306, 2010.
[14] F. Li, W. Liu, X. Fu, G. Bonsignori, U. Scarfogliero, C. Stefanini, and P. Dario, "Jumping like an insect: Design and dynamic optimization of a jumping mini robot based on bio-mimetic inspiration," *Mechatronics*, vol. 22, no. 2, pp. 167–176, 2012.
[15] J. Zhang, G. Song, Y. Li, G. Qiao, A. Song, and A. Wang, "A bio-inspired jumping robot: Modeling, simulation, design, and experimental results," *Mechatronics*, vol. 23, no. 8, pp. 1123–1140, 2013.
[16] R. Armour, K. Paskins, A. Bowyer, J. Vincent, and W. Megill, "Jumping robots: a biomimetic solution to locomotion across rough terrain," *Bioinsp. Biomim.*, vol. 2, no. 3, pp. 65–82, 2007.
[17] M. Noh, S.-W. Kim, S. An, J.-S. Koh, and K.-J. Cho, "Flea-inspired catapult mechanism for miniature jumping robots," *IEEE Trans. Robotics*, vol. 28, no. 5, pp. 1007–1018, 2012.

[18] S. Stoeter and N. Papanikolopoulos, "Autonomous stair-climbing with miniature jumping robots," *IEEE Trans. Systems, Man, and Cybernetics-Part B*, vol. 35, no. 2, pp. 313–325, 2005.
[19] B. G. A. Lambrecht, A. D. Horchler, and R. D. Quinn, "A small, insect inspired robot that runs and jumps," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 1240–1245.
[20] E. Watari, H. Tsukagoshi, A. Kitagawa, and T. Tanaka, "A higher casting and jump motions realized by robots using magnetic brake cylinder," *ASME Journal of mechanisms and robotics*, vol. 3, no. 4, 2011.
[21] D. H. Kim, J. H. Lee, I. Kim, S. H. Noh, and S. K. Oho, "Mechanism, control, and visual management of a jumping robot," *Mechatronics*, vol. 18, no. 10, pp. 591–600, 2008.
[22] K. Kikuchi, K. Sakaguchi, T. Sudo, N. Bushida, Y. Chiba, and Y. Asai, "A study on a wheel-based stair-climbing robot with a hopping mechanism," *Mechanical Systems and Signal Processing*, vol. 22, no. 6, pp. 1316–1326, 2008.
[23] E. Ackerman, "Boston dynamics sand flea robot demonstrates astonishing jumping skills," IEEE spectrum Robotics Blog, 2012.
[24] A. Jusufi, D. Goldman, S. Revzen, and R. Full, "Active tails enhance arboreal acrobatics in geckos," *Proceedings of the National Academy of Sciences*, vol. 105, no. 11, pp. 4215–4219, 2008.
[25] E. Chang-Siu, T. Libby, M. Tomizuka, and R. J. Full, "A lizard-inspired active tail enables rapid maneuvers and dynamic stabilization in a terrestrial robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, 2011, pp. 1887–1894.
[26] A. Johnson, T. Libby, E. Chang-Siu, M. Tomizuka, R. Full, and D. Koditschek, "Tail assisted dynamic self righting," in *Proceedings of the International Conference on Climbing and Walking Robots*, Baltimore, Maryland, USA, 2012, pp. 611–620.
[27] A. Demir, M. M. Ankarali, J. P. Dyhr, K. A. Morgansen, T. L. Daniel, and N. J. Cowan, "Inertial redirection of thrust forces for flight stabilization," in *Proceedings of the International Conference on Climbing and Walking Robots*, Baltimore, Maryland, USA, 2012, pp. 239–245.
[28] R. Briggs, J. Lee, M. Haberland, and S. Kim, "Tails in biomimetic design: Analysis, simulation, and experiment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Algarve, Portugal, 2012, pp. 1473–1480.
[29] N. Kohut, A. Pullin, D. Haldane, D. Zarrouk, and R. Fearing, "Precise dynamic turning of a 10 cm legged robot on a low friction surface using a tail," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, 2013, pp. 3299–3306.
[30] C. Casarez, I. Penskiy, and S. Bergbreiter, "Using an inertial tail for rapid turns on a miniature legged robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, 2013, pp. 5469–5474.
[31] J. Zhao, T. Zhao, N. Xi, F. J. Cintrón, M. W. Mutka, and L. Xiao, "Controlling aerial maneuvering of a miniature jumping robot using its tail," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, 2013, pp. 3802–3807.
[32] J. Zhao, N. Xi, F. J. Cintron, M. W. Mutka, and L. Xiao, "A single motor actuated miniature steerable jumping robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Algarve, Portugal, 2012, pp. 4274–4275.
[33] T. Mather and M. Yim, "Modular configuration design for a controlled fall," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, 2009, pp. 5905–5910.
[34] E. Yang, P. Chao, and C. Sung, "Landing posture control for a generalized twin-body system using methods of input–output linearization and computed torque," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 3, pp. 326–336, 2009.
[35] ——, "Optimal control of an under-actuated system for landing with desired postures," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 2, pp. 248–255, 2011.
[36] S. Agrawal and C. Zhang, "An approach to posture control of free-falling twin bodies using differential flatness," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 685–690.
[37] E. Chang-Siu, T. Libby, M. Brown, R. J. Full, and M. Tomizuka, "A nonlinear feedback controller for aerial self-righting by a tailed robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, 2013, pp. 32–39.
[38] Z. Li and R. Montgomery, "Dynamics and optimal control of a legged robot in flight phase," in *Proc. IEEE Int. Conf. Robot. Autom.*, Cincinnati, OH, USA, 1990, pp. 1816–1821.
[39] H. K. Khalil, *Nonlinear systems*, 3rd ed. Prentice Hall, 2002.

Jianguo Zhao received the B.E. degree in Mechanical Engineering from Harbin Institute of Technology, Harbin, China and his M.E. degree in Mechatronic Engineering from Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. He is currently pursuing the PhD degree in the Robotics and Automation Lab at Michigan State University. His research interests include bio-inspired robotics, dynamics and control, and vision based control. He is a member of the American Society of Mechanical Engineers (ASME).



Tianyu Zhao received the B.S. degree in mechanical engineering in 2013 from Michigan State University, East Lansing, MI, USA. He is currently working towards the M.S. degree in mechanical engineering at Carnegie Mellon University, Pittsburgh, PA, USA. His areas of interests are mechanical design, bio-inspired robotics, and mechatronic product design.

Ning Xi received his D.Sc. degree in Systems Science and Mathematics from Washington University in St. Louis in 1993, and B.S. degree in electrical engineering from Beijing University of Aeronautics and Astronautics. He is the University Distinguished Professor and John D. Ryder Professor of Electrical and Computer Engineering at Michigan State University. He is a fellow of IEEE. His research interests include robotics, manufacturing automation, micro/nano manufacturing, nano sensors and devices, and intelligent control and systems.

Matt W. Mutka received the B.S. degree in electrical engineering from the University of Missouri-Rolla, the M.S. degree in electrical engineering from Stanford University, and the Ph.D. degree in Computer Sciences from the University of Wisconsin-Madison. He is on the faculty of the Department of Computer Science and Engineering at Michigan State University, where he is currently professor and chairperson. He has been a visiting scholar at the University of Helsinki, Helsinki, Finland, and a member of technical staff at Bell Laboratories in Denver, Colorado. He is an IEEE Fellow. His current research interests include mobile computing, sensor networking, wireless networking and multimedia networking.

Li Xiao received the BS and MS degrees in computer science from Northwestern Polytechnic University, China, and the PhD degree in computer science from the College of William and Mary. She is an associate professor of computer science and engineering at Michigan State University. Her research interests are in the areas of distributed and networking systems, overlay systems and applications, and wireless networks. She is a senior member of the IEEE.