Nonlinear Dyn
https://doi.org/10.1007/s11071-023-08646-w

ORIGINAL PAPER

How a serpentine tail assists agile motions of kangaroo rats: a dynamics and control approach

Yujiong Liu · Pinhas Ben-Tzvi

Received: 1 February 2022 / Accepted: 4 June 2023

Abstract Kangaroo rat is a good representative for general bipedalism with a serpentine tail. Modeling and analyzing the kangaroo rat motion helps to understand the serpentine tail functionalities in agile motions of bipedal mobile platforms, and this understanding is expected to lay the foundation for the future development of such robotic systems. This paper analyzes the kangaroo rat motions through dynamic modeling and control. The system dynamic model is established using the inertia matrix method, and two typical serpentine tail models are considered: a continuum tail model where the tail is modeled as several constant curvature arcs, and an articulated tail model where the tail is discretized into rigid links. Regularized contact model is used to compute the ground reaction force (GRF). To automatically plan the tail motion, numerical optimal control techniques (i.e., direct collocation method) are utilized. Partial feedback linearization is then used to track the designed tail trajectory. Based on the formulated dynamic model and motion controller, two representative tail functions (airborne righting and supporting) were simulated and analyzed. The results validated the proposed modeling and control framework and showed the nontrivial functionalities of the serpentine tail in helping the kangaroo rat to achieve agile motions. Moreover, comparative studies on the two tail models and the tail segmentations were performed to analyze the model differences. The results demonstrated that the articulated tail model is a good approximation of the continuum tail model, and more tail segments and links enhance the kangaroo rat's ability to deliberately adjust its motion.

Keywords Serpentine tail · Kangaroo rat · Dynamic modeling · Dynamic analysis · Optimal control

1 Introduction

In recent years, inspired by the fascinating tail motions in the animal kingdom [1–4], roboticists have considered augmenting mobile robots with a robotic tail, to help the robot achieve enhanced locomotion performance. These trials include both using pendulum robotic tails [5–20] and serpentine robotic tails [21–27]. Due to its limited shape configuration, pendulum tails are usually designed for a specific purpose, such as performing airborne righting tasks

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11071-023-08646-w.

Y. Liu · P. Ben-Tzvi (✉)
Robotics and Mechatronics Lab, Mechanical Engineering Department, Virginia Tech, Blacksburg,
VA 24060, USA
e-mail: bentzvi@vt.edu

Y. Liu
e-mail: yjliu@vt.edu

Published online: 04 July 2023

[6–9], maneuvering [11, 12], and improving vehicle acceleration performance [13]. In comparison, serpentine robotic tails usually have more degrees of freedom (DOF) and thus have the benefits of being dexterous and versatile. For instance, the serpentine tail can usually carry out a range of tasks from maneuvering to stabilization [28] using the same tail. However, the tradeoff is that this kind of tail usually requires a heavier actuation unit and a more complex mechanical structure.

From a mechanics perspective, the tail functionalities could be summarized into two categories: static and dynamic functions. The static functions of the tail mean that the tail is only used to inject static forces into the system. The system motion is governed by statics, and only position information is used. Such functions include adjusting the system center of mass (COM), supporting the body, grasping, manipulation, etc. The dynamic functions of the tail mean that the tail is used to inject inertial forces into the system. The system motion is governed by the full-level dynamics, and kinematic information including acceleration is necessary. Such functions include airborne righting, maneuvering, stabilization, etc.

Kangaroo rat is an animal that relies heavily on its tail to assist its agile motions. For instance, Fig. 1 shows that a kangaroo rat flips over its body in the air using its tail after being attacked by a rattlesnake [29]. Further observations imply that kangaroo rats may use most of the functionalities mentioned above, that is, using the tail as an additional appendage (e.g., leg) to support its locomotion on the ground, using the tail in the air to change its torso orientation, and using the tail to adjust the ground reaction force (GRF) during ground contact events (indirectly affecting the locomotion). Therefore, the kangaroo rat is thought to be a great benchmark model to investigate the functions of a tail for bipedal locomotion. This idea recently started attracting attention and a series of research efforts were carried out, from both the biological side [30, 31] and the robotic side [10, 32, 33]. However, the existing efforts on the biological side mainly focus on data interpretation using statistical analysis, and the efforts on the robotic side mainly focus on using the pendulum tail abstraction. Both approaches use a simplified dynamic model, which is either in 2D or using a single-segment single-link tail model. This brings up an important missing part for this research—a high-fidelity 3D dynamic model that captures the serpentine nature of the tail.

Therefore, in this paper, we aim to investigate how the kangaroo rat uses its serpentine tail to achieve agile motions from a dynamics and control perspective, that is, establishing a 3D dynamic model of the kangaroo rat with a serpentine tail and analyzing the tail's functionalities with helping the kangaroo rat to achieve agile motions. This investigation is expected to provide a solid understanding (mechanically and quantitatively) of the kangaroo rat motions and what role the serpentine tail plays in achieving these motions. These understandings could further benefit

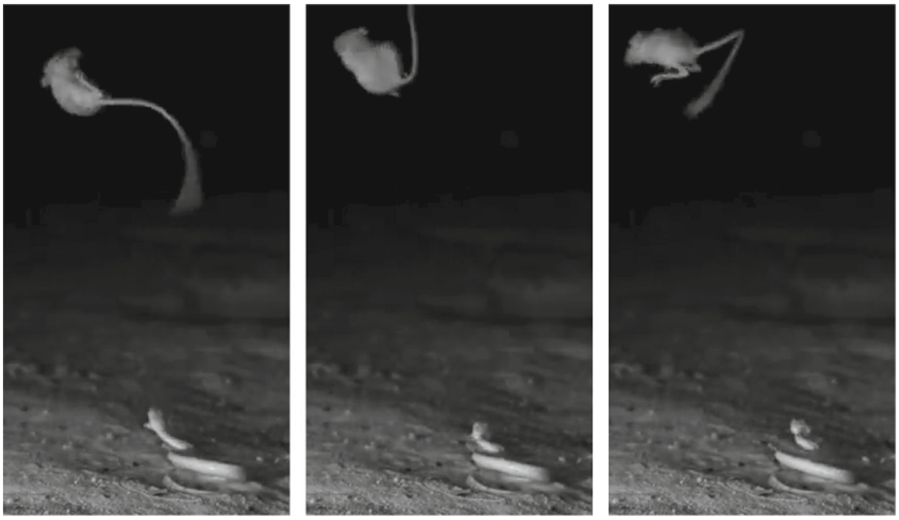

Fig. 1 A kangaroo rat flips itself in the air by swinging its serpentine tail. Snapshots are captured from the supplementary video 1 in [30], which is also available online at [29]

the development and control of such robotic systems, for instance, developing novel tailed biped robots for executing highly agile motions (military technology) or developing tail-assisted fast maneuvering techniques for space robots (space technology). In control theory, solving the unique kangaroo rat tail-assisted locomotion problem could further enrich the existing whole-body control theory.

The major contributions of this work are summarized as follows. Firstly, a general 3D bipedal model with multi-contact feet and a general serpentine tail is established. Secondly, two representative and general serpentine tail structures, namely the articulated tail model and the continuum tail model, are formulated. Thirdly, numerical optimal control methods (direct collocation) are used to automatically plan the tail motion, which allows incorporating advanced input and state constraints. Fourthly, through the established dynamic model and controller, agile motions, tail functionalities, and tail model comparative studies of the kangaroo rat are analyzed. Each of these four points is proposed in the literature for the first time. The combination of these four points provides a high-fidelity kangaroo rat model and explains the serpentine tail's functionalities with assisting the kangaroo rat to achieve agile motions from a mechanics perspective for the first time.

The overview of the modeling and control approaches in this work is shown in Fig. 2 where the entire process consists of three parts: dynamic modeling, motion planning, and simulation. The dynamic modeling part establishes the equation of motion (EOM) of a kangaroo rat model considering two types of serpentine tail models and the ground contact events. This model is used in both the motion planning part and the simulation part. The difference is that in motion planning, the dynamic model is used as path constraints for the tail trajectory optimization tasks while in simulation, the dynamic model is numerically solved to verify the motion planning effectiveness. To convert the designed trajectory to actual joint inputs, a feedback-based trajectory tracking controller was implemented in the simulation.

The following sections of this paper are organized as follows. Section 2 formulates the dynamic model of the kangaroo rat. Section 3 derives the motion planning algorithm and the trajectory tracking controller for the serpentine tail motion control problem. Section 4 implements the numerical experiments and explores the serpentine tail functionalities in kangaroo rat locomotion. Section 5 summarizes the main results of this paper.

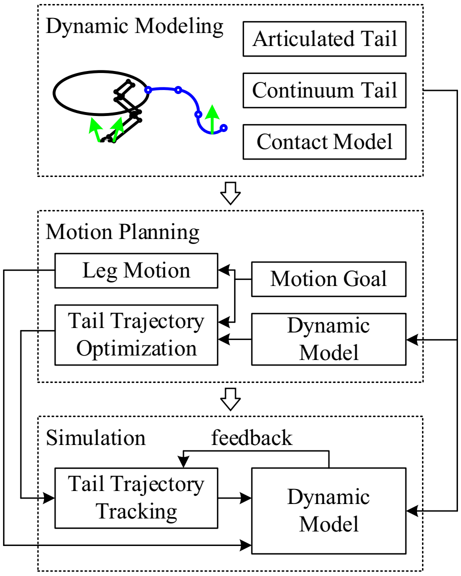

Fig. 2 An overview of the modeling and control approaches in this work

2 Dynamic modelling

The dynamic model of the kangaroo rat is shown in Fig. 3 where the body is abstracted into an ellipsoid

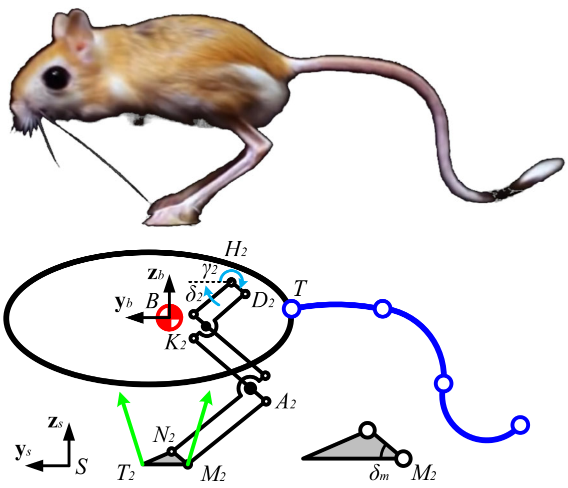

Fig. 3 Kangaroo rat kinematic model. Note that the animal on the top is for morphological reference only. It is a jerboa—a different species with similar morphology and behaviors

and the leg is modeled as a planar three-parallelogram mechanism. The foot is modeled with a flat sole such that it allows stable standing. The inertial frame $\Sigma S := (S, \mathbf{x}_s, \mathbf{y}_s, \mathbf{z}_s)$ is attached on the ground with $\mathbf{z}_s$ pointing vertically up and $\mathbf{y}_s$ pointing forward. The body-fixed frame of the torso $\Sigma B := (B, \mathbf{x}_b, \mathbf{y}_b, \mathbf{z}_b)$ is attached at the torso COM with its basis vector aligning with the principal axes of the torso moment of inertia (MOI). Note that the torso COM is not necessarily located at the ellipsoid center or its focus, but the MOI is always computed with respect to the ellipsoid center.

The EOM is established using the inertia matrix method in [34], as shown in Eq. (1). The leg inertial loading is neglected due to its small influence (lightweight legs) on the overall motion. However, its kinematics is still needed to generate the GRF.

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{J}_{ta}\boldsymbol{\tau}_{ta} + \mathbf{J}_f\mathbf{f}_c \tag{1}$$

where $\mathbf{H}$ is the system inertia matrix, $\mathbf{C}$ is the bias force containing all the gravitational, centrifugal, and Coriolis forces, $\boldsymbol{\tau}_{ta}$ is the actuation input from the tail, $\mathbf{f}_c$ is the GRF, $\mathbf{J}_{ta}$ and $\mathbf{J}_f$ are the corresponding Jacobians for the actuation and GRF, respectively, $\mathbf{q} = [\mathbf{p}_b^T\ \boldsymbol{\phi}_b^T\ \mathbf{q}_t^T]^T$ is the system generalized coordinates where $\mathbf{p}_b$ is the position of point B, $\boldsymbol{\phi}_b = [\phi_x\ \phi_y\ \phi_z]^T$ is the torso orientation, and $\mathbf{q}_t$ is the generalized coordinates that describe the tail configuration.

The serpentine tail consists of m serially connected segments. Each segment can bend spatially (i.e., bending to left and right or bending up and down), and is fully defined by these two mobilities. Therefore, the tail has $2m$ generalized coordinates and the system has $d = 6 + 2m$ DOF. The components in Eq. (1) are obtained as follows:

$$\mathbf{H} = m_b\mathbf{J}_{b,x}^T\mathbf{J}_{b,x} + \mathbf{J}_{b,\omega}^T\mathbf{I}_b\mathbf{J}_{b,\omega} + \sum\nolimits_{i=1}^{m}\mathbf{H}_i \tag{2}$$

$$\mathbf{C} = m_b\mathbf{J}_{b,x}^T\mathbf{g} + \mathbf{J}_{b,\omega}^T\widetilde{\boldsymbol{\omega}}_b\mathbf{I}_b\boldsymbol{\omega}_b + \mathrm{ID}(\mathrm{tail}, \mathbf{q}, \dot{\mathbf{q}}, 0) \tag{3}$$

where m_b, $\mathbf{I}_b$, $\mathbf{H}_i$ are the torso mass, torso MOI (measured in the inertial frame), and the inertia matrix for the i-th segment of the tail, respectively. $\mathbf{J}_{b,x}$ and $\mathbf{J}_{b,\omega}$ are the trivial Jacobians for $\mathbf{p}_b$ and $\boldsymbol{\phi}_b$, respectively. $\mathbf{g} = [0\ 0\ g]^T$ is the negative gravitational acceleration. $\boldsymbol{\omega}_b = \dot{\boldsymbol{\phi}}_b$ is the angular velocity of the torso. The tilde above a vector indicates its skew-symmetric expansion. The "ID(tail)" function represents the inverse dynamics of the tail subsystem that contains all the non-actuation loading (inertial, gravitational, frictional, etc.):

$$\mathrm{ID}(\mathrm{tail}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \sum\nolimits_{i=1}^{m}\boldsymbol{\tau}_i \tag{4}$$

where $\boldsymbol{\tau}_i$ is the inertial loadings for the i-th tail segment.

The critical kinematic and dynamic information required by Eqs. (1)–(4) is presented in the following subsections in detail. To compare different serpentine tail structures, a continuum tail model and a discrete tail model are formulated.

2.1 Leg kinematics

For simplicity, the leg mechanism is set to move only in the plane that is parallel to the $\mathbf{y}_b B \mathbf{z}_b$ plane, i.e., there is no abduction DOF in the leg. Due to the three-parallelogram mechanism, all the hip (H), knee (K), ankle (A), metatarsophalangeal (M), and toe (T) positions in the leg plane could be described by two hip angles γ_l and δ_l, where the subscript $l = \{1, 2\}$ indicates the right (1) or the left (2) leg, respectively. Therefore, the metatarsophalangeal and toe positions are calculated using Eqs. (5)–(9), where the left superscript is used to indicate the measuring frame other than the inertial frame. Other subscripts are labeled to indicate a point or a line segment. For instance, $\mathbf{p}_{m,l}$ stands for the position of point M_l, $\mathbf{p}_{h2m,l}$ means the vector from H_l to M_l, and L_{h2d} is the linkage length from H_l to D_l.

$$\mathbf{p}_{m,l} = \mathbf{p}_b + \mathbf{p}_{b2h,l} + \mathbf{p}_{h2m,l} \tag{5}$$

$$\mathbf{p}_{t,l} = \mathbf{p}_{h2m,l} + \mathbf{p}_{m2t,l} \tag{6}$$

$${}^{b}\mathbf{p}_{h2m,l} = (L_{h2d} + L_{k2a})\boldsymbol{u}_h(\gamma_l) + (L_{d2k} + L_{a2m})\boldsymbol{u}_h(\delta_l) \tag{7}$$

$${}^{b}\mathbf{p}_{m2t,l} = -L_{m2t}\boldsymbol{u}_h(\gamma_l - \delta_l) \tag{8}$$

$$\boldsymbol{u}_h(x) = [0\ \cos(x)\ \sin(x)]^T \tag{9}$$

The velocities could be calculated by differentiating the above equations directly and the corresponding Jacobians are obtained by factoring out the generalized velocity $\dot{\mathbf{q}}$. Then, the generalized GRF is obtained as

$$\mathbf{J}_f\mathbf{f}_c = \sum\nolimits_{l=1}^{2}\left(\mathbf{J}_{m,l}\mathbf{f}_{m,l} + \mathbf{J}_{t,l}\mathbf{f}_{t,l}\right) \tag{10}$$

2.2 Articulated tail model

The articulated tail assumes that each tail segment consists of n serially connected links and adjacent links are connected by universal joints. All joints have the same rotation in the same segment. To compute the kinematics of this tail, a similar approach as in [35] is used. As shown in Fig. 4, the body-fixed frames are defined as follows. Frame $\Sigma J_j := (J_j, \mathbf{x}_j, \mathbf{y}_j, \mathbf{z}_j)$ is attached to the j-th link and is located at the j-th joint center J_j. $\mathbf{y}_j$ points along with the link and to the tail root direction. $\mathbf{z}_j$ aligns with the yaw axis of the j-th universal joint. Note that the first joint coincides with the tail mounting point T.

With the above definitions, the orientation of each link ($\mathbf{R}_j$) could be obtained recursively, as shown in Eq. (11) and (12) where $\mathbf{R}_x$ and $\mathbf{R}_z$ are the principal rotation matrix functions with respect to the x-axis and z-axis, respectively. α_i and β_i are the pitch and yaw rotation angles for the universal joints in the i-th segment, respectively. Therefore, the tail generalized coordinates for the articulated tail case is $\mathbf{q}_t = [\alpha_1\ \beta_1 \ldots \alpha_m\ \beta_m]^T$.

$${}^{j-1}\mathbf{R}_j = \mathbf{R}_x(\alpha_i)\mathbf{R}_z(\beta_i),\ (i-1)n+1 \le j \le in \tag{11}$$

$$\mathbf{R}_j = \begin{cases} \mathbf{R}_b, & j = 0 \\ \mathbf{R}_{j-1} \cdot {}^{j-1}\mathbf{R}_j, & j > 0 \end{cases} \tag{12}$$

The joint positions $\mathbf{p}_{j,jnt}$ and the COM position $\mathbf{p}_{j,com}$ of each link could then be calculated recursively too, as shown in Eqs. (13)–(15), where L_{j2c} and L_{j2j} are the joint-to-COM and joint-to-joint distance, respectively.

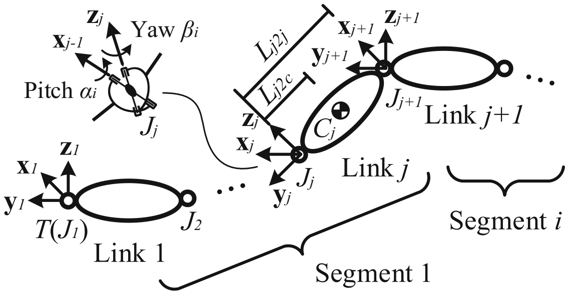

Fig. 4 Articulated tail model

$$\mathbf{p}_{j,com} = \mathbf{p}_{j,jnt} + \mathbf{p}_{j,j2c} \tag{13}$$

$$\mathbf{p}_{j,jnt} = \begin{cases} \mathbf{p}_b + \mathbf{p}_{b2t}, & j = 1 \\ \mathbf{p}_{j-1,jnt} + \mathbf{p}_{j-1,j2j}, & j > 1 \end{cases} \tag{14}$$

$$\begin{cases} \mathbf{p}_{j,j2c} = -L_{j2c}\mathbf{y}_j \\ \mathbf{p}_{j,j2j} = -L_{j2j}\mathbf{y}_j \end{cases} \tag{15}$$

The corresponding angular velocities, linear velocities, Jacobians, and accelerations are calculated by differentiating the position relationships directly, as outlined in Appendix A. Therefore, with the above kinematic information, the segment inertia and non-actuation loading could be computed as

$$\mathbf{H}_i = \sum_{j=(i-1)n+1}^{i\cdot n}\left(m_{at}\mathbf{J}_{j,com}^T\mathbf{J}_{j,com} + \mathbf{J}_{j,\omega}^T\mathbf{I}_{j,at}\mathbf{J}_{j,\omega}\right) \tag{16}$$

$$\boldsymbol{\tau}_i = \sum_{j=(i-1)n+1}^{i\cdot n}\left[\mathbf{J}_{j,com}^T m_{at}\left(\dot{\mathbf{v}}_{j,com} + \mathbf{g}\right) + \mathbf{J}_{j,\omega}^T\left(\mathbf{I}_{j,at}\dot{\boldsymbol{\omega}}_j + \widetilde{\boldsymbol{\omega}}_j\mathbf{I}_{j,at}\boldsymbol{\omega}_j\right)\right] \tag{17}$$

where m_{at} and $\mathbf{I}_{j,at}$ are the mass and the MOI (measured in the inertial frame) for the j-th link, respectively. Note that Eq. (17) only accounts for the inertial and gravitational loadings. The internal joint frictions are neglected to simplify the modeling. The reasoning for this is that previous modeling and simulations [35, 38] showed that the internal joint frictions mainly affect the control effort (requiring larger motor torques) and do not change the overall motion significantly. Since the focus of this paper is on the overall motions instead of the joint torques, the joint frictions were neglected for simplicity.

2.3 Continuum tail model

The continuum tail model assumes that each tail segment is a non-extensible constant curvature rod and there is no twist motion inside the rod. Mass is uniformly distributed along the rod. Due to the results in [36] that the angular kinetic energy constitutes only 5% of the total kinetic energy, the angular inertial loading for an infinitesimal slice (referring to Fig. 5) is neglected, i.e., the mass is assumed to concentrate on the neutral line. Similar to the articulated tail model, all frictions and elasticities are also neglected. Only the inertial loading and the gravitational loading are computed. Based on these assumptions, the shape of

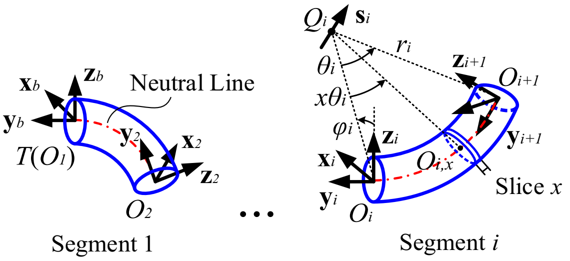

Fig. 5 Continuum tail model

each tail segment is totally defined by two local variables: the out-of-plane angle φ_i and the in-plane angle θ_i. And the tail generalized coordinates are $\mathbf{q}_t = [\varphi_1\ \theta_1\ \ldots\ \varphi_m\ \theta_m]^T$. Figure 5 illustrates the kinematic configuration of the continuum tail model. Note that since the tail is not extensible, $\theta_i r_i = L_{ct}$ where the L_{ct} is the segment length.

There are several existing methods to formulate the continuum robot dynamics, such as the classic [37] or the new [36] modal approaches. The main differences between these methods are their kinematic representations, i.e., how to describe the position and orientation of the slice x. We apply the same approach as in [38] due to its advantage to represent the kinematic integral explicitly. However, since the continuum tail model is non-extensible and non-elastic, the formulation is slightly modified. That is, the inertial matrix and the non-actuation loading of the i-th segment are computed as integrals along the neutral line:

$$\mathbf{H}_i = m_{ct}\int_0^1 \mathbf{J}_{i,o,x}^T \mathbf{J}_{i,o,x} dx \tag{18}$$

$$\boldsymbol{\tau}_i = m_{ct}\int_0^1 \mathbf{J}_{i,o,x}^T (\dot{\mathbf{v}}_{i,o,x} + \mathbf{g}) dx \tag{19}$$

where m_{ct} is the segment mass, $\mathbf{J}_{i,o,x}$ is the Jacobian of the slice center $O_{i,x}$, $\dot{\mathbf{v}}_{i,o,x}$ is the acceleration of $O_{i,x}$, and x is the local parametric variable. Note that Eq. (19) neglects the angular inertial loading due to its small contribution to the total inertial loading. This is equivalent to concentrating the mass on the neutral line.

With the constant curvature assumption, Eq. (18) and (19) turn out to be integrable, i.e., the integral variable x could be separated from the kinematic terms. Therefore, expressing the $O_{i,x}$ position as an explicit function of x yields

$$\mathbf{p}_{i,o,x} = \mathbf{p}_{i,q} + \mathrm{s}_x\mathbf{a}_i - \mathrm{c}_x\mathbf{b}_i \tag{20}$$

in which $\mathrm{s}_x = \sin\theta_i x$ and $\mathrm{c}_x = \cos\theta_i x$. $\mathbf{p}_{i,q}$, $\mathbf{a}_i$, and $\mathbf{b}_i$ are intermediate vectors defined by Eqs. (21)–(23).

$$\mathbf{p}_{i,q} = \mathbf{p}_{i,o} + \mathbf{b}_i \tag{21}$$

$$\mathbf{a}_i = -r_i\mathbf{y}_i \tag{22}$$

$$\mathbf{b}_i = r_i(\sin\varphi_i\mathbf{x}_i + \cos\varphi_i\mathbf{z}_i) \tag{23}$$

The velocity and Jacobian of $O_{i,x}$ are then obtained by directly differentiating Eq. (20):

$$\mathbf{v}_{i,o,x} = \dot{\mathbf{p}}_{i,q} + \mathrm{s}_x\dot{\mathbf{a}}_i + x\mathrm{c}_x\mathbf{a}_i\dot{\theta}_i - \mathrm{c}_x\dot{\mathbf{b}}_i + x\mathrm{s}_x\mathbf{b}_i\dot{\theta}_i \tag{24}$$

$$\begin{aligned}\mathbf{J}_{i,o,x}^T &= \mathbf{J}_{i,q}^T + \mathrm{s}_x\mathbf{J}_{i,a}^T + x\mathrm{c}_x\mathbf{J}_{i,\theta}^T\mathbf{a}_i^T - \mathrm{c}_x\mathbf{J}_{i,b}^T + x\mathrm{s}_x\mathbf{J}_{i,\theta}^T\mathbf{b}_i^T\\ &= \mathbf{J}_{i,v}^T \circ \boldsymbol{f}_{i,v}(x)\end{aligned} \tag{25}$$

$$\mathbf{J}_{i,v}^T = [\ \mathbf{J}_{i,q}^T \quad \mathbf{J}_{i,a}^T \quad \mathbf{J}_{i,\theta}^T\mathbf{a}_i^T \quad \mathbf{J}_{i,b}^T \quad \mathbf{J}_{i,\theta}^T\mathbf{b}_i^T\ ] \tag{26}$$

$$\boldsymbol{f}_{i,v}(x) = [\ 1 \quad \mathrm{s}_x \quad x\mathrm{c}_x \quad -\mathrm{c}_x \quad x\mathrm{s}_x\ ]^T \tag{27}$$

where $\mathbf{J}_{i,q}$, $\mathbf{J}_{i,a}$, $\mathbf{J}_{i,b}$, and $\mathbf{J}_{i,\theta}$ are the corresponding Jacobians for $\mathbf{p}_{i,q}$, $\mathbf{a}_i$, $\mathbf{b}_i$, and θ_i, respectively. In Eq. (25), the Jacobian is written as a block-wise matrix multiplication (denoted by "$\circ$") form such that the integral terms are separated from the non-integral terms. This form makes the expression concise and facilitates the following integrations. Therefore, the inertia matrix for the i-th segment could be evaluated as

$$\begin{aligned}\mathbf{H}_i &= m_{ct}\int_0^1 \mathbf{J}_{i,v}^T \circ \boldsymbol{f}_{i,v}(x)\boldsymbol{f}_{i,v}^T(x) \circ \mathbf{J}_{i,v} dx\\ &= m_{ct}\mathbf{J}_{i,v}^T \circ \mathbf{E}_{i,v} \circ \mathbf{J}_{i,v}\end{aligned} \tag{28}$$

where $\mathbf{E}_{i,v}(\theta_i) = \int_0^1 \boldsymbol{f}_{i,v}(x)\boldsymbol{f}_{i,v}^T(x)dx$ is a precomputed matrix that only depends on θ_i and its expression is presented in Appendix B. Similarly, the acceleration of $O_{i,x}$ is obtained by differentiating Eq. (24):

$$\begin{aligned}\dot{\mathbf{v}}_{i,o,x} = \ddot{\mathbf{p}}_{i,q} + \mathrm{s}_x\ddot{\mathbf{a}}_i + 2x\mathrm{c}_x\dot{\mathbf{a}}_i\dot{\theta}_i + x\mathrm{c}_x\mathbf{a}_i\ddot{\theta}_i - x^2\mathrm{s}_x\mathbf{a}_i\dot{\theta}_i^2 \\ - \mathrm{c}_x\ddot{\mathbf{b}}_i + 2x\mathrm{s}_x\dot{\mathbf{b}}_i\dot{\theta}_i + x\mathrm{s}_x\mathbf{b}_i\ddot{\theta}_i + x^2\mathrm{c}_x\mathbf{b}_i\dot{\theta}_i^2\end{aligned} \tag{29}$$

Adding up the gravitational acceleration and using the block-wise matrix multiplication notation, the non-actuation loading is computed as

$$\begin{aligned}\boldsymbol{\tau}_i &= m_{ct}\int_0^1 \mathbf{J}_{i,v}^T \circ \boldsymbol{f}_{i,v}(x)\boldsymbol{f}_{i,a}(x) \circ \mathbf{h}_{i,v}dx \\ &= m_{ct}\mathbf{J}_{i,v}^T \circ \mathbf{Q}_{i,v} \circ \mathbf{h}_{i,v}\end{aligned} \tag{30}$$

$$\mathbf{h}_{i,v}^T = [\ddot{\mathbf{p}}_{i,q}^T + \mathbf{g}^T \;\; \ddot{\mathbf{a}}_i^T \;\; \dot{\mathbf{a}}_i^T\dot{\theta}_i \;\; \mathbf{a}_i^T \;\; \ddot{\theta}_i \;\; \mathbf{a}_i^T\dot{\theta}_i^2 \;\; \ddot{\mathbf{b}}_i^T \dot{\mathbf{b}}_i^T\dot{\theta}_i \;\; \mathbf{b}_i^T\ddot{\theta}_i \;\; \mathbf{b}_i^T\dot{\theta}_i^2] \tag{31}$$

$$\boldsymbol{f}_{i,a}(x) = [1 \;\; \mathrm{s}_x \;\; 2x\mathrm{c}_x \;\; x\mathrm{c}_x \;\; -x^2\mathrm{s}_x \;\; -\mathrm{c}_x \;\; 2x\mathrm{s}_x \;\; x\mathrm{s}_x \;\; x^2\mathrm{c}_x] \tag{32}$$

where $\mathbf{Q}_{i,v}(\theta_i) = \int_0^1 \boldsymbol{f}_{i,v}(x)\boldsymbol{f}_{i,a}(x)dx$ is a precomputed matrix that is only dependent on θ_i. The detailed expression for this matrix is given in Appendix B.

The orientation and position of each segment frame $\sum O_i$ could be calculated recursively, as shown in Eq. (33) and (34). The higher-order information (velocities, Jacobians, and accelerations) could be calculated by directly differentiating their position relationships.

$$\mathbf{R}_{i+1} = \mathbf{R}_i\mathbf{R}_y(\varphi_i)\mathbf{R}_x(-\theta_i)\mathbf{R}_y(-\varphi_i) \tag{33}$$

$$\mathbf{p}_{i+1,o} = \mathbf{p}_{i,o} + \sin\theta_i\mathbf{a}_i + (1 - \cos\theta_i)\mathbf{b}_i \tag{34}$$

Moreover, since the above kinematic representation is essentially a variant of the Frenet-Serret frame [37], it is not able to handle the case when the curvature equals zero, i.e., the segment becomes a straight line. For such cases (e.g., when $\theta_i < 10^{-6}$rad), the segment dynamics are treated separately as a straight rigid link.

2.4 Ground contact model

The same ground contact model as in [39] is used, which is a regularized compliant contact model [40] that formulates the normal force as a nonlinear spring-damper system and the friction force as a linear spring-damper system. The friction force model obeys the Coulomb friction law, which states that the quotient of the friction force and the normal force cannot be greater than the friction coefficient. Therefore, for each contact point, the GRF is obtained as

$$\mathbf{f}_{grf} = \|\mathbf{f}_n\|\mathbf{z}_s + \|\mathbf{f}_x\|\mathbf{x}_s + \left\|\mathbf{f}_y\right\|\mathbf{y}_s \tag{35}$$

$$\|\mathbf{f}_n\| = \max\{K_n z^{3/2} + D_n K_n z^{1/2}\dot{z},\ 0\} \tag{36}$$

$$\|\mathbf{f}_x\| = \begin{cases} \mu\|\mathbf{f}_n\|, & K_f x + D_f K_f \dot{x} > \mu\|\mathbf{f}_n\| \\ K_f x + D_f K_f \dot{x}, & \text{else} \\ -\mu\|\mathbf{f}_n\|, & K_f x + D_f K_f \dot{x} < -\mu\|\mathbf{f}_n\| \end{cases} \tag{37}$$

where z, K_n, and D_n are the penetration depth, ground stiffness, and ground damping coefficient, respectively. K_f, D_f, x, and μ are the friction spring stiffness, friction spring damping coefficient, offset distance in $\mathbf{x}_s$ direction away from the first contact point, and friction coefficient, respectively. $\left\|\mathbf{f}_y\right\|$ is computed the same as the $\|\mathbf{f}_x\|$ except replacing x with y.

The contact detection problem (computing the contact points) for the legs and the articulated tail are straightforward, which are just the foot points M_l, T_l, A_l, and the tail joint points J_j. The contact point calculation for the continuum tail, however, requires special treatments since it changes continuously as the tail configuration changes. An easy way to find out these points is to calculate the lowest point of each tail segment, denoted as $C_{i,ct} = \{O_{i,x} : x \in [0, 1]$ minimizes the z component of $\mathbf{p}_{i,o,x}\}$. Note that x here has the same sense as that in Sect. 2.3, instead of the x in Eq. (39). The necessary condition for x to minimize the z component of $\mathbf{p}_{i,o,x}$ could be obtained by letting the derivative of Eq. (20) with respect to x vanish, which yields:

$$\mathrm{c}_{\bar{x}}a_{i,z} + \mathrm{s}_{\bar{x}}b_{i,z} = 0 \rightarrow \bar{x} = \mathrm{atan}(-a_{i,z}/b_{i,z})/\theta_i \tag{38}$$

where $\bar{x}$ is the optimal value of x, $a_{i,z}$ and $b_{i,z}$ are the z component of $\mathbf{a}_i$ and $\mathbf{b}_i$, respectively. Therefore, the contact points $C_{i,ct}$ could be determined as

$$C_{i,ct} = \begin{cases} O_i, & \mathrm{atan}\left(-a_{i,z}/b_{i,z}\right) \le 0 \\ O_{i,\bar{x}}\,, & \text{else} \\ O_{i+1}, & \mathrm{atan}\left(-a_{i,z}/b_{i,z}\right) \ge \theta_i \end{cases} \tag{39}$$

For the cases that $\bar{x}$ is undetermined (e.g., when the bending plane of segment i is horizontal), $\bar{x}$ is set to 0.5. For the cases that the tail segment becomes a line segment, the contact points are manually set as its two endpoints.

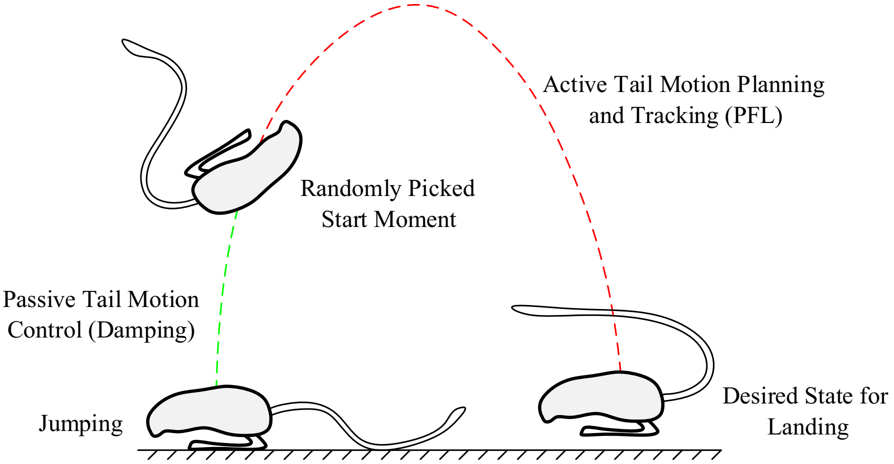

Fig. 6 Kangaroo rat tail motion control framework where the green dashed lines indicate the passive tail motion control phase, and the red dashed lines indicate the active tail motion control phase

3 Tail motion control

Controlling the motion of a 3D biped with a multi-segment serpentine tail is more challenging than a 2D biped [5, 10, 33] or a 3D single body with a single-segment single-link tail [7–9, 13, 32, 33] since the traditional intuition-based tail motion planning and control methods [5–20, 32, 33] are not applicable anymore. To solve this problem, we propose to use the numerical optimal control method [41] to automatically synthesize the tail motion for the kangaroo rat agile motions. The existing numerical optimal control techniques could be roughly divided into two classes: the indirect methods such as differential dynamic programming (DDP) [42], and the direct methods such as direct collocation [43] or orthogonal collocation [44]. In this paper, we chose to use the direct collocation method [43] due to its robustness for high-dimensionality systems and its flexibility to include advanced constraints (e.g., box constraints for joint motion range and joint torque limit). The motion planning will only focus on the airborne phase since the tail motion in this phase is more challenging to be planned manually. More importantly, contacts are not involved in the airborne phase and thus the system dynamics are smooth, which makes the optimization process efficient and robust. After the tail motion trajectory was generated, a partial feedback linearization (PFL) based controller [45] was implemented to track the trajectory. This motion control framework is illustrated in Fig. 6 where the active tail motion control starts from a randomly picked moment when the kangaroo rat is in the air and ends until the moment before the kangaroo rat lands on the ground. All the motions in other moments use a passive tail controller (simple damping) for which the details will be presented in Sect. 3.3.

3.1 Trajectory optimization through direct collocation

The core idea of the direct collocation technique is to treat both the system states and control inputs as decision variables for the nonlinear programming (NLP) problem [46, 47]. For the motion planning problem of the kangaroo rat in the airborne phase, the objective is to design appropriate tail motion profiles $\mathbf{q}_{t,r}(t)$, $\dot{\mathbf{q}}_{t,r}(t)$ and torque profile $\boldsymbol{\tau}_{ta,r}(t)$ such that the kangaroo rat body could reach desired orientation before landing. Using the direct collocation technique, this problem could be formulated as an NLP problem:

$$\text{minimize}: \ell_0(t_N, \mathbf{q}_1, \dot{\mathbf{q}}_{\boldsymbol{1}}, \mathbf{u}_{\boldsymbol{1}}, \mathbf{q}_N, \dot{\mathbf{q}}_N, \boldsymbol{\tau}_{\boldsymbol{ta},N}) + \sum_{\boldsymbol{k}=\boldsymbol{1}}^{\boldsymbol{N}} \ell_{\boldsymbol{k}}(\mathbf{q}_{\boldsymbol{k}}, \dot{\mathbf{q}}_{\boldsymbol{k}}, \boldsymbol{\tau}_{\boldsymbol{ta},\boldsymbol{k}}) \tag{40}$$

$$\begin{aligned}\text{subject to}: \dot{\mathbf{q}}_{k+1} - \dot{\mathbf{q}}_k &+ \frac{h}{2}\mathbf{H}_k^{-1}(\mathbf{C}_k - \mathbf{J}_{ta,k}\boldsymbol{\tau}_{ta,k}) \\ &+ \frac{h}{2}\mathbf{H}_{k+1}^{-1}(\mathbf{C}_{k+1} - \mathbf{J}_{ta,k+1}\boldsymbol{\tau}_{ta,k+1}) \\ &= 0\end{aligned} \tag{41}$$

$$\mathbf{q}_{k+1} - \mathbf{q}_k - \frac{h}{2}(\dot{\mathbf{q}}_k + \dot{\mathbf{q}}_{k+1}) = 0 \tag{42}$$

$$\begin{aligned}&\mathbf{f}_{bc}\left(\mathbf{q}_1, \dot{\mathbf{q}}_1, \mathbf{u}_1, \mathbf{q}_N, \dot{\mathbf{q}}_N, \boldsymbol{\tau}_{ta,N}\right) \leq 0, \\ &\mathbf{g}_{pc}\left(\mathbf{q}_k, \dot{\mathbf{q}}_k, \boldsymbol{\tau}_{ta,k}\right) \leq 0,\ bc, pc \in \mathbb{N}\end{aligned} \tag{43}$$

where the decision variables are $p_d = \{t_N, \mathbf{q}_1, \underline{\mathbf{q}}_{\boldsymbol{1}}, \boldsymbol{\tau}_{\boldsymbol{ta}},$ $1, \ldots,$ $\mathbf{q}_N, \underline{\mathbf{q}}_{\boldsymbol{N}}, \boldsymbol{\tau}_{\boldsymbol{ta},N}\}$. ℓ_0 and ℓ_k are the boundary objective and the path objective functions, respectively. $\mathbf{f}_{bc}$ and $\mathbf{g}_{pc}$ are the boundary and path constraints, respectively, including all the box constraints (e.g., joint motion range and joint torque limit) and more complicated kinematic constraints (e.g., equality constraints for the final state). Note that the dynamic constraints in Eqs. (41) and (42) do not include the GRF term (referring to Eq. (1) to see the difference) due to the airborne assumption. Moreover, Eqs. (41) and (42) implement the trapezoidal quadrature rule to enhance numerical stability. Other quadrature rules such as the Hermite-Simpson rule could be also used to increase the numerical integration accuracy with a cost of longer computation time [47].

3.2 Nontrivial optimization implementation details

Since the system has $d = 6 + 2m$ DOF and $2m$ inputs (the leg inputs γ_l and δ_l are ignored since the leg motions do not affect the airborne dynamics), the NLP problem complexity could easily go up to hundreds of the decision variables and involve highly nonlinear constraints, which makes the solving process time-consuming and susceptible to an infeasible local minimum. To ease these issues, several special optimization techniques are used.

The first technique is a warm-start technique, which utilizes dynamic simulation to generate the initial trajectory guess. This way, the initial guess always satisfies the dynamics constraints in Eqs. (41) and (42), which helps the NLP to converge faster. The dynamic simulation is carried out by solving Eq. (1) using the given boundary value $\mathbf{q}_1 = \mathbf{q}(t_1)$ and time span $[t_1, t_N]$. The tail joint torques are computed based on a purely damping controller (Eq. (47)), which will be detailed in the next section. The second technique is to design the objectives or constraints according to the simulation results. For instance, for the airborne righting tasks, the boundary constraints $\mathbf{f}_{bc}$ in Eq. (43) are usually set to make the body vertically up at the end of the trajectory, i.e., $\phi_x(t_N) = 0$, $\phi_y(t_N) = 0$. However, the body pitch angle ϕ_x and roll angle ϕ_y are both modulo 2π numbers. Therefore, strictly driving the body orientation back to $\phi_x = 0$ and $\phi_y = 0$ is not necessary. Instead, depending on the simulated results, $\phi_x(t_N)$ and $\phi_y(t_N)$ could be set as the closest multiples of 2π to the end body orientation. For instance, if the simulation shows that the body pitch angle reaches 15 radians and the body roll angle reaches $-$ 5 radians at the end, $\phi_x(t_N)$ and $\phi_y(t_N)$ constraints could be set to equal to 4π and -2π, respectively. This technique takes advantage of the existing system momentum and tends to make the system perform minimal work to reach the desired orientation.

3.3 Trajectory interpolation and joint-level control

The obtained tail trajectory from the NLP is a table of isolated values on collocation points. To construct a continuous trajectory for joint-level control, piecewise spline interpolation is utilized using the MATLAB function *spline*. To track this tail trajectory in the simulation, a PFL controller that was previously proposed by the authors [45] is used. That is, the tail control effort $\boldsymbol{\tau}_{ta}$ is calculated to just linearize part of the system dynamics, which is, in this case, the dynamics corresponding to the tail part of the generalized coordinates $\mathbf{q}_t$. Therefore, to track the tail trajectory $\mathbf{q}_{t,r}(\mathrm{t})$, the system output is constructed as

$$\mathbf{y} = \mathbf{q}_t - \mathbf{q}_{t,r}(\mathrm{t}) \tag{44}$$

To linearize the system to asymptotically stabilize this output, a spring-damper system that is known to be stable is constructed as

$$\ddot{\mathbf{y}} + \mathbf{K}_d\dot{\mathbf{y}} + \mathbf{K}_p\mathbf{y} = 0 \tag{45}$$

where $\mathbf{K}_d = K_d\mathbf{I}_{2m\times 2m}$ and $\mathbf{K}_p = K_p\mathbf{I}_{2m\times 2m}$ with $K_d, K_p > 0$ and $\mathbf{I}_{2m\times 2m} \in \mathbb{R}^{2m\times 2m}$ being the identity matrix. Substituting Eq. (44) into Eq. (55), using a selection matrix $\mathbf{S} \in \mathbb{R}^{2m\times d}$ such that $\mathbf{q}_t = \mathbf{S}\mathbf{q}$, and using Eq. (1), Eq. (45) could be solved as

$$\boldsymbol{\tau}_{ta} = \mathbf{X}^{+}(\mathbf{S}\mathbf{H}^{-1}\mathbf{C} + \ddot{\mathbf{q}}_{t,r} + \mathbf{K}_d(\dot{\mathbf{q}}_{t,r} - \dot{\mathbf{q}}_t) + \mathbf{K}_p(\mathbf{q}_{t,r} - \mathbf{q}_t)) \tag{46}$$

in which $\mathbf{X} = \mathbf{S}\mathbf{H}^{-1}\mathbf{J}_{ta}^{T}$ and $\mathbf{X}^{+}$ is the Moore–Penrose inverse of $\mathbf{X}$. Again, the GRF term does not appear due to the airborne assumption. It is worth noting that in practice, we did not use the $\ddot{\mathbf{q}}_{t,r}$ term and it turned out that the controller still worked well. It is also worth noting that due to the coupled and highly nonlinear

Table 1 Kangaroo rat model properties

Var	Value	Var	Value	Var	Value
g	9.8 ms^{-2}	K_n	1000 Nm^{-1}	K_{dmp}	0.001
L_{h2d}	8 mm	D_n	0.75	m_b	105.72 g
L_{d2k}	24 mm	K_f	600 Nm^{-1}	${}^{b}\boldsymbol{I}_b$	diag ([672.2 363.9 727.9]) gcm^2
L_{k2a}	32 mm	D_f	0.01	Tail Mass	3.51 g
L_{a2m}	35 mm	μ	1	Tail Length	194.66 mm
L_{m2t}	16 mm	K_d	400	${}^{b}\mathbf{p}_{b2t}^{T}$	$[0{-}40.5\ 4.5]$ mm
δ_m	30°	K_p	4e4	${}^{b}\mathbf{p}_{b2h,l}^{T}$	$[(-1)^{l+1}22.5{-}18\ 11.25]$ mm

dynamics between the tail and the body, a simple model-free proportional-derivative (PD) controller turned out not to work well in this case, which is the major reason that we chose to use PFL, which empirically shows better tracking performances.

For the phase when the tail is not tracking a prescribed trajectory, the tail joint controller is set to a passive damping controller, which simply resists the tail motion and purely consumes the system energy. The control effort is calculated as

$$\boldsymbol{\tau}_{ta} = -K_{dmp}\mathbf{I}_{2m\times 2m}\dot{\mathbf{q}}_t \tag{47}$$

4 Numerical experiments

All computations are carried out using MATLAB. The simulation is performed using a variable step ordinary differential equation (ODE) solver *ode45* with an absolute tolerance of 1e-6 and a relative tolerance of 1e-8 to guarantee the simulation accuracy. The NLP problem for the tail trajectory optimization is solved using the function *fmincon* with "interior-point" method. The computing environment consists of a workstation with an AMD Ryzen 9 3900X 12 Core CPU (3.793 GHz) and a MATLAB R2021b distribution running on Microsoft Windows 10 Education operating system. All the model properties are summarized in Table 1 where the critical kangaroo rat parameters (body mass, tail mass, tail length) are taken from the measurements in [31]. The detailed tail parameters such as the segment length of the continuum tail or the link mass of the articulated tail, are calculated by evenly distributing the total tail properties onto each segment/link, depending on the actual segment number m and link number n. For instance, the link mass of the articulated tail is determined to be $m_{at} = \text{tail mass}/mn$. The MOI of the body and each link in the articulated tail model are estimated by assuming that each rigid body is an ellipsoid. The COM of each articulated tail link locates at the geometric center of the link, i.e., $L_{j2j} = 2L_{j2c}$. It is worth noting that the friction coefficient μ is set to 1 in the airborne righting motion tests in Sects. 4.2 and 4.3 to guarantee a no-slip condition between the foot and the ground during jumping. This value is later set to 0.5 in the tail-ground contact motion tests in Sect. 4.4 to allow slippage between the tail and the ground. This change is accommodated to facilitate the numerical computations without affecting the motion conclusions.

Since this paper focuses on the tail behaviors, the leg motion is planned manually, and only simple trajectories are used. For instance, for the airborne simulations, the leg motions are designed to be

$$\gamma_l = \frac{7}{6}\pi,\ \delta_l = \max\left\{-\frac{1}{12}\pi - 70(t - t_{kick}), -\frac{5}{12}\pi\right\},\ t \geq t_{kick} \tag{48}$$

which keeps γ_l constant and rotates δ_l in a constant speed from −15° to −75°. t_{kick} is the moment to start the leg motion and in this paper, all the simulations use the same $t_{kick} = 0.4$s. This leg kicking motion drives the kangaroo rat to jump to a height of around 0.6 m. It is worth noting that for this kicking motion, the kangaroo rat model uses its ankle (A) point and metatarsophalangeal (M) point to push the ground, instead of the toe (T) point. This is similar to the jumping motion exhibited by the real kangaroo rat.

For better illustration, animations were created for the simulations in this section, and they can be found online at https://youtu.be/wy067QQ0Cvs.

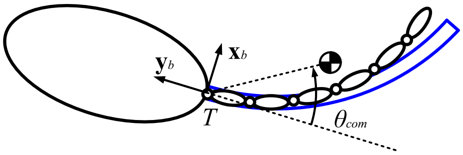

Fig. 7 Tail trajectory setting for the comparative study

4.1 Comparative study on tail models

This section investigates the performance difference between the continuum tail model and the articulated tail model. Since the fundamental difference between these two models is that the articulated model discretizes the continuum tail, this comparative study aims to investigate the effects of the number of links on the tail performance. Therefore, the tail segment number is set to 1 ($m = 1$), and standard airborne yaw rotation motions are simulated. All other model parameters are set to be the same. The tail trajectories are planned so that the articulated tail has the same COM angle as the continuum tail, as shown in Fig. 7. This criterion was previously proposed by the authors to compare the dynamic differences for different robotic tail structures [35]. The tail COM angle θ_{com} is defined as the rotation angle of the line segment connecting the tail COM and the tail mounting point T. To compute the θ_{com} of the continuum tail (which is an arc), Eq. (86) in [35] is used, which gives

$$\tan(\theta_{com}) = \frac{\theta_1 - \sin\theta_1}{1 - \cos\theta_1} \tag{49}$$

The θ_{com} trajectory is planned to be a simple point-to-point motion from 0 radians to atan$(\pi/2)$ radians using cubic polynomial, which corresponds to θ_1 from 0° to 180°.

The comparison results are presented in Fig. 8 where the Fig. 8a shows the body yaw responses $\phi_z(t)$ for the continuum tail model and the articulated tail model (four links are used, i.e., $n = 4$), respectively. Figure 8b plots the trend of the body yaw angle differences as the link number increases. From Fig. 8b, it can be found that the overall model difference is small (less than 0.5°) and this difference approaches zero as more links are used for the articulated model. The largest difference happens when the link number is four. However, even for this worst case, the absolute model difference is still small, which can be also observed from Fig. 8a.

4.2 Airborne righting tests

From the results of Sect. 4.1, the dynamic difference between the articulated tail model and the continuum tail model is not significant, especially when the link number of the articulated tail becomes large. Therefore, for this section and the next section, the experiments will only focus on the articulated tail structure. This section aims to verify the developed dynamic model and the tail motion controller, where the tail trajectory is generated by the direct collocation method in Sect. 3. The objective function and constraints settings are summarized as follows:

(1) $\ell_0 = 0$, $\ell_1 = h\boldsymbol{\tau}_{ta,1}^T\boldsymbol{\tau}_{ta,1}/2$, $\ell_N = h\boldsymbol{\tau}_{ta,N}^T\boldsymbol{\tau}_{ta,N}/2$, $\ell_k = h\boldsymbol{\tau}_{ta,k}^T\boldsymbol{\tau}_{ta,k}$ $(k = 2, 3, \ldots, N-1)$ to minimize the control effort over a given time span. $h = t_N/(N-1)$ is the step size.

(2) $\mathbf{f}_1 = \mathbf{q}_1 - \mathbf{q}(t_1) = \mathbf{0}$, $\mathbf{f}_2 = \boldsymbol{\phi}_{b,N} - \boldsymbol{\phi}_{b,d,N} = \mathbf{0}$, $\mathbf{f}_3 = \dot{\boldsymbol{\phi}}_{b,N} = \mathbf{0}$ to start from a given initial state $\mathbf{q}(t_1)$ and reach a desired body orientation $\boldsymbol{\phi}_{b,d,N}$ with a zero body angular velocity.

(3) $\mathbf{g}_1 = \mathbf{q}_{t,k} - \beta_{t,lim}\mathbf{e}_{2m} \leq \mathbf{0}$,
$\mathbf{g}_2 = -\mathbf{q}_{t,k} - \beta_{t,lim}\mathbf{e}_{2m} \leq \mathbf{0}$,
$\mathbf{g}_3 = \boldsymbol{\tau}_{ta,k} - \beta_{ta,lim}\mathbf{e}_{2m} \leq \mathbf{0}$,
$\mathbf{g}_4 = -\boldsymbol{\tau}_{ta,k} - \beta_{ta,lim}\mathbf{e}_{2m} \leq \mathbf{0}$,
$\mathbf{g}_5 = [t_1 - t_N \; t_N - t_1 - t_s]^T \leq \mathbf{0}$ to add joint motion limit ($\beta_{t,lim} = 180°/mn$), joint torque limit ($\beta_{ta,lim} = 0.02$Nm), and motion planning time span $t_s > 0$. $\mathbf{e}_{2m}$ is a column vector of ones that has $2m$ elements.

The initial state $\mathbf{q}(t_1)$ for the trajectory optimization is selected at a random moment t_1 after the kangaroo rat leaves the ground, although this "random" selection has a prerequisite that it must guarantee that there exists at least one feasible solution to the NLP problem. For instance, if t_1 is chosen to be too close to t_N, it is possible that there does not exist a feasible tail trajectory that can drive the system to the desired state, i.e., no matter how the tail rotates, the kangaroo rat is not able to reach its desired posture in time. This arbitrary selection of starting moment is meant to show the effectiveness of the motion planning algorithm on handling the airborne righting tasks for any given state. For the experiments in this section, we used a one-segment tail with three links per segment ($m = 1$, $n = 3$), selected $t_1 = 0.5$s (0.1 s after the leg

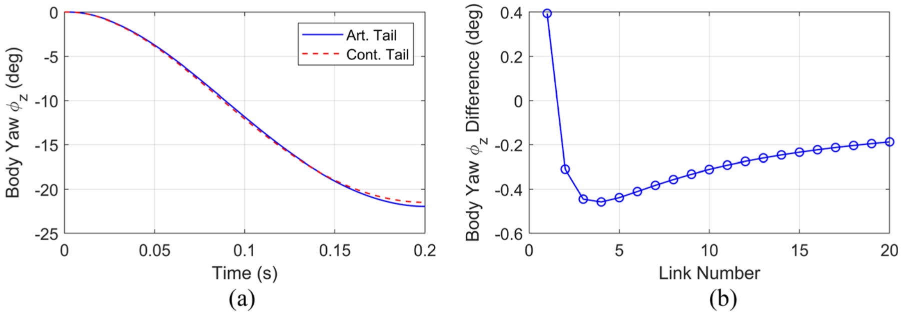

Fig. 8 Comparison results for the continuum tail model and the articulated tail model: **a** the body yaw motion responses $\phi_z(t)$, and **b** the final body yaw angle differences for different link number

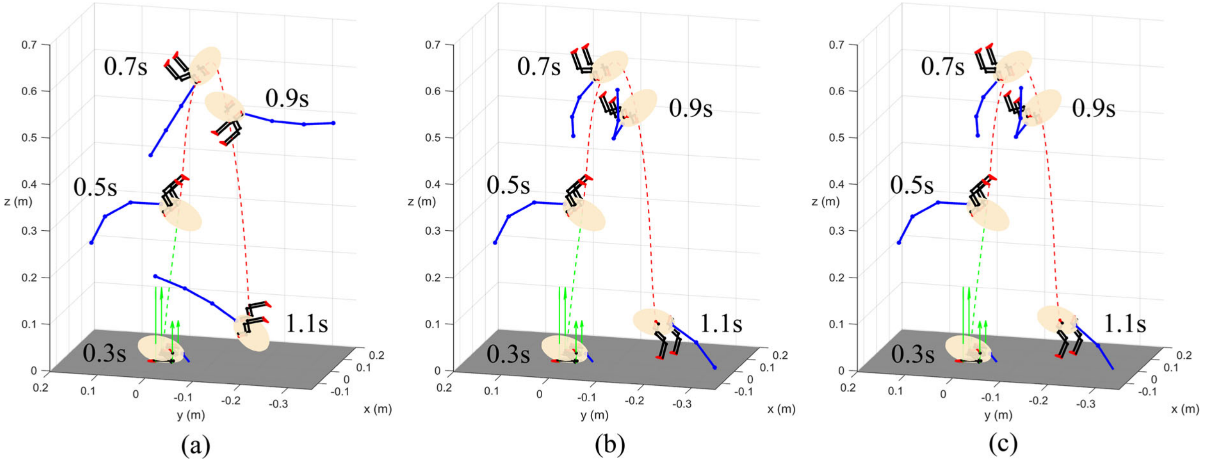

Fig. 9 Snapshots of **a** a simulated jumping motion without active tail control, **b** a designed whole-body motion using the proposed motion planning algorithm, and **c** the simulated motion by tracking the designed tail motion

kicking the ground), $t_s = 0.58$s (the moment just before the landing), used a desired final posture of $\phi_x = 0$, $\phi_y = 0$, and set a collocation rate of 100 Hz (step size of 10 ms). The snapshots for the entire jumping motion with active tail control are shown in Fig. 9b and c where the dashed lines show the body COM (point B) trajectories. To compare the tail motion effects, a simulation for the same jumping motion but without active tail motion control was also conducted and included in Fig. 9a. The entire motion in Fig. 9a was simulated using the full dynamics of Eq. (1) and the tail controller of Eq. (47). In Fig. 9b, the green segment (green dashed line) of the motion was simulated using Eqs. (1) and (47), but the red segment (red dashed line) was a direct plot of the NLP results in Sect. 3.1, which used all the designed motion information including both the body motion and the tail motion. To verify that the designed motion was actually feasible and can meet the desired goal, Fig. 9c simulated the red part using Eq. (1) and the tail controller in Eq. (46). Note the slight differences of how Fig. 9b used the designed whole-body motion for plotting while Fig. 9c used only the tail trajectory. This is because in simulation, the body motion is the under-actuated DOF and only the tail trajectories could be properly controlled. It is also worth noting that the tail contact model was not included for these experiments and therefore the tail could penetrate the ground in Fig. 9. Comparing Fig. 9a and c validates the tail's functionalities in helping the airborne righting maneuvers of kangaroo rats. Comparing Fig. 9b and c verifies the effectiveness of the motion planning algorithms in Sect. 3.1 although only a simple trapezoidal quadrature rule was used.

Figure 10 shows the time domain details for the motions in Fig. 9, where the Fig. 10a plots the body COM trajectory $\mathbf{p}_b(t)$ ($p_{b,x}$, $p_{b,y}$, and $p_{b,z}$ are the x, y, z components, respectively), Fig. 10b and c show the body orientation trajectories, Fig. 10d plots the tail joint trajectories, Fig. 10e plots the trajectory tracking

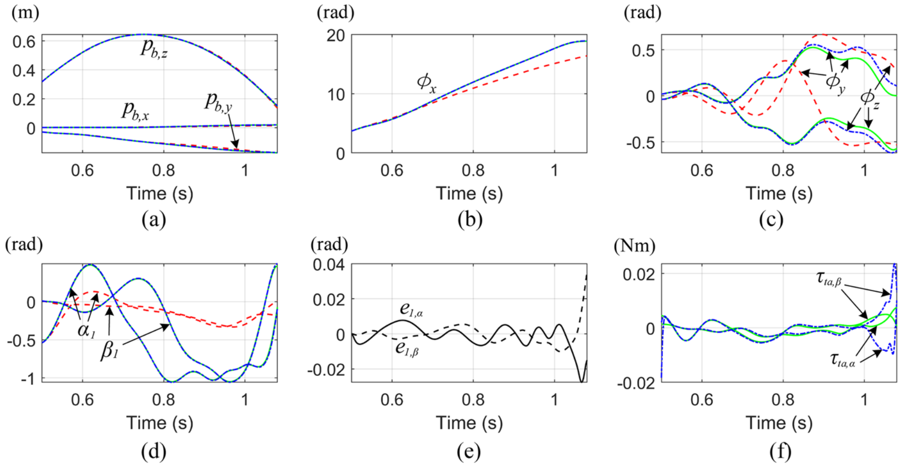

Fig. 10 Detailed plots of the planned motion (green solid line), the simulated motion with active tail control (blue dot-dash line), and the simulated motion with passive tail control (red dashed line)

error of the tail joints $e_{1,\alpha} = \alpha_1 - \alpha_{1,d}$ and $e_{1,\beta} = \beta_1 - \beta_{1,d}$ ($\alpha_{1,d}$ and $\beta_{1,d}$ are the planned tail joint trajectory from the NLP), Fig. 10(f) plots the planned and simulated tail joint torques. Further calculation shows that the optimal trajectory found by the motion planning algorithm consumes 0.0165 Joule energy and requires a peak power of 0.2699 Watt. The mean value and standard deviation of the tail joint trajectory tracking errors are -0.0012 rad and 0.0065 rad for $e_{1,\alpha}$, -0.0003 rad and 0.0055 rad for $e_{1,\beta}$.

According to Fig. 10a, the matching trend among the three trails verifies the effectiveness of the motion planning algorithm. It also implies that the tail motion has marginal effects on the body COM motion. However, Fig. 10b and c demonstrate the tail's usefulness in affecting the body's orientation, although the mismatch between the planned (blue dot-dashed line) and simulated (green solid line) body orientation trajectories reflect the accuracy limit of using the trapezoidal quadrature rule to implement the dynamics constraints. Figure 10d and e show the effectiveness of the PFL controller for the tail's joint-level trajectory tracking control. Figure 10f first verifies how the PFL controller can generate appropriate control effort. It also shows how the open-loop motion planning algorithm can generate a similar control effort as the closed-loop feedback controller.

To further investigate the numerical aspects (e.g., accuracy and efficiency) of the motion planning algorithm, different collocation frequencies are used, and the results are collected in Fig. 11. The dynamics error is measured based on the distance between the planned motion $\mathbf{x}_{nlp}$ and the simulated motion $\mathbf{x}_{track}$ with the planned tail trajectory being tracked, which is a L^2-norm:

$$\begin{aligned}\text{Dyn.Err.} &= \left\| \mathbf{x}_{nlp}(t) - \mathbf{x}_{track}(t) \right\|_2 \\ &= \sqrt{\int_{t_1}^{t_N} (\mathbf{x}_{nlp} - \mathbf{x}_{track})^T (\mathbf{x}_{nlp} - \mathbf{x}_{track}) dt}\end{aligned} \tag{50}$$

From the results, it can be found that increasing the collocation frequency helps reduce the dynamics error but increased computation time. However, when the dynamics error is already reduced to a low level, this trend does not continue (> 100 Hz in Fig. 11). The iteration number and objective value are shown not to be significantly affected by the collocation frequency.

4.3 Comparative study on tail segmentation

After verifying the modeling and control frameworks using a one-segment tail, this section investigates the differences of the tail with multiple segments, that is, how the tail's number of segments affects the airborne righting performance as well as the numerical aspects of the trajectory optimization algorithm. For this purpose, the total number of links of the articulated tail is set to 6 so that it could be divided by the segment numbers 1, 2, and 3. For instance, when the segment number is set to $m = 2$, the link number is set to $n = 3$. The remaining settings are the same as in Sect. 4.2 except for changing $\beta_{t,lim} = 360^\circ/mn$ to allow a larger workspace of the tail to find feasible solutions for the NLP problem.

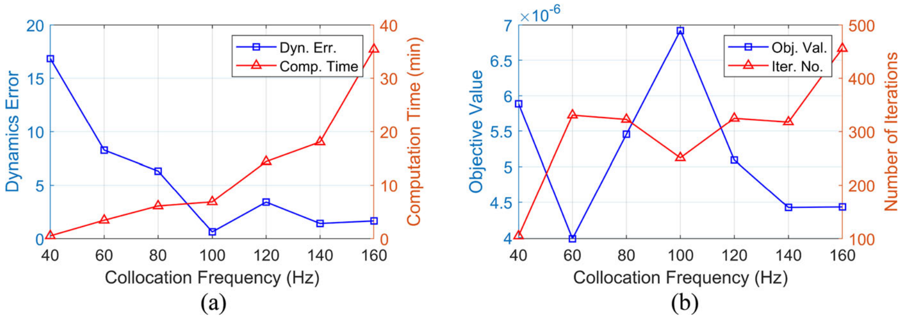

Fig. 11 Computational performance of the motion planning algorithm

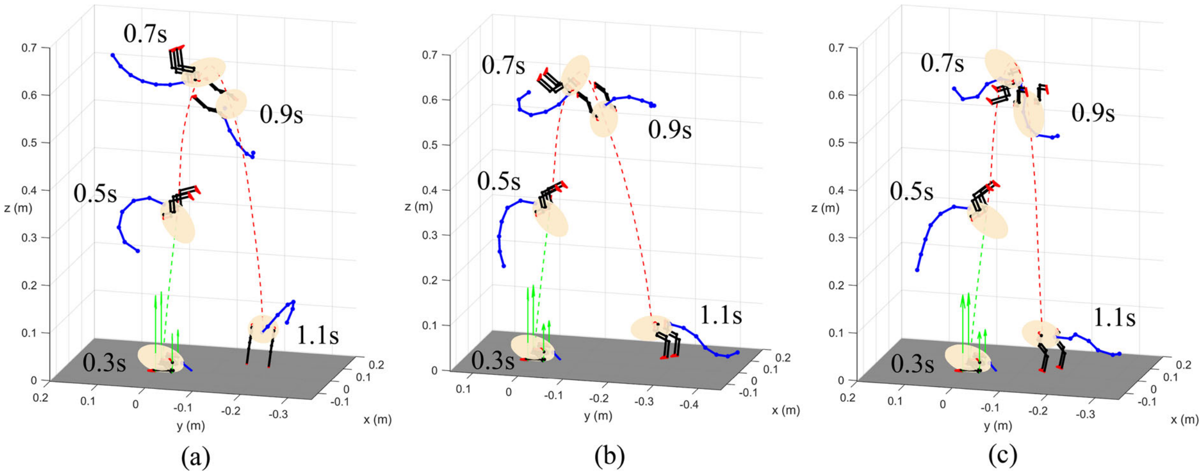

Fig. 12 Snapshots of the planned motion of **a** $m = 1, n = 6$, **b** $m = 2, n = 3$, and **c** $m = 3, n = 2$

The resulting motion and trajectories are presented in Figs. 12 and 13, where Fig. 13a–c plot the body orientation trajectories $\boldsymbol{\phi}_b(t)$, Fig. 13d–f plot the tail trajectories $\mathbf{q}_t(t)$, and Fig. 13g–i plot the tail joint torque profiles $\boldsymbol{\tau}_{ta}(t)$. From the figures, it can be found that more segments in the tail enable more delicate control on the body movement and result in a smaller control effort on the tail joints (see Fig. 13g–i). Other statistical information of interest is collected in Table 2 where the tracking error is computed according to $\max\{\mathrm{abs}\{e_{1,\alpha}, e_{1,\beta}, \ldots, e_{3,\alpha}, e_{3,\beta}\}\}$. Table 2 shows similar observations as in the figures whereas more segments in the tail tend to save energy and reduce peak power by finding better tail trajectories (with lower optimal value). However, this trend is not significant between the two-segment case and the three-segment case. It is worth noting that Table 2 shows a noticeable dynamics error (the larger the error, the larger the difference between the planned motion and the simulated motion). However, based on the conclusions from Sect. 4.2, this issue could be solved by increasing the collocation frequency.

4.4 Tail-ground contact tests

When the kangaroo rat is on the ground, the tail could be used as an additional appendage to assist the locomotion. This section investigates this static tail functionality by conducting case studies of a jumping motion and a self-recovery motion. Both the articulated tail model and the continuum tail model are included to evaluate the effectiveness of the contact model established in Sect. 2.4. Due to the large number of contact events, the non-stiff ODE solver *ode45* is replaced by a stiff ODE solver *ode15s* for faster evaluation of the contact events. The friction coefficient μ is also adjusted to 0.5 to allow sliding motion on the contact surface.

Table 2 Performance statistics for different number of segments

No. of segments	No. of iterations	Optimization time (min)	Optimal value	Energy (J)	Peak power (W)	Dynamics error	Tracking error (rad)
1	364	15	1.35e–5	0.0308	1.02	3.98	0.0848
2	646	39	5.81e–7	0.0152	0.208	8.21	0.391
3	1469	121	5.06e–7	0.0208	0.551	19.5	0.109

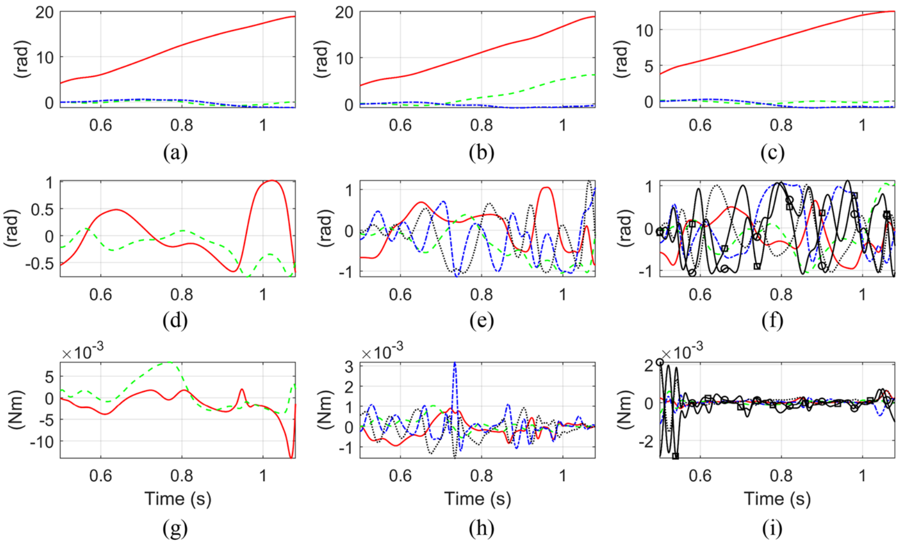

Fig. 13 Detailed plots of the planned motion for one- (subfigures (**a**), (**d**), (**g**)), two- (subfigures (**b**), (**e**), (**h**)), and three-segment (subfigures (**c**), (**f**), (**i**)) tail case. The line style code for the subfigures **a**–**c** is that the "red solid", "green dashed", and "blue dash-dotted" lines correspond to ϕ_x, ϕ_y, and ϕ_z, respectively. The line style code for the subfigures **d**–**f** is that the "red solid" line, "green dashed" line, "blue dash-dotted" line, "black dotted" line, "black solid" line with circle markers, and the "black solid" line with square markers correspond to α_1, β_1, α_2, β_2, α_3, and β_3, respectively. The line style code for the subfigures **g**–**i** is that the "red solid" line, "green dashed" line, "blue dash-dotted" line, "black dotted" line, "black solid" line with circle markers, and the "black solid" line with square markers correspond to $\tau_{ta,\alpha,1}$, $\tau_{ta,\beta,1}$, $\tau_{ta,\alpha,2}$, $\tau_{ta,\beta,2}$, $\tau_{ta,\alpha,3}$, and $\tau_{ta,\beta,3}$, respectively

4.4.1 Case study on jumping motion

Two sets of experiments were designed to compare the difference between the two tail types: one jumping motion using the articulated tail and one jumping motion using the continuum tail. Both cases utilized the tail contact event to stabilize locomotion. For comparison purposes, one additional jumping motion without using the tail for locomotion was also performed. It is worth noting that other than changing the ODE solver and the friction coefficient, the leg trajectory component $\delta_l(t)$ was also changed to rotating from $-40°$ to $-90°$ for kicking and then rotating back to $-30°$ for landing. Both tails used three segments ($m = 3$) and the articulated tail used four links per segment ($n = 4$).

The simulation results are demonstrated in Fig. 14, where Fig. 14a–d show the jumping motion without using the tail support on the ground, Fig. 14e–h show the jumping motion using the articulated tail support on the ground, and Fig. 14i–l show the jumping motion using the continuum tail support on the ground. From the case study results, it can be found that the tail-ground contact model is able to generate realistic contact phenomena, which enables future investigations on contact-rich motion planning and control of the kangaroo rat model. In addition, the case study shows that through environmental contacts, the tail

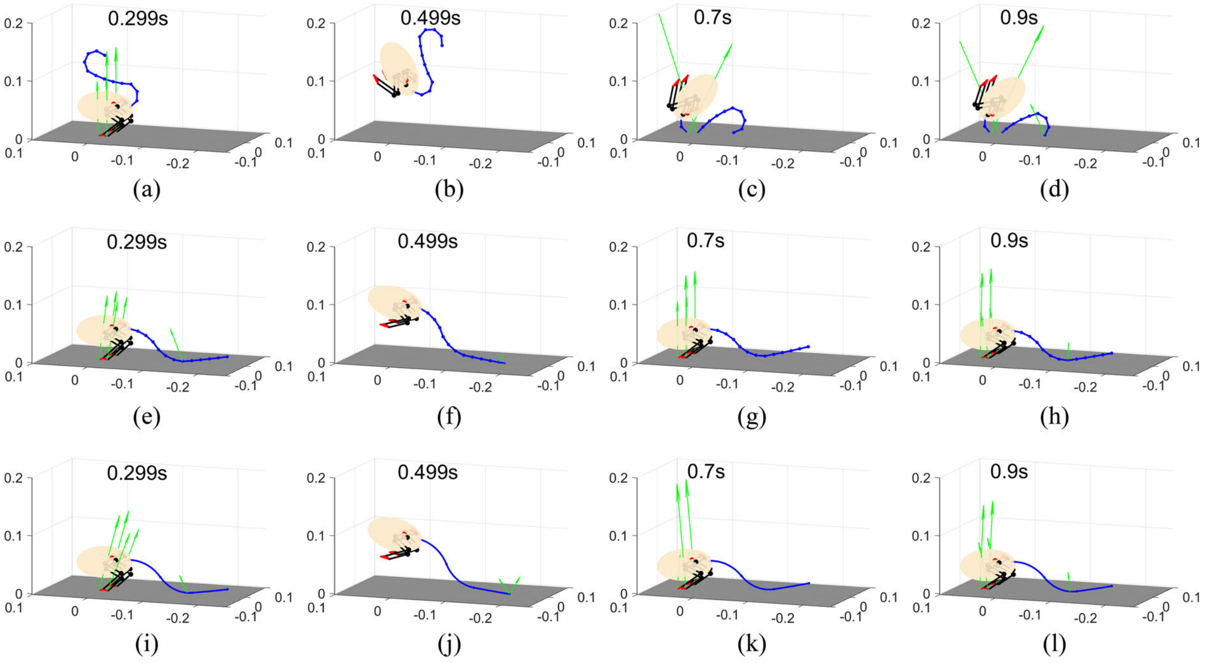

Fig. 14 Jumping motion snapshots

becomes significantly useful in helping the kangaroo rat locomotion.

4.4.2 *Case study on self-recovery motion*

For the self-recovery test, the robot was initialized laying down on the ground. The tail motion was manually planned to push the body to stand up. The results are presented in Fig. 15 where Fig. 15a–d show the snapshots of the self-recovery motion of the kangaroo rat using the articulated tail and Fig. 15e–h show the snapshots of the same motion using the continuum tail. Note that due to the soft contact model, the tail may penetrate the ground for transient motions. From Fig. 15, the tail-ground contact models are verified further, especially for the varying contact point problem (Eqs. 38 and 39) of the continuum tail. The importance of this case study is that it illustrates how the tail could be used to help the manipulation tasks of the kangaroo rat through tail-environment contacts. In these scenarios, the serpentine tail structure helps to enlarge the tail workspace and enhance the tail dexterity significantly, which is a fundamental benefit in comparison with single-link tails.

4.5 Discussion

The tail controller in this paper mostly focuses on the airborne righting motions, for which the dynamics are continuous. In future work, we wish to extend the work into the non-smooth domain, i.e., designing the tail controller together with the leg motion planning, which involves tail-ground contact and leg-ground contact. Due to the rich contact type and large contact number, the traditional method based on mode schedule [48] is less attractive. To solve this problem, through-contact trajectory optimization may be required, such as those based on direct methods [49, 50] and those based on indirect methods [51, 52].

This work was motivated by understanding the tail's functionalities on agile motions of the kangaroo rat from dynamics and control perspectives. This understanding, together with the modeling and control framework developed in this paper, aims to lay the foundation for further developments of agile biped robots with a serpentine robotic tail. Therefore, one important future work is to develop such a robotic system that achieves the same level of agility and dexterity as the kangaroo rat, which may have unique applications as a highly agile terrestrial moving platform.

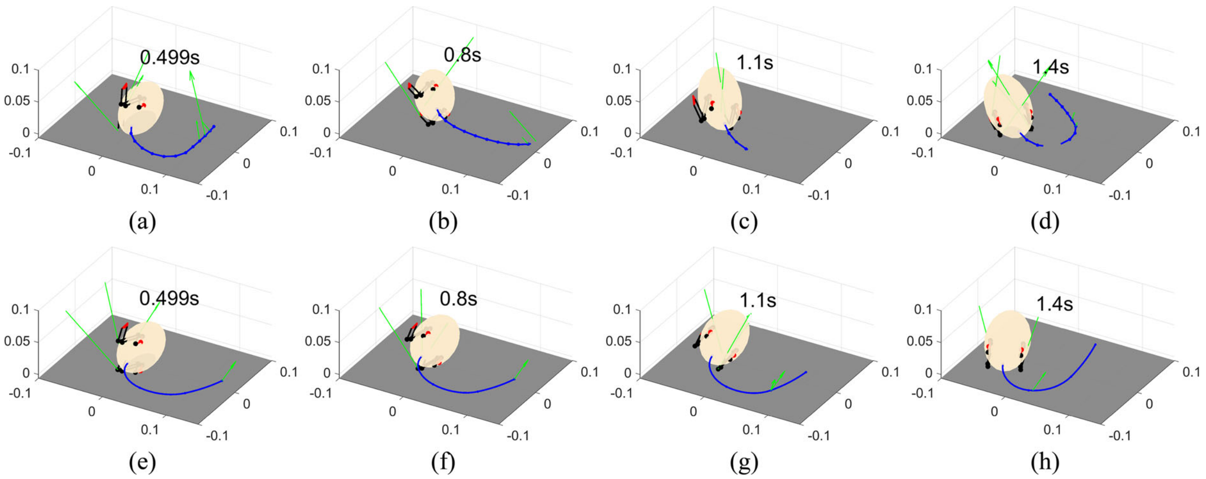

Fig. 15 Self-recovery motion snapshots

Although the proposed dynamic model computes fast (less than 0.1 ms), the motion planning algorithm takes a considerable amount of time (from several minutes to a couple of hours) to converge. To mitigate this drawback in the future, an analytical model with automatic differentiation [53] may be used to fasten the NLP solving process. Moreover, due to the nonlinear nature of the system dynamics, the motion planning algorithm is only able to find a local minimum. To find the global minimum or a unique solution for every motion planning, the model complexity may need to be reduced so that the NLP problem could be formulated as a convex programming problem [54]. In addition, the more generic techniques (e.g., genetic algorithm, Monte-Carlo method) may also be used to find a global optimum or a suboptimum at the cost of a longer computation time. For the specific case in this paper, one practical way is to generate more initial guesses spanning the entire feasible region, and then use a small disturbance to reinitiate the optimization process after finding a local minimum. This technique may help to find a better solution but global optimum is still not guaranteed.

5 Conclusion

This work investigated the agile motions and tail functionalities of the kangaroo rat from the perspective of dynamics and control. Two representative, general serpentine tail models were proposed: one articulated tail model and one continuum tail model. The contact events among the feet, tail, and the ground were modeled using a regularized compliant contact model. To automatically design the tail motion, a direct collocation-based numerical optimal control method was utilized. The planned tail trajectories were then tracked using a partial feedback linearization controller. Using the established dynamic model and control framework, various numerical experiments were performed. A comparative study on the two serpentine tail models was conducted first and the results validated the common impression that the articulated tail model approaches the continuum tail model as the number of links increases. Two sets of airborne righting tests were then performed, focusing on verifying the dynamics and control framework and investigating the differences between tail segmentations, respectively. To test the tail contact model and explore the tail's functionalities in supporting the body, case studies on jumping and self-recovery motions were conducted. The simulation results demonstrated the unique functionalities of the serpentine tail with assisting the kangaroo rat's locomotion.

Acknowledgements This material is based upon work supported by (while serving at) the National Science Foundation under Grant No. 1906727.

Funding This work was supported by the National Science Foundation under Grant No. 1906727.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Code availability Custom code.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

Appendix A: Articulated tail kinematics

The velocities, Jacobians, acceleration, and MOI for each link of the articulated tail model are computed recursively using Eqs. (A1–A13). Eq. (A14) computes the torso MOI. $\mathbf{u}_{x,y}$ is an x-dimension unit column vector with 1 on the y-th entry.

$$\boldsymbol{\omega}_j = \begin{cases} \boldsymbol{\omega}_b, j = 0 \\ \boldsymbol{\omega}_{j-1} + \dot{\alpha}_i \mathbf{x}_{j-1} + \dot{\beta}_i \mathbf{z}_j, j > 0 \end{cases} \tag{A1}$$

$$\mathbf{v}_{j,com} = \mathbf{v}_{j,jnt} + \mathbf{v}_{j,j2c} \tag{A2}$$

$$\mathbf{v}_{j,jnt} = \begin{cases} \mathbf{v}_b + \boldsymbol{\omega}_b \times \mathbf{p}_{b2t}, j = 1 \\ \mathbf{v}_{j-1,jnt} + \mathbf{v}_{j-1,j2j}, j > 1 \end{cases} \tag{A3}$$

$$\begin{cases} \mathbf{v}_{j,j2c} = \boldsymbol{\omega}_j \times \mathbf{p}_{j,j2c} \\ \mathbf{v}_{j,j2j} = \boldsymbol{\omega}_j \times \mathbf{p}_{j,j2j} \end{cases} \tag{A4}$$

$$\mathbf{J}_{j,\omega} = \begin{cases} [\,0_{3\times3} \quad \mathbf{I}_{3\times3} \quad 0_{3\times2m}\,], j = 0 \\ \mathbf{J}_{j-1,\omega} + \mathbf{x}_{j-1}\mathbf{u}_{d,2i+5}^T + \mathbf{z}_j\mathbf{u}_{d,2i+6}^T, j > 0 \end{cases} \tag{A5}$$

$$\mathbf{J}_{j,com} = \mathbf{J}_{j,jnt} + \mathbf{J}_{j,j2c} \tag{A6}$$

$$\mathbf{J}_{j,jnt} = \begin{cases} [\,\mathbf{I}_{3\times3} \quad -\tilde{\mathbf{p}}_{b2t} \quad 0_{3\times2m}\,], j = 1 \\ \mathbf{J}_{j-1,jnt} + \mathbf{J}_{j-1,j2j}, j > 1 \end{cases} \tag{A7}$$

$$\begin{cases} \mathbf{J}_{j,j2c} = -\tilde{\mathbf{p}}_{j,j2c}\mathbf{J}_{j,\omega} \\ \mathbf{J}_{j,j2j} = -\tilde{\mathbf{p}}_{j,j2j}\mathbf{J}_{j,\omega} \end{cases} \tag{A8}$$

$$\dot{\omega}_j = \begin{cases} \dot{\omega}_b, j = 0 \\ \dot{\omega}_{j-1} + \ddot{\alpha}_i \mathbf{x}_{j-1} \\ +\dot{\alpha}_i \tilde{\omega}_{j-1}\mathbf{x}_{j-1} + \ddot{\beta}_i \mathbf{z}_j + \dot{\beta}_i \tilde{\omega}_j \mathbf{z}_j, j > 0 \end{cases} \tag{A9}$$

$$\dot{\mathbf{v}}_{j,com} = \dot{\mathbf{v}}_{j,jnt} + \dot{\mathbf{v}}_{j,j2c} \tag{A10}$$

$$\dot{\mathbf{v}}_{j,jnt} = \begin{cases} \dot{\mathbf{v}}_b + \tilde{\dot{\omega}}_b \mathbf{p}_{b2t} + \tilde{\omega}_b^2 \mathbf{p}_{b2t}, j = 1 \\ \dot{\mathbf{v}}_{j-1,jnt} + \dot{\mathbf{v}}_{j-1,j2j}, j > 1 \end{cases} \tag{A11}$$

$$\begin{cases} \dot{\mathbf{v}}_{j,j2c} = \tilde{\dot{\omega}}_j \mathbf{p}_{j,j2c} + \tilde{\omega}_j^2 \mathbf{p}_{j,j2c} \\ \dot{\mathbf{v}}_{j,j2j} = \tilde{\dot{\omega}}_j \mathbf{p}_{j,j2j} + \tilde{\omega}_j^2 \mathbf{p}_{j,j2j} \end{cases} \tag{A12}$$

$$\mathbf{I}_{j,at} = \mathbf{R}_j\, {}^j\mathbf{I}_{j,at}\mathbf{R}_j^T \tag{A13}$$

$$\mathbf{I}_b = \mathbf{R}_b {}^b\mathbf{I}_b \mathbf{R}_b^T \tag{A14}$$

Appendix B: Continuum tail kinematics

The detailed expression of matrix $\mathbf{E}_{i,v}$ is given as follows where $\mathrm{c}_\theta = \cos\theta_i$, $\mathrm{s}_\theta = \sin\theta_i$, $\mathrm{c}_{2\theta} = \cos2\theta_i$, $\mathrm{s}_{2\theta} = \sin2\theta_i$. Since $\mathbf{E}_{i,v}$ is symmetric, only the upper triangle elements are listed.

$$\mathbf{E}_{i,v}(1,1) = 1$$

$$\mathbf{E}_{i,v}(1,2) = (1 - \mathrm{c}_\theta)/\theta_i$$

$$\mathbf{E}_{i,v}(1,3) = (-1 + \mathrm{c}_\theta + \theta_i \mathrm{s}_\theta)/\theta_i^2$$

$$\mathbf{E}_{i,v}(1,4) = -\mathrm{s}_\theta/\theta_i$$

$$\mathbf{E}_{i,v}(1,5) = (-\theta_i \mathrm{c}_\theta + \mathrm{s}_\theta)/\theta_i^2$$

$$\mathbf{E}_{i,v}(2,2) = 1/2 - \mathrm{s}_{2\theta}/(4\theta_i)$$

$$\mathbf{E}_{i,v}(2,3) = (-2\theta_i \mathrm{c}_{2\theta} + \mathrm{s}_{2\theta})/(8\theta_i^2)$$

$$\mathbf{E}_{i,v}(2,4) = (\mathrm{c}_{2\theta} - 1)/(4\theta_i)$$

$$\mathbf{E}_{i,v}(2,5) = (1 - \mathrm{c}_{2\theta} - 2\theta_i \mathrm{s}_{2\theta} + 2\theta_i^2)/(8\theta_i^2)$$

$$\mathbf{E}_{i,v}(3,3) = (4\theta_i^3 + 6\theta_i \mathrm{c}_{2\theta} + (6\theta_i^2 - 3)\mathrm{s}_{2\theta})/(24\theta_i^3)$$

$$\mathbf{E}_{i,v}(3,4) = \mathbf{E}_{i,v}(2,5) - 1/2$$

$$\mathbf{E}_{i,v}(3,5) = (-1 + (1 - 2\theta_i^2)\mathrm{c}_{2\theta} + 2\theta_i \mathrm{s}_{2\theta})/(8\theta_i^3)$$

$$\mathbf{E}_{i,v}(4,4) = 1 - \mathbf{E}_{i,v}(2,2)$$

$$\mathbf{E}_{i,v}(4,5) = -\mathbf{E}_{i,v}(2,3)$$

$$\mathbf{E}_{i,v}(5,5) = (4\theta_i^3 - 6\theta_i \mathrm{c}_{2\theta} + (3 - 6\theta_i^2)\mathrm{s}_{2\theta})/(24\theta_i^3)$$

The elements in matrix $\mathbf{Q}_{i,v}$ are given as:

$$\mathbf{Q}_{i,v}(1,1) = 1$$

$$\mathbf{Q}_{i,v}(2,1) = \mathbf{Q}_{i,v}(1,2) = (1 - \mathrm{c}_\theta)/\theta_i$$

$$\mathbf{Q}_{i,v}(3,1) = \mathbf{Q}_{i,v}(1,4) = (-1 + \mathrm{c}_\theta + \theta_i \mathrm{s}_\theta)/\theta_i^2$$

$$\mathbf{Q}_{i,v}(4,1) = \mathbf{Q}_{i,v}(1,6) = -\mathrm{s}_\theta/\theta_i$$

$$\mathbf{Q}_{i,v}(5,1) = \mathbf{Q}_{i,v}(1,8) = (-\theta_i \mathrm{c}_\theta + \mathrm{s}_\theta)/\theta_i^2$$

$$\mathbf{Q}_{i,v}(2,2) = 1 - \mathbf{Q}_{i,v}(4,6) = 1/2 - \mathrm{s}_{2\theta}/(4\theta_i)$$

$$\mathbf{Q}_{i,v}(3,2) = (-2\theta_i \mathrm{c}_{2\theta} + \mathrm{s}_{2\theta})/(8\theta_i^2)$$

$$\mathbf{Q}_{i,v}(3,2) = \mathbf{Q}_{i,v}(2,4) = -\mathbf{Q}_{i,v}(5,6) = -\mathbf{Q}_{i,v}(4,8)$$

$$\mathbf{Q}_{i,v}(4,2) = \mathbf{Q}_{i,v}(2,6) = (\mathrm{c}_{2\theta} - 1)/(4\theta_i)$$

$$\begin{aligned}\mathbf{Q}_{i,v}(5,2) &= \mathbf{Q}_{i,v}(2,8) \\ &= (1 - \mathrm{c}_{2\theta} - 2\theta_i \mathrm{s}_{2\theta} + 2\theta_i^2)/(8\theta_i^2)\end{aligned}$$

$$\mathbf{Q}_{i,v}(3,4) = (4\theta_i^3 + 6\theta_i \mathrm{c}_{2\theta} + (6\theta_i^2 - 3)\mathrm{s}_{2\theta})/(24\theta_i^3)$$

$$\mathbf{Q}_{i,v}(3,4) = -\mathbf{Q}_{i,v}(4,9)$$

$$\mathbf{Q}_{i,v}(4,4) = \mathbf{Q}_{i,v}(3,6) = \mathbf{Q}_{i,v}(5,2) - 1/2$$

$$\mathbf{Q}_{i,v}(5,4) = (-1 + (1 - 2\theta_i^2)\mathrm{c}_{2\theta} + 2\theta_i \mathrm{s}_{2\theta})/(8\theta_i^3)$$

$$\mathbf{Q}_{i,v}(5,4) = \mathbf{Q}_{i,v}(4,5) = \mathbf{Q}_{i,v}(3,8) = \mathbf{Q}_{i,v}(2,9)$$

$$\mathbf{Q}_{i,v}(:,3) = 2\mathbf{Q}_{i,v}(:,4)$$

$$\mathbf{Q}_{i,v}(1,5) = \left(2 + \left(\theta_i^2 - 2\right)\mathrm{c}_\theta - 2\theta_i \mathrm{s}_\theta\right)/\theta_i^3$$

$$\mathbf{Q}_{i,v}(2,5) = -\mathbf{Q}_{i,v}(5,8) = \mathbf{Q}_{i,v}(3,4) - 1/3$$

$$\mathbf{Q}_{i,v}(3,5) = (2\theta_i(2\theta_i^2 - 3)\mathrm{c}_{2\theta} - (6\theta_i^2 - 3)\mathrm{s}_{2\theta})/(16\theta_i^4)$$

$$\mathbf{Q}_{i,v}(3,5) = -\mathbf{Q}_{i,v}(5,9)$$

$$\begin{aligned}\mathbf{Q}_{i,v}(5,5) = {} & (3 + 2\theta_i\left(2\theta_i^2 - 3\right)\mathrm{s}_{2\theta} + (6\theta_i^2 - 3)\mathrm{c}_{2\theta} \\ & - 2\theta_i^4)/(16\theta_i^4)\end{aligned}$$

$$\mathbf{Q}_{i,v}(5,5) = \mathbf{Q}_{i,v}(3,9) - 1/4$$

$$\mathbf{Q}_{i,v}(:,7) = 2\mathbf{Q}_{i,v}(:,8)$$

$$\mathbf{Q}_{i,v}(1,9) = \left(\left(\theta_i^2 - 2\right)\mathrm{s}_\theta + 2\theta_i \mathrm{c}_\theta\right)/\theta_i^3$$

The block-wise matrix multiplication notation "$\circ$" is introduced in Sect. 2.3 to express the sum of scalar multiplications (linear combination of vectors). To better present its operations, an example is given here.

$$\mathbf{X} = [\,\mathbf{A}_{3\times2} \quad \mathbf{B}_{3\times2}\,]$$

$$\mathbf{Y} = [\,\mathrm{a} \quad b\,]$$

where the subscripts of $\mathbf{A}$, $\mathbf{B}$ denote their dimensions. a and b are scalars. Then $\mathbf{X} \circ \mathbf{Y}^T$ is evaluated as

$$\begin{aligned}\mathbf{X} \circ \mathbf{Y}^T &= [\,\mathbf{A}_{3\times2} \quad \mathbf{B}_{3\times2}\,] \circ [\,a \quad b\,]^T \\ &= a\mathbf{A}_{3\times2} + b\mathbf{B}_{3\times2}\end{aligned}$$

It has the similar transpose property as matrix multiplication.

$$\left(\mathbf{X} \circ \mathbf{Y}^T\right)^T = \mathbf{Y} \circ \mathbf{X}^T$$

References

1. Hickman, G.C.: The mammalian tail: a review of functions. Mammal Rev. **9**(4), 143–157 (1979)
2. Schwaner, M.J., Hsieh, S.T., Braasch, I., Bradley, S., Campos, C.B., Collins, C.E., Donatelli, C.M., Fish, F.E., Fitch, O.E., Flammang, B.E., Jackson, B.E.: Future tail tales: a forward-looking, integrative perspective on tail research. Integr. Comp. Biol. **61**(2), 521–537 (2021)
3. Young, J.W., Chadwell, B.A., Dunham, N.T., McNamara, A., Phelps, T., Hieronymus, T., Shapiro, L.J.: The stabilizing function of the tail during arboreal quadrupedalism. Integr. Comp. Biol. **61**(2), 491–505 (2021)
4. Dawson, R.S., Warburton, N.M., Richards, H.L., Milne, N.: Walking on five legs: investigating tail use during slow gait in kangaroos and wallabies. Aust. J. Zool. **63**(3), 192–200 (2015)
5. Zeglin, G.J.: Uniroo—a one legged dynamic hopping robot. Bachelor thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (1991)
6. Libby, T., Moore, T.Y., Chang-Siu, E., Li, D., Cohen, D.J., Jusufi, A., Full, R.J.: Tail-assisted pitch control in lizards, robots and dinosaurs. Nature **481**(7380), 181–184 (2012)
7. Jusufi, A., Kawano, D.T., Libby, T., Full, R.J.: Righting and turning in mid-air using appendage inertia: reptile tails, analytical models and bio-inspired robots. Bioinspir. Biomim. **5**(4), 045001 (2010)
8. Chang-Siu, E., Libby, T., Brown, M., Full, R.J., Tomizuka, M.: A nonlinear feedback controller for aerial self-righting by a tailed robot. In: Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany (2013)
9. Libby, T., Johnson, A.M., Chang-Siu, E., Full, R.J., Koditschek, D.E.: Comparative design, scaling, and control

of appendages for inertial reorientation. IEEE Trans. Rob. **32**(6), 1380–1398 (2016)

10. De, A., Koditschek, D.E.: Parallel composition of templates for tail-energized planar hopping. In: Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, USA (2015).
11. Casarez, C.S., Fearing, R.S.: Steering of an underactuated legged robot through terrain contact with an active tail. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain (2018)
12. Kohut, N.J., Pullin, A.O., Haldane, D.W., Zarrouk, D., Fearing, R.S.: Precise dynamic turning of a 10 cm legged robot on a low friction surface using a tail. In: Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany (2013).
13. Patel, A., Boje, E.: On the conical motion of a two-degree-of-freedom tail inspired by the cheetah. IEEE Trans. Rob. **31**(6), 1555–1560 (2015)
14. Norby, J., Li, J.Y., Selby, C., Patel, A., Johnson, A.M.: Enabling dynamic behaviors with aerodynamic drag in lightweight tails. IEEE Trans. Robot. (2021). https://doi.org/10.1109/TRO.2020.3045644
15. Zhao, J., Zhao, T., Xi, N., Mutka, M.W., Xiao, L.: Msu tailbot: controlling aerial maneuver of a miniature-tailed jumping robot. IEEE/ASME Trans. Mechatron. **20**(6), 2903–2914 (2015)
16. Liu, G.H., Lin, H.Y., Lin, H.Y., Chen, S.T., Lin, P.C.: A bio-inspired hopping kangaroo robot with an active tail. J. Bionic Eng. **11**(4), 541–555 (2014)
17. Briggs, R., Lee, J., Haberland, M., Kim, S.: Tails in biomimetic design: analysis, simulation, and experiment. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal (2012)
18. Heim, S.W., Ajallooeian, M., Eckert, P., Vespignani, M., Ijspeert, A.J.: On designing an active tail for legged robots: simplifying control via decoupling of control objectives. Indus. Robot Int. J. **43**(3), 338–346 (2016)
19. Ikeda, F., Toyama, S.: A proposal of right and left turning mechanism for quasi-passive walking robot. In: Proceedings of the International Conference on Advanced Robotics and Intelligent Systems, Taipei, Taiwan (2015)
20. Machairas, K., Papadopoulos, E.: On quadruped attitude dynamics and control using reaction wheels and tails. In: Proceedings of the European Control Conference, Linz, Austria (2015)
21. Rone, W.S., Saab, W., Ben-Tzvi, P.: Design, modeling, and integration of a flexible universal spatial robotic tail. J. Mech. Robot. **10**(4), 041001 (2018)
22. Liu, Y., Wang, J., Ben-Tzvi, P.: A cable length invariant robotic tail using a circular shape universal joint mechanism. J. Mech. Robot. **11**(5), 051005 (2019)
23. Saab, W., Rone, W., Kumar, A., Ben-Tzvi, P.: Design and integration of a novel spatial articulated robotic tail. IEEE/ASME Trans. Mechatron. **24**(2), 434–446 (2019)
24. Liu, Y., Ben-Tzvi, P.: Design, analysis, and integration of a new two-degree-of-freedom articulated multi-link robotic tail mechanism. J. Mech. Robot. **12**(2), 021101 (2020)
25. Santiago, J.L.C., Godage, I.S., Gonthina, P., Walker, I.D.: Soft robots and kangaroo tails: modulating compliance in continuum structures through mechanical layer jamming. Soft Rob. **3**(2), 54–63 (2016)
26. Simon, B., Sato, R., Choley, J.Y., Ming, A.: Development of a bio-inspired flexible tail systemxs. In: Proceedings of the 12th France-Japan and 10th Europe-Asia Congress on Mechatronics, Tsu (2018)
27. Nabeshima, J., Saraiji, M.Y., Minamizawa, K.: Prosthetic Tail: Artificial Anthropomorphic Tail for Extending Innate Body Functions. In: Proceedings of the 10th Augmented Human International Conference, Reims (2019)
28. Rone, W.S., Saab, W., Kumar, A., Ben-Tzvi, P.: Controller design, analysis, and experimental validation of a robotic serpentine tail to maneuver and stabilize a quadrupedal robot. J. Dyn. Syst. Meas. Contr. **141**(8), 081002 (2019)
29. Kangaroo rat mid-air maneuver via tail rotation. Ninja Rat, https://www.youtube.com/watch?v=aV8_iv6SXqc, Retrieved 2021
30. Freymiller, G.A., Whitford, M.D., Higham, T.E., Clark, R.W.: Escape dynamics of free-ranging desert kangaroo rats (Rodentia: Heteromyidae) evading rattlesnake strikes. Biol. J. Lin. Soc. **127**(1), 164–172 (2019)
31. Schwaner, M.J., Freymiller, G.A., Clark, R.W., McGowan, C.P.: How to stick the landing: kangaroo rats use their tails to reorient during evasive jumps away from predators. Integr. Comp. Biol. **61**(2), 442–454 (2021)
32. Moore, J., Gutmann, A., Craig, M., McKinley, P.: Exploring the role of the tail in bipedal hopping through computational evolution. Artif. Life **25**(3), 236–249 (2019)
33. An, J., Chung, T.Y., Lo, C.H.D., Ma, C., Chu, X., Au, K.S.: Development of a bipedal hopping robot with morphable inertial tail for agile locomotion. In: Proceedings of the 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics, New York (2020)
34. Featherstone, R.: Rigid body dynamics algorithms. Springer, Berlin (2014)
35. Liu, Y., Ben-Tzvi, P.: Dynamic modeling, analysis, and comparative study of a quadruped with bio-inspired robotic tails. Multibody Sys. Dyn. **51**(2), 195–219 (2021)
36. Godage, I.S., Webster, R.J., Walker, I.D.: Center-of-gravity-based approach for modeling dynamics of multisection continuum arms. IEEE Trans. Rob. **35**(5), 1097–1108 (2019)
37. Chirikjian, G.S.: Hyper-redundant manipulator dynamics: a continuum approximation. Adv. Robot. **9**(3), 217–243 (1994)
38. Liu, Y., Ben-Tzvi, P.: A new approach to model constant curvature continuum robot dynamics. In: Proceedings of the ASME 2019 Dynamic Systems and Control Conference, Park City, Utah (2019)
39. Azad, M., Featherstone, R.: A new nonlinear model of contact normal force. IEEE Trans. Rob. **30**(3), 736–739 (2014)
40. Flores, P.: Contact mechanics for dynamical systems: a comprehensive review. Multibody Syst. Dyn. (2021). https://doi.org/10.1007/s11044-021-09803-y
41. Rao, A.V.: A survey of numerical methods for optimal control. Adv. Astronaut. Sci. **135**(1), 497–528 (2009)
42. Tassa, Y., Mansard, N., Todorov, E.: Control-limited differential dynamic programming. In: Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong (2014)

43. Kelly, M.: An introduction to trajectory optimization: how to do your own direct collocation. SIAM Rev. **59**(4), 849–904 (2017)
44. Garg, D., Patterson, M., Hager, W.W., Rao, A.V., Benson, D.A., Huntington, G.T.: A unified framework for the numerical solution of optimal control problems using pseudospectral methods. Automatica **46**(11), 1843–1851 (2010)
45. Liu, Y., Ben-Tzvi, P.: Dynamic modeling, analysis, and design synthesis of a reduced complexity quadruped with a serpentine robotic tail. Integr. Comp. Biol. **61**(2), 464–477 (2021)
46. Bryson, A.E., Ho, Y.C.: Applied optimal control: optimization, estimation, and control. Taylor & Francis Group, New York (1975)
47. Betts, J.T.: Practical methods for optimal control and estimation using nonlinear programming, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2010)
48. Kuindersma, S., Deits, R., Fallon, M., Valenzuela, A., Dai, H., Permenter, F., Koolen, T., Marion, P., Tedrake, R.: Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. Auton. Robot. **40**(3), 429–455 (2016)
49. Mordatch, I., Todorov, E., Popović, Z.: Discovery of complex behaviors through contact-invariant optimization. ACM Trans. Graph. **31**(4), 1–8 (2012)
50. Posa, M., Cantu, C., Tedrake, R.: A direct method for trajectory optimization of rigid bodies through contact. Int. J. Robot. Res. **33**(1), 69–81 (2014)
51. Mastalli, C., Budhiraja, R., Merkt, W., Saurel, G., Hammoud, B., Naveau, M., Carpentier, J., Righetti, L., Vijayakumar, S., Mansard, N.: Crocoddyl: An efficient and versatile framework for multi-contact optimal control. In: Proceedings of the IEEE International Conference on Robotics and Automation, Paris (2020)
52. Chatzinikolaidis, I., Li, Z.: Trajectory optimization of contact-rich motions using implicit differential dynamic programming. IEEE Robot. Automat. Lett. **6**(2), 2626–2633 (2021)
53. Carpentier, J., Mansard, N.: Analytical derivatives of rigid body dynamics algorithms. In: Proceedings of the Robotics: Science and Systems, Pittsburgh, Pennsylvania (2018)
54. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press, New York (2004)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.