Robust Balancing Control of a Spring-legged Robot based on a High-order Sliding Mode Observer

Juan D. Gamba, Antonio C. Leite, Roy Featherstone

***Abstract*— This paper presents a simulation study of the balancing problem for a monopod robot in which the lower body (the leg) has been modified to include a passively spring-loaded prismatic joint. Such a mechanism can move by hopping but can also stand and balance on a single point. We aim to investigate the extent to which a balance controller can deal with the large values and rapid changes in the spring-damper forces, while controlling the absolute positions and orientations of its parts and balancing on one leg. It can be shown that a good performance is achieved if the spring-loaded joint is instrumented and calibrated so that its position and velocity, as well as the stiffness and damping coefficients, are considered when calculating the controller state variables. We also demonstrate the effectiveness of the balance controller by adding a high-order sliding mode (HOSM) observer based on the finite-time algorithm for robust parameter estimation of the stiffness and damping coefficients. The stability analysis and convergence proofs are presented based on the Lyapunov stability theory. Numerical simulations are included to illustrate the performance and feasibility of the proposed methodology.**

I. Introduction

In the last decades, the spring-loaded inverted pendulum (SLIP) mechanism has been widely applied in the biomechanical analysis of lower limb prostheses since it may reproduce the kinematic pattern of the human swing leg motion [1]. In this context, this paper is part of a series on the design, build and demonstration of an experimental high-performance 3D hopping and balancing robot, called Skippy [2], which uses springs to help it achieve higher maximum speed at launch; shock reduction on landing; and recycling mechanical energy from one hop to another.

The original balance theory has already been tested on different legged robots to balance [3], hop [4], and even walk along a line (ninja walk) [5] by controlling separately the walking and balancing actions. This balance theory, however, cannot be directly applied to a scenario where more complex mechanisms with passive joints are required to balance and track a given trajectory because the null-space projection matrix depends on all joint variables together, not only the driven ones [6].

In this paper we investigate to what extent a planar double inverted pendulum (DIP) system, in which the lower body is split into a pair of bodies (leg and foot) connected by a passively spring-loaded prismatic joint (see Fig. 2) [7], is able to successfully land and balance after suffering a 50% loss of spring stiffness caused by mechanical damage to the spring when the foot hits the ground. Moreover, the robustness and stability of the balance controller described in [6] are evaluated by performing a fast sequence of motions where the legged robot is capable of tracking a trajectory while balancing around a support point.

The decision to study only a planar system, rather than a fully 3D one, can be justified by observing that hopping is a nearly planar activity, so it makes sense to design the robot so that the hopping movement, which also includes the spring harmonic motion, is planar. This work aims to achieve three main goals by developing a novel and improved balance controller: (i) to demonstrate the ability of a spring-loaded DIP system to perform high-speed motions without losing its balance; (ii) to indirectly control the angular position between the foot and the ground by using null-space motions, relaxing the usual assumption of controlling the robot's movements based only on the motor-actuated joints; (iii) to guarantee the stability and full operation of the SLIP mechanism even when its spring loses part of its elastic capacity due to degradation, failures, or partial ruptures.

Legged robots can be able to move faster or get over obstacles by controlling their balance and jumping skills. Such a control strategy should produce fast motions while keeping, recovering or purposely losing the robot's balance. The well-known one-point balancing solutions [8], [9], [10] are not fast enough for producing such motions, while [11], [12] show better performance by adding special features such as reaction wheels. The first hopping machine [13] had an actuated prismatic leg with a springy behavior due to the pneumatic actuator. However, such a device could only hop, and balancing was out of the scope. A recent robot, named Salto [14], resembles a miniature, electrically actuated Raibert-style hopping machine, having an explicit series elastic element to provide the springiness in the leg. Salto has shown its ability to balance as well as hop [4] using the balance controller in [6]. However, Salto's upper link is a reaction wheel, which means that balancing has a negligible effect on the spring, and vice-versa. The balance controller presented in [15] can drive a monopod robot through fast motions without compensating the system dynamics with a reasonable error when tracking a reference trajectory. This performance can be improved by considering the system dynamics when obtaining the actuated torque [6]. However, the presence of parametric uncertainties degrades the system's efficiency and may cause instability. Thus, an observer is useful to estimate any given system uncertainty.

Juan D. Gamba and Roy Featherstone are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia, Genoa, Italy. Email: juan.gamba@iit.it, roy.featherstone@ieee.org.

Juan D. Gamba is also with Dibris, University of Genoa, Italy.

Antonio C. Leite is with the Faculty of Science and Technology, Norwegian University of Life Science, 1430 Ås, Norway. Email: acleite@ieee.org

The parameter identification theory deals with the problem of efficiently extracting data about the system dynamics from its measurements. Most of these strategies involve mainly least-squares (LS) methods, Bayes methods, Kalman filter extensions, among others. Finite-time algorithms have also demonstrated their effectiveness by identifying mechanical parameters in combination with a recursive LS algorithm where the design of the non-linear injection terms is based on the generalized super-twisting algorithm (STA) [16], leading to finite-time convergence [17], [18]. In [19] the authors use STA and a non-recursive LS algorithm to identify constant parameters in nonlinear systems [20]. In general, the finite-time convergence is based on the adaptive control theory, requiring to solve matrix-valued ordinary differential equations and check the invertibility property of a matrix online [21]. This scheme allows the reconstruction of the unknown parameters in finite-time (FT) provided that a given persistence of excitation (PE) condition holds [21]. A well-known approach used to ensure the PE condition in adaptive controllers is to add a bounded perturbation signal to the set-point or trajectory, or even use it as the reference input, which in contrast may degrade the specified regulation or tracking performance [19]. Here, the PE condition can be guaranteed by performing balance and trajectory tracking tasks.

There is an important distinction between the study presented here and those presented in [6], [22] on the topic of balancing in the presence of other motions. In the earlier works, it was assumed that the other motions were produced by a motion controller that was executing a prescribed motion command that was independent of the actions of the balance controller, and that was known in advance. Then, the robot could lean in anticipation of balance disturbances that would be caused by executing the motion command. This implies that the other motions are actuated, controlled and known in advance. In contrast, we consider here a passive springy joint, which is neither actuated nor controlled, and which moves in response to the actions of the balance controller.

In this work, we employ a modified version of the existing balance controller [6] to achieve high performance in balancing and absolute angular position tracking tasks [23] for a spring-loaded monopod, rather than a rigid-leg one (Fig. 1). A high-order sliding mode (HOSM) observer algorithm is used to estimate the external force generated by a viscoelastic element (spring plus dashpot), located between the robot leg and foot, identifying its elastic and viscous coefficients. The stability analysis of the overall closed-loop system is demonstrated by using the Lyapunov stability theory. Numerical simulations are included to illustrate the performance and feasibility of the proposed control methodology.

II. Problem Formulation

Here, we address the balancing and hopping motion control problem for a spring-loaded inverted pendulum (SLIP) based robot leg, in the presence of uncertainties and external disturbances. We consider a real-world scenario where a given monopod robot suffers a failure in its embedded

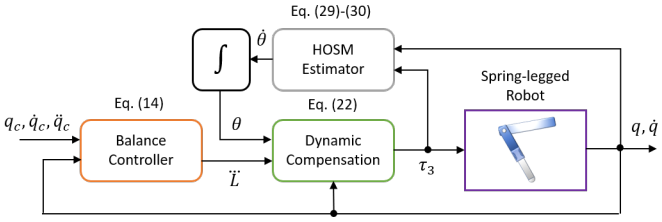

Fig. 1. Block diagram of the proposed balance control strategy.

array of spring-damper pairs during the landing phase. Some factors that cause viscoelastic elements to fail are: fatigue, inadequate materials, poor manufacturing processes, improper service and environmental effects. Then, using a high-order sliding mode observer, the robot can achieve a successful landing and balancing, while performing an efficient tracking control for its absolute angular position, even in the presence of such failure modes.

A. Single-leg robot models

At the tracking phase, we consider two single-leg robot models: one with a springy leg and the other with a rigid leg. The springy-leg robot is shown in Fig. 2, where the joint variables are q_1, q_2 and q_3 for joints 1, 2 and 3 respectively. The coordinates of the robot's centre of mass (CoM) relative to the support point at the ground are given by $c=(c_x, c_y)$. Notice that a stationary balance condition is given by $c_x=0$ and $\dot{q}_i = 0$ for $i = 1, 2, 3$. Here, since Joint 2 is a spring-damper prismatic joint, let us define the absolute orientation of the torso as $q_\mathrm{a} := q_1 + q_3$. Then, when all joints are zero, the leg is vertical, the leg's length is $l_1 + l_2$, and the torso is horizontal out to the right. The rigid-leg robot is obtained from the springy-leg robot by locking Joint 2 in the position it takes when the robot is balanced, the torso is horizontal (so $q_\mathrm{a} = 0$), and the spring is holding the weight of the upper leg and torso, which works out as $q_2 = -0.0282\,\mathrm{m}$.

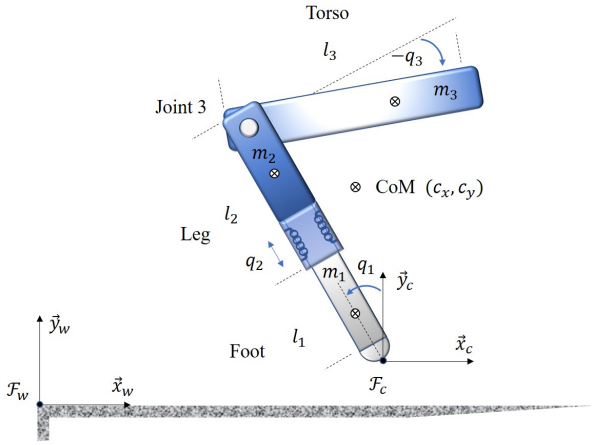

Fig. 2. Legged robot model: joint variables, lengths and masses.

The springy-leg robot is a planar, three-link mechanism in which Links 1, 2 and 3 are the lower leg (or foot), upper leg and torso, respectively. Joint 1 is a passive revolute joint that models the contact between the foot and ground; Joint 2 is a spring-loaded passive prismatic joint; and Joint 3 is the

actuated joint. The links' mass and length parameters are given as follows: $m_1 = 0.2\,\mathrm{kg}$, $m_2 = 0.5\,\mathrm{kg}$, $m_3 = 2.0\,\mathrm{kg}$, $l_1 = 0.2\,\mathrm{m}$, $l_2 = 0.3\,\mathrm{m}$, and $l_3 = 0.5\,\mathrm{m}$. These values were chosen to resemble the parameters of Skippy and Tippy [2], [3]. Each link i is modelled as a uniform thin rod, which means that the CoM is half way along the rod, and the rotational inertia about the CoM is $I_i = m_i\, l_i^2/12$ for $i = 1, 2, 3$. Positive motion of a revolute Joint i rotates link i counter-clockwise relative to link $i-1$; and positive motion of Joint 2 extends the leg, so that the actual length of the leg is $l_1 + l_2 + q_2$. In Fig. 2, joint variable q_1 is positive and q_3 is negative.

B. Simulation setup

Joint 2 is passively actuated by an array of two parallel and identical springs with stiffness and damping coefficients of $1000\,\mathrm{N\,m^{-1}}$ and $8\,\mathrm{N\,s\,m^{-1}}$, which results in an under-damped system. The total stiffness is appropriate for a legged robot to make small hops of around $1\,\mathrm{m}$ and is soft enough that Joint 2 moves significantly during landings and fast balancing movements. In other words, the stiffness is low enough to interfere significantly with the actions of the balance controller. The simulator uses an accurate model of compliant frictional contact between the foot and the ground that allows the foot both to slip and to lose contact with the ground. It resembles the contact model described in [15]. Contact forces acting on the foot along normal and tangential directions are given by:

$$F_y = \max(0, K_n\, z^{3/2} + D_n\, z^{1/2}\, \dot{z})\,, \quad (1)$$

$$F_x = \mathrm{clip}(K_t\, z^{1/2}\, u + D_t\, z^{1/2}\, \dot{u}\,, -\mu F_y\,, \mu F_y)\,, \quad (2)$$

where $z \in \mathbb{R}$ and $u \in \mathbb{R}$ are the ground compression and shear deformation, and $\mu > 0$ is the coefficient of friction. Moreover, K_n and K_t are the normal and tangential stiffness coefficients, D_n and D_t are the normal and tangential damping coefficients, all assumed to be positive. The function $\mathrm{clip}(u, \alpha, \beta) = \max(\min(u, \beta), \alpha)$ returns the value of u clipped to the specified range $[\alpha, \beta]$. The values of the parameters used in the simulations are: $K_t = 12.7 \times 10^6$ $D_t = 3.1 \times 10^5$, $\mu = 1$, $K_n = 8.5 \times 10^6$ and $D_n = 3.1 \times 10^5$, which are consistent with a hard floor and a high-friction hard rubber foot.

III. Balance Theory

In this section, the analysis presented in [6] is modified for the planar robot mechanism shown in Fig. 2. The upper joint (Joint 3) is commanded by the controller, the middle joint (Joint 2) is actuated by a spring-damper element, and the lower joint (Joint 1) represents the contact between the bottom of the lower link (the foot) and the ground, being therefore non-actuated. By representing the contact with a revolute joint, in the control design (but not in the simulation) we assume that the foot neither slips nor loses contact with the ground and that the movement of the contact point as the foot rotates can be neglected. The equation of dynamic motion for this mechanism is given by [6]:

$$\begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{12} & H_{22} & H_{23} \\ H_{13} & H_{23} & H_{33} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 0 \\ F_s \\ \tau_3 \end{bmatrix}, \quad (3)$$

where H_{ij} are the elements of the joint-space inertia matrix, $\ddot{q}_1$, $\ddot{q}_2$ and $\ddot{q}_3$ are the joint acceleration variables, C_1, C_2 and C_3 elements contain gravity, Coriolis and centrifugal forces. Here, τ_3 is the torque command at the actuated joint (Joint 3) and $F_s = -K_s\, q_2 - D_s\, \dot{q}_2$ is the force produced by the spring-damper system, with K_s and D_s being the stiffness and damping coefficients.

Since Joint 1 is non-actuated, it follows that gravity is the only force capable of exerting a moment about the support point and, hence, changing the angular momentum of the robot about this point. Let L be the total angular momentum of the leg robot about the support point, then we have

$$\dot{L} = -m\, g\, c_x\,, \quad (4)$$

where m is the total mass of the leg robot, g is the magnitude of gravitational acceleration (a positive number), and c_x is the horizontal coordinate of the CoM with respect to a point at the bottom of the foot, that is assumed to be the point that makes contact with the ground. Then, from (4) yields:

$$\ddot{L} = -m\, g\, \dot{c}_x\,, \qquad \dddot{L} = -m\, g\, \ddot{c}_x\,, \quad (5)$$

and following a special property of joint-space momentum, as proven in Appendix B of [6], we can obtain

$$L = H_{11}\, \dot{q}_1 + H_{12}\, \dot{q}_2 + H_{13}\, \dot{q}_3\,. \quad (6)$$

Notice that L and $\ddot{L}$ are linear functions of the joint velocities and x-axis CoM velocity respectively, while $\dot{L}$ is a constant multiple of x-axis CoM position. This implies that the condition $L = \ddot{L} = \dot{q}_2 = 0$ is equivalent to $\dot{q}_i = 0$ for $i = 1, 2, 3$ assuming linear independence. Then, the balance conditions for the springy-leg robot are simply given by $L = \dot{L} = \ddot{L} = 0$ [6], assuming that the passivity of Joint 2 ensures $\dot{q}_2 \to 0$. Note that any balance controller that successfully brings L to zero will ensure the robot's balance, but will not necessarily lead q_3 to the desired angle.

The next step is to add a fictitious joint between Joint 1 and the ground, which is a prismatic joint acting in the x-axis direction. We will call it Joint 0 so as not to disturb the numbering of the other three joints. Such a joint never moves, so it does not affect the robot's dynamics. However, the dimension of the robot motion equation increases, which now reads:

$$\begin{bmatrix} H_{00} & H_{01} & H_{02} & H_{03} \\ H_{01} & H_{11} & H_{12} & H_{13} \\ H_{02} & H_{12} & H_{22} & H_{23} \\ H_{03} & H_{13} & H_{23} & H_{33} \end{bmatrix} \begin{bmatrix} 0 \\ \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} \tau_0 \\ 0 \\ F_s \\ \tau_3 \end{bmatrix}, \quad (7)$$

where τ_0 is an external torque which assumes any value necessary to ensure that $\ddot{q}_0$ is equal to zero. The reason for adding Joint 0 into the dynamic model of the springy-leg robot (3) is that such a key idea provides us with an extra motion equation linking the joint space dynamics with the

motions of the CoM. In this way, we transform an under-actuated system into a virtually fully-actuated system. Then, let p_0 be the linear momentum of the robot along the x-axis direction given by

$$p_0 = m\,\dot{c}_x = H_{01}\,\dot{q}_1 + H_{02}\,\dot{q}_2 + H_{03}\,\dot{q}_3 = -\ddot{L}/g\,, \quad (8)$$

and τ_0 be the external torque, due to the ground reaction force acting on the robot, given by

$$\tau_0 = m\,\ddot{c}_x = -\dddot{L}/g\,. \quad (9)$$

Then, combining (6) and (8), and neglecting the contribution of the Joint 2 for the linear and angular momentum (i.e., $\dot{q}_2 \approx 0$) results in:

$$\begin{bmatrix} L \\ \ddot{L} \end{bmatrix} = \begin{bmatrix} H_{11} & H_{13} \\ -g\,H_{01} & -g\,H_{03} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_3 \end{bmatrix}. \quad (10)$$

Then, to control the absolute angular position $q_\mathrm{a} = q_1 + q_3$ we use the following mapping

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_\mathrm{a} \end{bmatrix}, \quad (11)$$

and the linear relationship (10) can be solved as:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_\mathrm{a} \end{bmatrix} = \frac{1}{gD} \begin{bmatrix} -gH_{03} & -H_{13} \\ g(H_{01}-H_{03}) & H_{11}-H_{13} \end{bmatrix} \begin{bmatrix} L \\ \ddot{L} \end{bmatrix}, \quad (12)$$

where $D = H_{13}H_{01} - H_{11}H_{03}$. This equation requires $D \neq 0$, which holds in any configuration in which the robot is physically capable of balancing itself [6]. So, $\dot{q}_\mathrm{a}$ can be expressed as:

$$\dot{q}_\mathrm{a} = Y_1\,L + Y_2\,\ddot{L}\,, \quad (13)$$

where $Y_1 = (H_{01} - H_{03})/D$ and $Y_2 = (H_{11} - H_{13})/g\,D$. It is worth noticing that Y_1 and Y_2 vary with joint configuration and can be expressed as simple functions of two physical properties of the robot: (i) the time constant of toppling T_c, which measures how quickly the robot falls if the balance controller fails or simply does not act; (ii) the velocity gain G_v, which measures the effect on the CoM velocity for a unitary change in the velocity of the actuated joint [6]. The time constant T_c is an explicit parameter of the acausal filter to be presented in what follows.

IV. Robot Balance Controller

The goal of the balance controller is to calculate a suitable value for $\dddot{L}$ to ensure that q_a follows a given desired signal q_c, while maintaining the robot's balance (Fig. 3). A state feedback-based controller is given by [6]:

$$\dddot{L} := k_{dd}\,\ddot{L} + k_d\,\dot{L} + k_L\,L + k_q\,(q_\mathrm{a} - u)\,, \quad (14)$$

where u is the input signal, typically a position command, to be designed (see below). The feedback gains are obtained via pole placement technique as:

$$k_{dd} = -a_3\,, \qquad k_d = -a_2 + a_0(Y_2/Y_1)\,, \quad (15)$$

$$k_L = -a_1\,, \qquad k_q = -a_0/Y_1\,, \quad (16)$$

with

$$a_0 = \lambda_1\lambda_2\lambda_3\lambda_4\,, \quad (17)$$

$$a_1 = -\lambda_1\lambda_2(\lambda_3+\lambda_4) - \lambda_3\lambda_4(\lambda_1+\lambda_2)\,, \quad (18)$$

$$a_2 = \lambda_1(\lambda_2+\lambda_3+\lambda_4) + \lambda_4(\lambda_2+\lambda_3) + \lambda_2\lambda_3\,, \quad (19)$$

$$a_3 = -\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4\,, \quad (20)$$

where λ_1, λ_2, λ_3 and λ_4 are the desired values of the roots of the corresponding closed-loop characteristic polynomial $\Delta(\lambda) = \lambda^4 + a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0$. It can be shown that if $Y_1 = 0$ then $\lambda = 0$ is always a root of $\Delta(\lambda)$ regardless of the choice of gains and, thus, $Y_1 \neq 0$ is a stability condition [6]. Here, λ_1 is chosen to be the closed-loop bandwidth of the controller, while λ_2 and λ_3 are destined to be cancelled by two introduced zeros, mentioned below, and λ_4 is chosen to be $-1/T_\mathrm{c}$ in order to cancel a natural zero in the transfer function. The input signal u for the control law (14) is computed from the desired signal q_c and its time-derivatives, $\dot{q}_\mathrm{c}$ and $\ddot{q}_\mathrm{c}$, according to:

$$u = \mathrm{AF}\{\, q_\mathrm{c} - (\lambda_2+\lambda_3)\,\dot{q}_\mathrm{c} + (\lambda_2\lambda_3)\,\ddot{q}_\mathrm{c}\,\}\,. \quad (21)$$

Notice that the formula (21) is designed to provide two key features: (i) to introduce two zeros into the closed-loop transfer function $G_f(s)$ at the vicinity of λ_2 and λ_3, which may cancel or attenuate the effect of the corresponding poles; (ii) to apply an acausal filter, denoted by $\mathrm{AF}\{\cdot\}$, to the desired signal which makes the robot lean in anticipation of the balance disturbances that will be caused by future commanded motions [6]. In particular, AF creates a first-order low-pass filter with time constant T_c, that runs backwards in time from a point far enough in the future back to the present. To implement this acausal filter, the controller needs to know the expected short-term future value of q_c. In general, such an information can be provided by the robot's high-level controller, which usually interacts with task planning algorithms and, therefore, knows what movements the robot intends to perform next [6]. It is worth mentioning that $\dddot{L}$ is the control signal computed by (14), but the output of the control system has to be either a torque command τ_3 or an acceleration command $\ddot{q}_3$ for Joint 3. Then, substituting (9) into (7), and assuming that K_s and D_s are both uncertain, we can rearrange the unknowns into a single vector yielding:

$$\begin{bmatrix} 0 & H_{01} & H_{02} & H_{03} \\ 0 & H_{11} & H_{12} & H_{13} \\ 0 & H_{21} & H_{22} & H_{23} \\ -1 & H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} \tau_3 \\ \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \begin{bmatrix} -\dddot{L}/g - C_0 \\ -C_1 \\ \hat{F}_s - C_2 \\ -C_3 \end{bmatrix}, \quad (22)$$

where $\hat{F}_s$ is the estimated spring-damper force; and the equation can be solved for both τ_3 and $\ddot{q}_3$. Now, let us apply the acausal filter $\mathrm{AF}\{\cdot\}$ to the desired signal q_c such that $q_\mathrm{f} = \mathrm{AF}\{q_\mathrm{c}\}$ and so on, such that:

$$q_\mathrm{f}(s) = \frac{1}{1 - T_\mathrm{c}s}\,q_\mathrm{c}(s)\,. \quad (23)$$

For a given control law (14), with gains (15) and (16) under (17)-(20), and the input signal (21), it can be shown that the

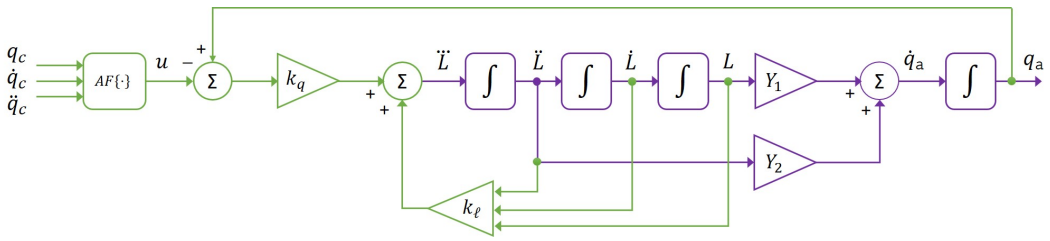

Fig. 3. Block diagram for the balancing control problem of the springy-leg robot, with $k_\ell = \text{diag}(k_{dd}, k_d, k_L)$. The green lines indicate the balance controller, and the purple lines denote the plant.

complete transfer function from q_c to q_a would be:

$$q_a(s) = \frac{a_0(1+T_c s)(1+\alpha_1 s+\alpha_2 s^2)}{s^4+a_3 s^3+a_2 s^2+a_1 s+a_0}\, q_c(s)\,, \tag{24}$$

where $\alpha_1 = -(\lambda_2+\lambda_3)$ and $\alpha_2 = \lambda_2\lambda_3$. Then, by setting one of the poles to be $-1/T_c$, the mechanism-dependent zero can be removed, and the transfer function simplifies to three poles and up to two zeros, all of them freely tunable by the control system designer.

In the next section, we propose the use of a state estimator (or observer) to provide an accurate estimate of the internal force F_s, according to (22), by identifying the stiffness and damping coefficients, K_s and D_s, of the viscoelastic element located at the Joint 2, while the springy-leg robot is balancing and tracking a given position desired signal.

V. Parameter Identification via High-order Sliding Mode (HOSM) Observer

The lack of knowledge, or even the existence of uncertainties in the system's parameters, is a common obstacle in the design of a model-based controller capable of successfully executing a given task. Now, consider the practical case in which one of the pairs of the spring-damper system fails due to the rupture of the spring in the landing phase and the internal force F_s drops down (e.g., 50%), affecting the robot's ability to balance and hop. Then, we propose a high-order sliding mode (HOSM) observer to ensure a successful landing, balancing, and steering for the springy-leg robot while it estimates the stiffness and damping coefficients of the viscoelastic element. It is worth noticing that the controller introduced in [24] can be applied to the proposed control problem given by (22) for similar balancing and trajectory tracking tasks. However, less accuracy is expected since the torque is directly computed from the control law without compensating the robot dynamics.

Here, let us introduce the following notation: (i) H and C denote respectively the inertia matrix and the gravity, Coriolis and centrifugal, and friction forces for the springy-leg robot, similar to (3); (ii) q_x and q_y are extra prismatic joints used to describe the position of the robot's foot with respect to the world frame; (iii) positions, velocities and accelerations of the robot are described by the vectors q, $\dot{q}$ and $\ddot{q}$ respectively, where $q = \begin{bmatrix} q_x & q_y & q_1 & q_2 & q_3 \end{bmatrix}^\top$ and so on; (iv) the vector $y = \begin{bmatrix} q_x & q_y & q_1 & q_2 & q_a \end{bmatrix}^\top$ satisfies $\dot{q} = T\,\dot{y}$ and $\ddot{q} = T\,\ddot{y}$, similar to (11); (v) $\tau = \begin{bmatrix} 0 & 0 & 0 & 0 & \tau_3 \end{bmatrix}^\top$ denotes the torque produced by the balance controller; (vi) $F = \begin{bmatrix} 0 & 0 & 0 & F_s & 0 \end{bmatrix}$ denotes the internal force generated by the spring-damper element; (vii) for a given matrix $Q \in \mathbb{R}^{m\times n}$, the pseudo-inverse and transpose matrices of Q are denoted by $Q^\dagger$ and $Q^\top$ respectively, the induced norm is given by $\|Q\| = \sqrt{\lambda_{max}(Q^\top Q)}$, where λ_{max} is the maximum eigenvalue; (viii) $\lceil\cdot\rfloor^\gamma = |\cdot|^\gamma\,\text{sgn}(\cdot)$ for $\gamma > 0$, where $\text{sgn}(\cdot)$ is the sign function.

Then, similarly to (7) and using the aforementioned notation, the motion dynamics for the springy-leg robot can be rewritten simply as [6]:

$$\ddot{y} = (HT)^{-1}\left[\tau + F - C\right]\,, \tag{25}$$

or, in a more detailed manner, as:

$$\ddot{y} = (HT)^{-1}\left[\tau - C\right] + (HT)^{-1}\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -q_2 & -\dot{q}_2 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} K_s \\ D_s \end{bmatrix}\,, \tag{26}$$

which may be written more compactly as

$$\ddot{y} = f(q,\dot{q},\tau_3) + \Gamma(q,\dot{q})\,\theta\,, \tag{27}$$

where $f(\cdot,\cdot,\cdot)$ is a nonlinear matrix function, $\Gamma(\cdot,\cdot)$ is the regressor matrix and $\theta = \begin{bmatrix} K_s & D_s \end{bmatrix}^\top$ denotes the vector of uncertain parameters to be identified. Here, without loss of generality, we assume that $\Gamma(\cdot,\cdot)$ is measurable and bounded in norm i.e., $0 < \Gamma_m \leq \|\,\Gamma(q,\dot{q})\,\| \leq \Gamma_M < \infty$ for $\forall q, \dot{q}$, and satisfies the persistent excitation (PE) condition [25]. Selecting the last coordinate of the acceleration vector $\ddot{y}$ in (27) we obtain:

$$\ddot{q}_a = f_l(q,\dot{q},\tau_3) + \Gamma_l(q,\dot{q})\,\theta\,, \tag{28}$$

where $f_l(\cdot,\cdot,\cdot)$ is a nonlinear scalar function obtained from the last row of $f(\cdot,\cdot,\cdot)$ and $\Gamma_l = \begin{bmatrix} \Gamma_{l1} & \Gamma_{l2} \end{bmatrix}$ is a regressor vector obtained from the last row of the regressor matrix $\Gamma(\cdot,\cdot)$. Here, the objective is to estimate θ using a nonlinear parameter identification algorithm provided that $\dot{q}_a$ and Γ_l are both measurable from the system signals. Then, let us briefly address the high-order sliding mode (HOSM) observer approach based on the finite-time algorithm and briefly describe its convergence properties. A complete explanation for the key features of the HOSM observer approach can be found in [20], [26].

According to (28), a finite-time parameter identification algorithm can be described by:

$$\dddot{\hat{q}}_a = -k_1\phi(e) + \Gamma_l(q,\dot{q})\,\hat{\theta} + f_l(q,\dot{q},\tau_3)\,, \tag{29}$$

$$\dot{\hat{\theta}} = -\alpha(e)\begin{bmatrix} k_2 & 0 \\ 0 & k_3 \end{bmatrix}\Gamma_l^\top(q,\dot{q})\,, \tag{30}$$

where k_1, k_2, k_3 are positive gains, $e = \dot{\hat{q}}_\mathrm{a} - \dot{q}_\mathrm{a}$ is the parameter estimation error where $\dot{q}_\mathrm{a}$ is obtained from the sum of joint velocities $\dot{q}_1$ and $\dot{q}_3$ (both assumed to be measurable), $\dot{\hat{q}}_\mathrm{a}$ is the absolute angular velocity estimated by the super-twisting or the fixed-time observer algorithms [20], $\tilde{\theta} = \hat{\theta} - \theta$ is the parameter identification error and $\hat{\theta}$ is the vector of uncertain parameters to be identified by using the gradient-type adaptation law given by (30). Finally, $\phi(e)$ and $\alpha(e)$ are nonlinear functions of the parameter estimation error e to be defined in what follows:

$$\phi(e) = \lceil e \rfloor^{1/2} + e\,, \tag{31}$$

$$\alpha(e) = \frac{1}{2}\lceil e \rfloor^{0} + \frac{3}{2}\lceil e \rfloor^{1/2} + e\,. \tag{32}$$

Now, we can state the following theorem about the convergence properties of the finite-time algorithm.

Theorem 1: Consider the dynamic system described by (28) and assume that the regressor vector $\Gamma_l(q, \dot{q})$ is measurable and satisfies the PE condition. Then, the finite-time parameter identification algorithm (29)-(30) with (31)-(32) ensures the uniformly finite-time stability property of the system errors, and the following properties hold: (i) $\lim_{t\to\infty} e(t) = 0$; (ii) $\lim_{t\to\infty} \tilde{\theta}(t) = 0$.

Proof: Let the system error dynamics be written as follows: $\dot{e} = -k_1\,\phi(e) + \Gamma_l\,\tilde{\theta}$ and $\dot{\tilde{\theta}} = -k_2\,\alpha(e)\,\Gamma_l^\top$, and define the state vector $\zeta = [\,\phi(e)\;\; \theta^\top\,]^\top$. Now, consider the following Lyapunov function candidate: $V(t, e, \tilde{\theta}) = \zeta^\top P(t)\,\zeta$ where $P(t)$ is a symmetric, bounded and positive definite matrix satisfying $\dot{P}(t) = -\phi'(e)[P(t)A(t) + A^\top(t)P(t) + Q(t)]$. The function $V(t, e, \tilde{\theta})$ is continuous and continuously differentiable everywhere in $\mathbb{R}^3$, except on the set $\Omega = \{(e, \tilde{\theta}) \in \mathbb{R}^3 \,|\, e = 0\}$, where it is not Lipschitz continuous and $\dot{V}$ is not well defined for the partial derivative $\phi'(e)$. Notice that when $e = 0$ in isolated points it implies that $\alpha(e) = \phi'(e)\phi(e)$ and $\dot{\zeta} = \phi'(e)A(t)\zeta$. Then, the time-derivative of V takes the form: $\dot{V}(t, e, \tilde{\theta}) = \zeta^\top[\,\dot{P}(t) + \phi'(e)P(t)A(t) + \phi'(e)A(t)^\top P(t)\,]\,\zeta$, at the points where V is differentiable, which implies that $\dot{V}(t, e, \tilde{\theta}) = -\phi'(e)\zeta^\top Q(t)\zeta = -1/2\,\left(|e|^{-1/2} + 2\right)\zeta^\top Q(t)\zeta$. Notice that $|e_1|^{\frac{1}{2}} \leq \|\zeta\|_2$, with $\|\zeta\|_2^2 = \zeta^\top\zeta = |e| + 2|e|^{\frac{3}{2}} + e^2 + \|\tilde{\theta}\|_2^2$ as the Euclidean norm of ζ. Thus, $\dot{V}$ may be upper bounded as follows: $\dot{V}(t, e, \tilde{\theta}) \leq -c_3\,(1/2)(\|\zeta\|_2 + 2\|\zeta\|_2^2)$. This expression holds when the trajectories of system errors are outside of the set Ω, and it indicates that $\dot{V} < 0$ in the complement of the set Ω. To remain on the set Ω for a given time interval $t \in [t, t + T_0]$, it is necessary that $e(t) = 0$ and $\Gamma_l\,\tilde{\theta}(t) = 0$ during that interval, which violates the persistent excitation (PE) condition. Then, it follows from Zubov's theorem [25] that the origin is asymptotically stable. For a more detailed proof, please consult [20], [26]. ■

VI. Simulation Results

In this section, we present the simulation results of an experimental case in which the springy-leg robot lands with a velocity of $-5\,\mathrm{m\,s^{-1}}$ along the y-axis. This is similar to falling from a height of $1.3\,\mathrm{m}$ from the foot to the ground. At the touch-down, one of the springs breaks when q_2 is compressed in $70\,\mathrm{mm}$ dropping the spring array's force by 50% percent. Next, q_2 reaches the maximum compression at $164\,\mathrm{mm}$ due to the low stiffness in the spring's array, producing a small hop after all the elastic energy is released, and lifting the robot $0.45\,\mathrm{m}$ from the foot to the ground. When the robot lands for a second time the spring compresses $62\,\mathrm{mm}$, and the balance controller is able to stabilize the system after a small bounce, and then executes a motion command consisting of a sequence of ramps and sinusoids. The estimator and the balance controller work collectively during the whole experiment, and the dynamic simulation of the springy-leg robot can be seen in the accompanying video clip.

In this simulation, we use the *ode23t* solver from MATLAB with relative tolerance set to 10^{-6} and other parameters chosen at their default values. Although the balance controller assumes that the foot never loses contact nor slips with the ground, in the simulation we have included the ground-contact model presented in Section II-B.

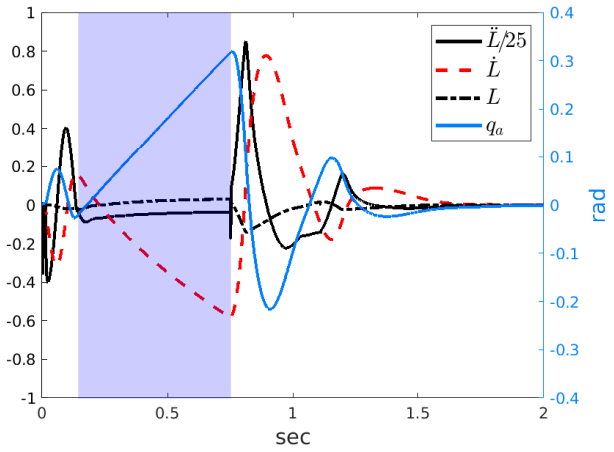

Fig. 4. Evolution of the controller's state variables from the moment of landing until the robot has settled. The left-side scale applies to $\ddot{L}$, $\dot{L}$ and L, and the right-side scale applies to q_a. The shaded area marks the flight phase of the small hop.

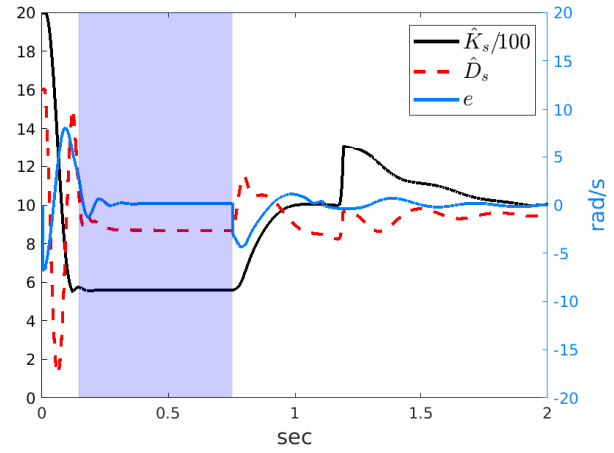

Fig. 5. Evolution of the estimator's parameters from the moment of landing until the robot has settled. The left-side scale applies to $\hat{K}_s$ and $\hat{D}_s$, and the right-side scale applies to e.

The balance controller and the online estimator are switched on since the beginning of the simulation. The behavior over time of the controller's state variables and estimation variables are depicted respectively in Fig. 4 and Fig. 5. The poles of the controller are $\lambda_1 = -20$, $\lambda_2 = -20$, $\lambda_3 = -20$, and $\lambda_4 = -1/T_c$, all expressed $\mathrm{rad\,s^{-1}}$. The gains of the estimator are: $k_1 = 10.83$, chosen so that it ensures the persistent excitation condition, $k_2 = 9.9 \times 10^3$ and $k_3 = 18.26$, are selected according to the magnitude of θ and the desired convergence rate of $\hat{\theta}$. From Fig. 4, it can be seen that q_a is initially pushed to about $0.08\ \mathrm{rad}$ by the momentum of the landing before swinging slightly negative. Next, it rises to nearly $0.32\ \mathrm{rad}$ during the small hop (shaded area) and then pushed back to approximately $-0.21\ \mathrm{rad}$ at the second landing. Finally, after a small bounce q_a is driven close to zero. Referring to Fig. 5, the estimated parameters $\tilde{\theta}$ and e are initialized to zero at the beginning of the simulation when $K_s = 2000\ \mathrm{N\,m^{-1}}$ and $D_s = 16\ \mathrm{N\,s\,m^{-1}}$ at the spring's array. The failure occurs at $t = 0.0114\ \mathrm{s}$ when the q_2 compresses to $70\ \mathrm{mm}$. After the rupture, the force produced by the spring's array drops 50% to $K_s = 1000\ \mathrm{N\,m^{-1}}$ and $D_s = 8\ \mathrm{N\,s\,m^{-1}}$, the estimated $\hat{K}_s$ decreases to $550\ \mathrm{N\,m^{-1}}$ and creates a small overshoot just before the small hop (shaded area). On the other hand, $\hat{D}_s$ oscillates to 1.274, 14.93 and $10.9\ \mathrm{N\,s\,m^{-1}}$ before the small hop. This oscillatory behavior may arise due to the lack of persistent excitation of $\dot{q}_a$ between the rupture point and the small hop (before the shaded area). q_2 only shrinks and expands once during this period. Notice that $\hat{\theta}$ tends to drift away when the persistent excitation ($\dot{q}_a \approx 0$) is absent or very small, and the system continues moving due to the elastic energy on the spring (spike drift between $1.15\ \mathrm{s}$ and $1.25\ \mathrm{s}$ in Fig. 5). Then, the robot's foot loses contact with the ground (small hop) between $t = 0.1456\ \mathrm{s}$ and $t = 0.7510\ \mathrm{s}$, and reaches the maximum height of $0.45\ \mathrm{m}$ from the foot to the ground. This flight period is shown shaded in both Fig. 4 and Fig. 5. The balance controller does not know that the foot has left the ground, and continues using the model described in Section III, which assumes that the robot's foot is always on the ground. On the other hand, the observer knows the position and velocity of the whole system to the world frame. This flight period does not significantly affect the controller's and estimator's stability, and the robot has come to rest within about $1\ \mathrm{s}$ after the second landing. Notice that the balance controller is assumed to know the orientation of the robot at all times, e.g., using an onboard IMU, so it always knows the correct values of q_1 and $\dot{q}_1$.

At the time of $2.011\ \mathrm{s}$ the desired signal q_c switches from zero to the sequence of ramps and sinusoids mentioned before. This portion of the action sequence is plotted in Fig. 6. The desired sequence consists of three ramps between 0 and $0.8\ \mathrm{rad}$, all at a speed of $3\ \mathrm{rad\,s^{-1}}$, followed by a long ramp from $0.8\ \mathrm{rad}$ to $-0.6\ \mathrm{rad}$, followed by three more ramps at speeds of $-2.5\,\mathrm{rads^{-1}}$, $2.5\,\mathrm{rads^{-1}}$ and $-1\,\mathrm{rads^{-1}}$, followed by three cycles of a sine wave at $1\ \mathrm{Hz}$. Notice that, such a desired sequence asks the robot to make large and fast motions. Two zeros are introduced to the desired signal at $-20\ \mathrm{rad\,s^{-1}}$, as mentioned in (21). For this springy-leg robot, T_c varies between $0.1562\ \mathrm{s}$ at $q_a = -0.6\ \mathrm{rad}$ and $0.2507\ \mathrm{s}$ at $q_a = 0.8\ \mathrm{rad}$ in this action sequence. The behavior of the signals is shown in Fig. 6.

The rigid-leg response is obtained from a separate simulation in which the initial conditions are set as close as possible to the actual state of the springy-leg robot at time $2.011\ \mathrm{s}$. At the landing phase, the spring reached a compression peak of about $16.7\,\mathrm{cm}$ which is 33.4% of the leg length. During the tracking task, the spring compression varied in a range of $5.5\,\mathrm{cm}$, which is 11% of the leg length.

In Fig. 6, it can be seen that there are some significant tracking errors, particularly at the beginning and end of the ramps. Three main factors contribute to such errors: (i) the balance controller assumes that the plant in Fig. 3 is linear when in reality it is not; (ii) for simplicity, the acausal filter uses values of T_c calculated for a balanced configuration at each instant instead of a leaning configuration; (iii) at each of the balance configurations used to calculate T_c, it assumes that q_2 is constant for the whole tracking when in fact it oscillates around 10% and reaches a peak of 20%, causing tracking inaccuracies as shown in Figures 6 and 7. Then, in this experimental case, we can conclude that the balance controller combined with an HOSM observer has accomplished three tasks simultaneously using only a single actuator: balance the robot, follow the commanded signal, and suppress vibrations, even in the presence of uncertainties in the stiffness and damping coefficients of the viscoelastic element.

VII. Concluding Remarks

This work has studied to what degree a springy leg may affect the performance of the balance controller [6], combined to an HOSM observer [26], when a failure occurs in the spring-loaded device which loses 50% of its stiffness due to a spring fracture. This strategy aims to keep the robot in balance in the presence of large and fast motions, as well as parametric uncertainties. One of the purposes of this work was also to investigate the effectiveness of a monopod robot that uses a compliant leg to help it to perform more complex motions (e.g., higher hops, somersaults) without losing the balance capability achieved in previous works with rigid-leg robots. It was shown that the balance controller can cope with large, fast motions that occur during landing, including small bounces (Figures 4 and 5). Moreover, it can track large, fast motion commands even with a low-noise spring force estimation after landing, with almost the same accuracy that could be achieved if the leg was rigid (Fig. 6). This study also showed the possibility of performing an absolute motion tracking, not only controlling the angle of the actuated joint q_3 but also the angle of the passive joint q_1 by introducing a state mapping for the balance controller given as (11). Finally, it was also shown that the balance controller can suppress vibrations caused by the spring-loaded device. In future works, we intend to design a robust balance controller based on [6], and investigate the effects of a series elastic element in the actuated joint.

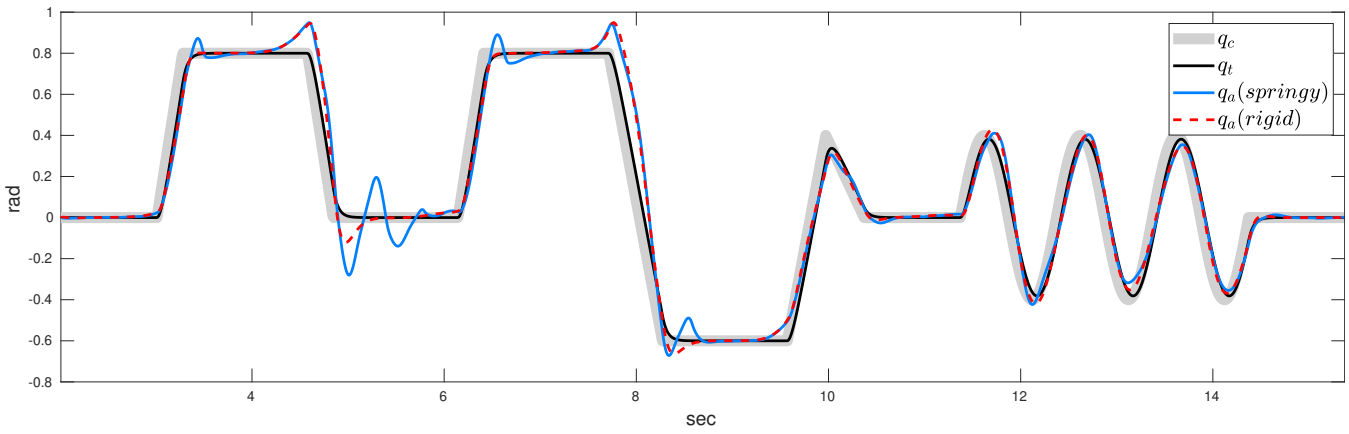

Fig. 6. Absolute motion tracking performance of the balance controller. q_c is the original desired signal (Eq. (21)), q_t denotes the theoretical response of the balance controller if the plant was really linear, $q_a\,(springy)$ is the actual response of the absolute joint on the springy-leg robot, and $q_a\,(rigid)$ denotes the actual response on the rigid-leg robot.

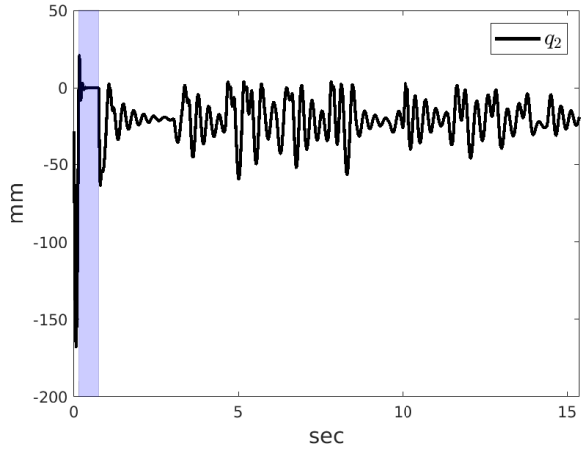

Fig. 7. Motion of the spring during landing and tracking of the desired signal in Fig. 6.

REFERENCES

[1] S. Kajita and C. Ott, "Limbed systems," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2016, pp. 419–442.

[2] R. Featherstone, "The Skippy Project," http://royfeatherstone.org/skippy, 2020, last access: Oct. 3rd 2020.

[3] J. J. M. Driessen, A. E. Gkikakis, R. Featherstone, and B. R. P. Singh, "Experimental Demonstration of High-Performance Robotic Balancing," in *2019 International Conference on Robotics and Automation*, Montreal, Canada, May 20-24 2019, pp. 9459–9465.

[4] J. K. Yim, B. R. P. Singh, E. K. Wang, R. Featherstone, and R. S. Fearing, "Precision Robotic Leaping and Landing Using Stance-Phase Balance," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3422–3429, 2020.

[5] C. Gonzalez, V. Barasuol, M. Frigerio, R. Featherstone, D. G. Caldwell, and C. Semini, "Line Walking and Balancing for Legged Robots with Point Feet," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, USA, 2020, pp. 3649–3656.

[6] R. Featherstone, "A Simple Model of Balancing in the Plane and a Simple Preview Balance Controller," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1489–1507, 2017.

[7] J. D. Gamba and R. Featherstone, "Balancing on a Springy Leg," in *2021 IEEE International Conference on Robotics and Automation*, Xi'an, China, June 01-03 2021.

[8] M. D. Berkemeier and R. S. Fearing, "Tracking Fast Inverted Trajectories of the Underactuated Acrobot," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, pp. 740–750, 1999.

[9] J. W. Grizzle, C. H. Moog, and C. Chevallereau, "Nonlinear Control of Mechanical Systems with an Unactuated Cyclic Variable," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 559–576, 2005.

[10] M. W. Spong, "The Swing up Control Problem for the Acrobot," *IEEE Control Systems Magazine*, vol. 15, no. 1, pp. 49–55, 1995.

[11] M. Gajamohan, M. Merz, I. Thommen, and R. D'Andrea, "The Cubli: A Cube that can Jump up and Balance," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 7–12 2012, pp. 3722–3727.

[12] Murata Manufacturing Co., Ltd., "Murata BOY's Capabilities," http://www.murata.com/en-global/about/mboymgirl/mboy/capabilities, 2020, last access. Oct. 3rd 2020.

[13] M. Raibert, *Legged Robots That Balance*. Cambridge, MA, USA: MIT Press, 1986.

[14] D. W. Haldane, M. M. Plecnik, J. K. Yim, and R. Fearing, "Robotic Vertical Jumping Agility via Series-elastic Power Modulation," *Science Robotics*, vol. 1, pp. 1–9, Dec. 2016.

[15] M. Azad and R. Featherstone, "Angular Momentum based Balance Controller for an Under-actuated Planar Robot," *Autonomous Robot*, vol. 40, pp. 93–107, 2016.

[16] J. A. Moreno and M. Osorio, "Strict Lyapunov Functions for the Super-Twisting Algorithm," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 1035–1040, 2012.

[17] J. Davila, L. Fridman, and A. Poznyak, "Observation and Identification of Mechanical Systems via Second Order Sliding Modes," in *2006 International Workshop on Variable Structure Systems*, Alghero, Italy, Jun. 5–7 2006, pp. 232–237.

[18] N. K. M'sirdi, A. Rabhi, L. Fridman, J. Davila, and Y. Delanne, "Second Order sliding Mode Observer for Estimation of Velocities, Wheel Sleep, Radius and Stiffness," in *2006 American Control Conference*, Minneapolis MN, USA, Jun. 14–16 2006, pp. 3316–3321.

[19] V. Adetola and M. Guay, "Finite-Time Parameter Estimation in Adaptive Control of Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 807–811, 2008.

[20] O. Boubaker and R. Iriarte, Eds., *The Inverted Pendulum in Control Theory and Robotics: From Theory to New Innovations*. Institution of Engineering and Technology, 2017.

[21] A. Polyakov and L. Fridman, "Stability Notions and Lyapunov Functions for sliding Mode Control Systems," *Journal of the Franklin Institute*, vol. 351, no. 4, pp. 1831–1865, 2014.

[22] R. Featherstone, "Quantitative Measures of a Robot's Physical Ability to Balance," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1681–1696, 2016.

[23] B. R. P. Singh, "Angular momentum based balancing control and shock-proof design of legged robots," Ph.D. dissertation, Dept. Information Engineering, University of Pisa, Italy, 2021.

[24] M. Azad and R. Featherstone, "Balancing and Hopping Motion of a Planar Hopper with One Actuator," in *2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 6–10 2013, pp. 2027–2032.

[25] A. Poznyak, *Advanced Mathematical Tools for Control Engineers: Volume 1: Deterministic Systems*. Amsterdam-Boston: Elsevier Science, 2008.

[26] J. A. Moreno and E. Guzman, "A New Recursive Finite-Time Convergent Parameter Estimation Algorithm," in *18th IFAC World Congress*, Milan, Italy, Aug. 28 - Sep. 2 2011, pp. 3439–3444.