Balancing on a Springy Leg

Juan D. Gamba[1,2] Roy Featherstone[1]

Abstract— **This paper presents a simulation study of the problem of balancing a planar double pendulum in which the lower body (the leg) has been modified to include a spring-loaded passive prismatic joint. Robots of this kind can travel by hopping, and can also stand and balance on a single point. The purpose of this study is to investigate the degree to which a balance controller can cope with the large and rapidly changing forces from the spring. It is shown that good performance can be achieved using an existing balance controller if the spring-loaded joint is instrumented so that its position and velocity can be taken into account when calculating the state variables needed by the balance controller.**

I. Introduction

This paper is part of a series on the design and construction of an experimental high-performance monopedal robot, called Skippy [1], which uses springs to help it achieve high hops. A new balance controller has already been developed for Skippy, which can cope with the high-speed motions that Skippy is intended to exhibit [2], [3]; and this controller has already been demonstrated on a balancing-only precursor to Skippy, called Tippy [4]. However, the interaction between the balance controller and the springs has not yet been investigated, and that is the topic of this paper.

Specifically, this paper investigates the following problem. Given a planar inverted double pendulum in which the lower body (the leg) has been replaced with a pair of bodies (upper and lower leg) connected by a spring-loaded passive prismatic joint (see Fig. 1), how well does the balance controller described in [2], [3] cope with the spring, both in the immediate aftermath of a landing and after the landing transients have died away?

The decision to study only a planar system, rather than full 3D, 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 includes the spring) is planar.

The main result of this paper is that the balance controller in [2], [3] still works when the leg is springy, although the performance is not as good as with a rigid leg, but the position and velocity variables of the passive joint have to be taken into account when calculating the state variables needed by the balance controller.

The rest of this paper is organized as follows: first, a review of relevant previous works; then the robot model; then the theory of balancing used in this paper; then the balance controller; and finally the simulation experiments and their results.

[1]Dept. Advanced Robotics, Istituto Italiano di Tecnologia, Genoa, Italy
[2]Dept. Informatics, Bioengineering, Robotics and Systems Engineering (Dibris), University of Genoa, Italy

II. Previous Works

Hopping involves fast motions, so a balance controller must be able to make fast motions while maintaining, recovering, or deliberately losing its balance. The problem of robotic balancing on a point is known to be a solved problem, but most existing solutions, such as [5], [6], [7], are not fast enough for our application. More recent results, such as [8], [9], show better performance, but require the robot to have special features, such as reaction wheels.

Raibert's original hopping machines [10] had an actuated prismatic leg that was springy because of the use of a pneumatic actuator; but these machines could only hop, so balancing (without hopping) was not an issue. A more recent robot, Salto [11], resembles a miniature, electrically actuated Raibert-style hopping machine having an explicit series elastic element to provide the springyness in the leg. Salto has demonstrated its ability to balance as well as hop [12] using the balance controller in [3], [4]. However, Salto's upper link is a reaction wheel, which means that the action of balancing has negligible effect on the spring, and vice versa.

The presence of a spring in the leg improves the machine's performance for walking or running activities by providing the ability to store and release elastic energy. This allows energy to be recycled efficiently from one step to the next, and also increases the instantaneous peak power available. The disadvantage is that it makes the control problem more complicated. In locomotion applications, the Spring-Loaded Inverted Pendulum (SLIP) model has proven itself to be very useful [13]. Many studies using the SLIP model have addressed the problem of controlling the transition from flight to stance and from stance back to flight for performing walking or running motions [14], [15], [16], [17]; but this is different from the task of stopping, where there is a need to remove kinetic energy quickly.

The problem of hopping with an asymmetric upper link was studied from a control perspective by Poulakakis and Grizzle [18]; while Batts *et al.* presented a 3D hopping machine with a parallel spring, which is more controllable than a series spring [19]. Their robot was able to perform continuous hops for about seven seconds before losing its balance. Xiong and Ames introduced a hopping and landing solution for a bipedal robot named Cassie [20]. They described the mechanical compliance of the robot leg by a virtual SLIP mechanism with nonlinear stiffness (the nonlinearity being a function of the kinematics of the leg). Then the prismatic motions of the virtual model are optimized to produce the desired hop.

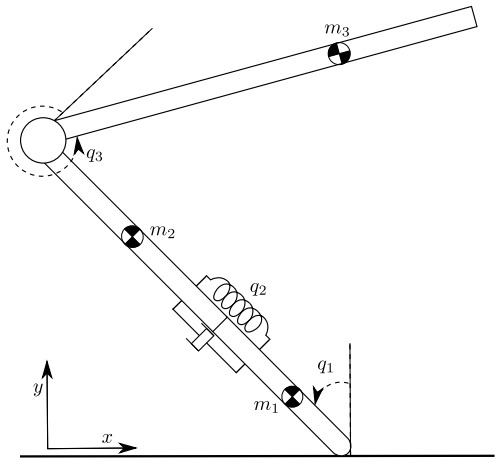

Fig. 1. Robot model. q_3 is negative in this configuration, and has been drawn as $q_3 + 2\pi$.

The study of combined balancing and hopping on a general planar double pendulum was pioneered by Berkemeier in [21], and this was the inspiration for Azad's later work in [22], [23]. These works assume a rigid leg. However, in an experiment that was never published, Azad repeated the work on single hops in [22], but with the rigid leg replaced by a springy one (i.e., the same arrangement as in Fig. 1, but with different parameter values). Although the work was never published, an animation of his result can be viewed at [1]. It can be seen that Azad's controller works very well with a springy leg. The study in this paper goes beyond Azad's work by using a faster, more responsive balance controller, and by investigating the spring's effect on the controller's ability to execute large, fast movements after the landing transient has died away.

Finally, there is an important distinction between the work presented here and that presented in [2], [3] on the topic of balancing in the presence of other motions. In the earlier work, the other motions were assumed to be known in advance, and executed accurately by a motion controller, so that the balance controller could be told in advance what the movements would be, and hence make the robot lean in anticipation of the balance disturbances that these motions were expected to cause. In contrast, we consider here a passive springy joint, which moves in response to the actions of the balance controller, and therefore causes unanticipated disturbances.

III. Experimet Setup

We consider two robots in this paper: one with a springy leg and one with a rigid leg. The springy-leg robot is shown in Fig. 1. The rigid-leg robot is obtained from the springy-leg robot by locking joint 2 (the prismatic joint) in the position it takes when the robot is balanced, the torso is at right angles to the leg ($q_3 = 0$), and the spring is holding the weight of the upper leg and torso against gravity (specifically, $q_2 = -0.01065\,\mathrm{m}$).

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 shown in Table I. These values were chosen to resemble the parameters of Skippy and Tippy. The symbols m_i, l_i and I_i appearing below denote the mass, length and rotational inertia about the centre of mass (CoM), respectively, of link i.

Link (i)	Mass (kg)	Length (m)	CoM (m)	Inertia at CoM (kgm^2)
1	0.2	0.2	0.1	0.0027
2	0.3	0.3	0.15	0.0150
3	2	0.5	0.33	0.0800

TABLE I

LENGTH AND INERTIA PARAMETERS OF THE ROBOT SHOWN IN FIG.1

The joint variables are q_1, q_2 and q_3. When all three are zero, the leg is vertical, the leg's length is l_1+l_2, and the torso is horizontal out to the right. 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 the actual length of the leg is $l_1 + l_2 + q_2$). In Fig. 1, q_1 is positive and q_3 is negative.

The stiffness of the spring is $2000\,\mathrm{N/m}$, and the damping coefficient is $15\,\mathrm{Ns/m}$, which results in an under-damped system. The stiffness is appropriate for a robot able to make small hops of around $1\,\mathrm{m}$ with a spring compression significantly less than l_1, yet 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.

Finally, 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 [23]. Contact forces acting on the foot in the normal and tangent directions are

$$\begin{aligned} F_y &= \max(0, K_n z^{3/2} + D_n z^{1/2}\dot{z}), \\ F_x &= \mathrm{clip}(K_t z^{1/2} u + D_t z^{1/2}\dot{u}, -\mu F_y, \mu F_y), \end{aligned} \tag{1}$$

where z and u are the ground compression and shear deformation, and μ is the coefficient of friction, K_n and D_n are the normal and K_t and D_t are the tangential stiffness and damping coefficients. The function $\mathrm{clip}(a, b, c)$ returns the value of a clipped to b and c. The parameter values used in the simulations are

$$\begin{aligned} K_t = 12.7 \times 10^6 \quad D_t = 3.1 \times 10^5 \quad \mu = 1 \\ K_n = 8.5 \times 10^6 \quad D_n = 3.1 \times 10^5 \end{aligned} \tag{2}$$

which are consistent with a hard floor and a high-friction hard rubber foot.

IV. Balance Theory

In this section, the analysis presented in [3] is modified for the planar robot mechanism shown in Fig. 1. The upper joint (joint 3) is actuated by the controller, the middle joint (joint 2) is actuated by a spring, and the lower joint (joint 1) represents the contact between the bottom of the lower link (the foot) and the ground and is consequently un-actuated. By representing the contact with a revolute joint, the controller

(but not the simulator) assumes that the foot neither slips nor loses contact with the ground and that the movement of the contact point as the foot rotates is negligible. The equation of motion of this robot is

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

where H_{ij} are 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 contain gravity and velocity terms, $\tau_s = -K_s q_2 - D_s \dot{q}_2$, K_s and D_s denote the stiffness and damping coefficients of the spring and τ_3 is the torque at the actuated joint.

As joint one is un-actuated, it follows that the force of gravity is the only force capable of exerting a moment about the support point and change the angular momentum of the robot about this point. If we define L to be the angular momentum of the whole robot about the support point then we have

$$\dot{L} = -m\, g\, c_x \tag{4}$$

where m is the total mass of the robot, g is the acceleration due to gravity (a positive number), and c_x is the x coordinate of the robot's center of mass (CoM). The expression $-m\, g\, c_x$ is the moment of gravity about the support. The equation that follows directly from (4) is

$$\ddot{L} = -m\, g\, \dot{c}_x \tag{5}$$

and the expression for L is

$$L = H_{11}\, \dot{q}_1 + H_{12}\, \dot{q}_2 + H_{13}\, \dot{q}_3 \tag{6}$$

which is proved in the appendix of [3].

The next step is to add a fictitious joint between joint one and the ground, which is a prismatic joint in the x-direction. We can call it joint zero so as not to disturb the numbering of the other joints. This joint never moves, so it does not affect the robot's dynamics. However, it does increase the size of the equation of motion, which now reads

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

The reason for this extra joint is that it provides us with two equations linking the joint-space dynamics with the motion of the CoM:

$$m\, \dot{c}_x = H_{01}\, \dot{q}_1 + H_{02}\, \dot{q}_2 + H_{03}\, \dot{q}_3 \tag{8}$$

and

$$\tau_0 = m\, \ddot{c}_x = -\dddot{L}/g \tag{9}$$

Combining (5), (6) and (8) gives

$$\begin{bmatrix} L \\ \ddot{L} \end{bmatrix} = \begin{bmatrix} H_{11} & H_{13} \\ -g\, H_{01} & -g\, H_{03} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} H_{12} \\ -g\, H_{02} \end{bmatrix} \dot{q}_2. \tag{10}$$

We omit the last term (the one involving $\dot{q}_2$) in order to make the point that the balance controller does not need to know about the dynamics introduced by the spring. The simplified equation can then be solved to give

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{g\, D} \begin{bmatrix} -g\, H_{03} & -H_{13} \\ g\, H_{01} & H_{11} \end{bmatrix} \begin{bmatrix} L \\ \ddot{L} \end{bmatrix} \tag{11}$$

where

$$D = H_{01}\, H_{13} - H_{11}\, H_{03} \tag{12}$$

assuming that the matrix is invertible (which it will be if the robot is physically capable of balancing [3]). Consequently $\dot{q}_3$ can be expressed as

$$\dot{q}_3 = Y_1\, L + Y_2\, \ddot{L} \tag{13}$$

where

$$Y_1 = \frac{H_{01}}{D}, \qquad Y_2 = \frac{H_{11}}{g\, D} \tag{14}$$

Y_1 and Y_2 vary with configuration, and can be expressed as simple functions of two physical properties of the mechanism: its time constant of toppling, T_c, which measures how quickly the robot falls if the controller does nothing, and its velocity gain [24], [25], which measures the effect on centre of mass (CoM) velocity of a unit change in the velocity of the actuated joint. The formulae are

$$Y_1 = \frac{1}{mg\, T_c^2\, G_v} \qquad Y_2 = -\frac{1}{mg\, G_v} \tag{15}$$

where G_v is the linear velocity gain as defined in [25]. T_c appears again in the acausal filter mentioned in the next section.

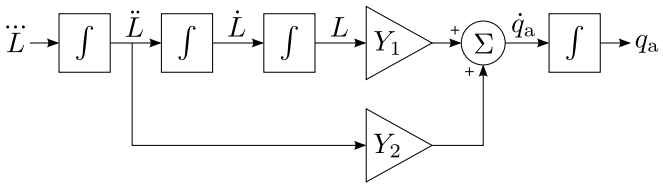

Fig. 2. Plant describing the dynamics of balancing. q_a is the actuated joint variable, which is q_3 in Fig. 1.

V. Balance Controller

The balance controller works by controlling the plant shown in Fig. 2, as explained in [2], [3]. The job of the balance controller is to calculate a value for $\dddot{L}$ to make q_a follow a given command signal, q_c, without losing balance. Denoting $q_\mathrm{a} := q_3$, a suitable control law to accomplish this is

$$\dddot{L} := k_{dd}\ddot{L} + k_d\dot{L} + k_L L + k_q(q_\mathrm{a} - u)\,, \tag{16}$$

where u is the input to the control law (see below). The feedback gains are obtained via pole placement as

$$\begin{aligned} k_{dd} &= -a_3 & k_d &= -a_2 + a_0 Y_2/Y_1 \\ k_L &= -a_1 & k_q &= -a_0/Y_1\,, \end{aligned} \tag{17}$$

where

$$\begin{aligned} a_0 &= \lambda_1\lambda_2\lambda_3\lambda_4 \\ a_1 &= -\lambda_1\lambda_2\lambda_3 - \lambda_1\lambda_2\lambda_4 - \lambda_1\lambda_3\lambda_4 - \lambda_2\lambda_3\lambda_4 \\ a_2 &= \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_1\lambda_4 + \lambda_2\lambda_3 + \lambda_2\lambda_4 + \lambda_3\lambda_4 \\ a_3 &= -\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4 \end{aligned} \tag{18}$$

and $\lambda_1, \ldots, \lambda_4$ are the chosen values of the poles. We set λ_1 to the closed-loop bandwidth that we want the controller to achieve, while λ_2 and λ_3 are destined to be cancelled by two introduced zeros, mentioned below, and λ_4 is set to $-1/T_c$ in order to cancel a natural zero in the transfer function. The use of pole placement involves an assumption that Y_1 and Y_2 are constants, implying that the plant is linear. This assumption is justified by the observation that in practice Y_1 and Y_2 vary slowly with configuration.

The input to the control law, u, is computed from the command signal and its derivatives, q_c, $\dot{q}_c$ and $\ddot{q}_c$, according to

$$u = \mathrm{AF}(q_c - (\lambda_2 + \lambda_3)\dot{q}_c + \lambda_2\lambda_3\ddot{q}_c)\,. \tag{19}$$

This formula has two effects. First, it introduces two zeros into the transfer function, at λ_2 and λ_3, which cancel the corresponding poles. Second, it applies an acausal filter, AF, which makes the robot lean in anticipation of the balance disturbances that will be caused by future commanded motions [3]. Specifically, AF is a first-order low-pass filter with time constant T_c, which runs backwards in time from a point sufficiently far in the future back to the present. To implement this filter, the controller needs to know the expected short-term future value of q_c. Information of this kind can be found in the robot's high-level controller, which usually knows what movement it intends to make next.

Given the control law in (16), with gains as in (17) and (18), and the input signal as in (19), it can be shown that the complete transfer function from q_c to q_a would be

$$q_a(s) = \frac{1}{1 + s/(-\lambda_4)}\, q_c(s) \tag{20}$$

if it were really true that Y_1 and Y_2 were constants [3]. We take this expression as the theoretical transfer function of the balance controller, and compare the actual response with the theoretical one in the experimental results reported below.

Finally, the output of the control law ($\dddot{L}$) must be converted to a torque or an acceleration at the actuated joint, which can be done by solving the equations (7) and (9),

$$\begin{bmatrix} 0 & H_{01} & H_{02} & H_{0a} \\ 0 & H_{11} & H_{12} & H_{1a} \\ 0 & H_{21} & H_{22} & H_{2a} \\ -1 & H_{a1} & H_{a2} & H_{aa} \end{bmatrix} \begin{bmatrix} \tau_a \\ \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_a \end{bmatrix} = \begin{bmatrix} -\dddot{L}/g - C_0 \\ -C_1 \\ \tau_s - C_2 \\ -C_a \end{bmatrix}. \tag{21}$$

VI. Simulation Experiment

This section presents the results of a simulation experiment in which the springy-leg robot starts in an upright position, tips itself forward (positive x direction), crouches, launches into a hop of 0.5 m length and 0.62 m height (measured as the rise of the CoM), lands with a single bounce, stabilizes itself in a balanced configuration with $q_a = 0$, and then executes a motion command consisting of a sequence of ramps and sinusoids.

The complete motion can be seen in the accompanying video. However, the initial part of this action sequence, from the beginning up to the moment of first landing, are outside the scope of this paper, and will not be discussed. They employ control systems different from the one presented in this paper.

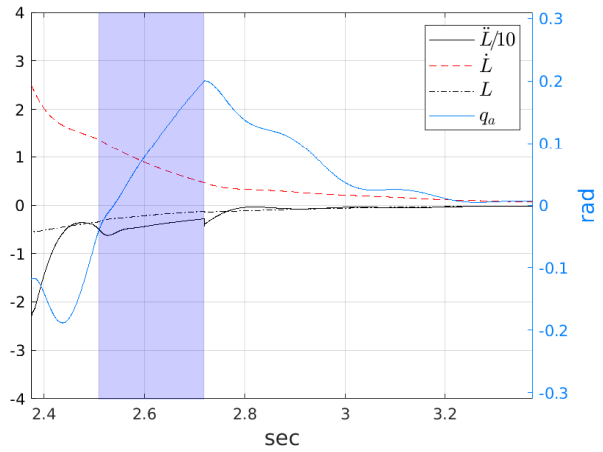

Fig. 3. Evolution of the controller's state variables from the moment of landing until the robot has settled. The shaded zone shows the time interval during which the foot has lost contact with the ground because of the bounce.

In this experiment we use the ode23t integrator from MATLAB with a relative tolerance set to 10^{-6} and other parameters at their default values. The controller is implemented as a continuous-dynamics subsystem, and the sensors are assumed to be perfect. Although the balance controller assumes that the foot never loses contact nor slips with the ground, in the simulation we incorporate the contact model described at the end of section III.

The balance controller is switched on at the moment of the first landing, which occurs at $t = 2.374\,\mathrm{s}$; and the subsequent evolution of the controller's state variables during the settling period is plotted in Fig. 3. At this phase, the poles of the controller are set to -20,-7,-7,$-1/T_c$, all in units of $\mathrm{rad/s}$, to produce a bounce after the first touchdown in order to demonstrate the controller's response to situations when it's assumption about the ground foot contact is not satisfied for short periods (bounces). In the graph it can be seen that q_3 is initially pushed to about $-0.18\,\mathrm{rad}$ by the momentum of the landing, then rises to nearly $0.19\,\mathrm{rad}$ during the bounce (shaded area). After the second touchdown, the joint is driven close to zero.

The robot's foot loses contact with the ground briefly between $t = 2.5091\,\mathrm{s}$ and $t = 2.7182\,\mathrm{s}$ and reaches a maximum height of $5.0973\,\mathrm{cm}$. This period is shown shaded in the graph. During this period, the balance controller does not know that the foot has left the ground, and continues to use the model described in Section IV, which assumes that the robot's foot is on the ground. It can be seen that this short period of flight does not significantly affect the controller. It can also be seen that the robot has come to rest within about $0.48\,\mathrm{s}$ of the second landing. Note that the controller is assumed to know the orientation of the robot at all times, e.g. from an onboard inertial measurement unit, so it always knows the correct values of q_1 and $\dot{q}_1$.

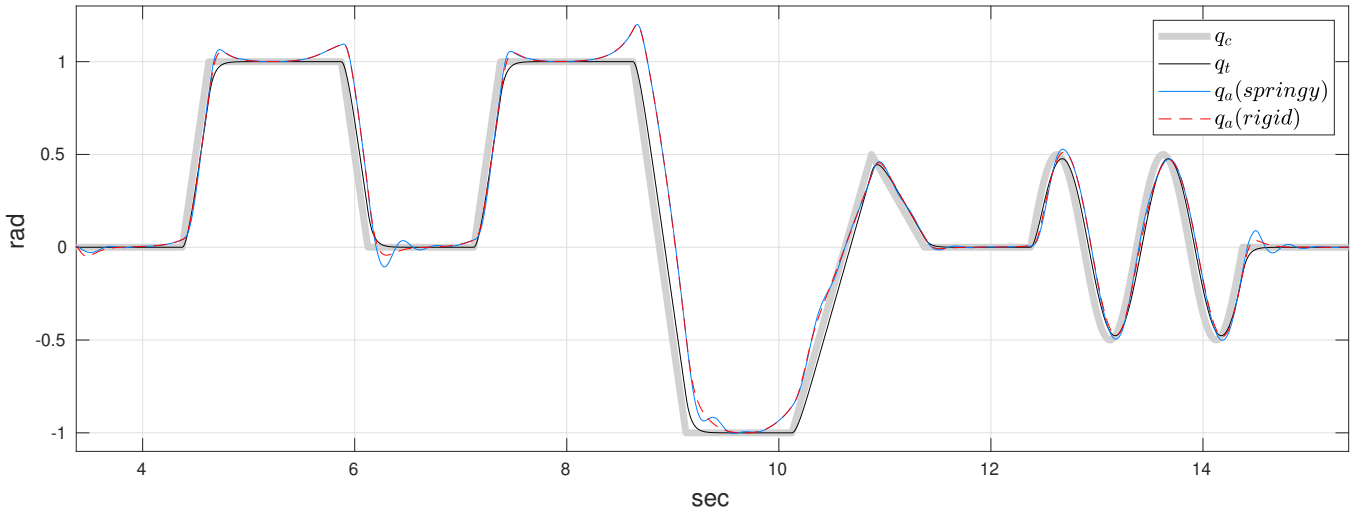

Fig. 4. Tracking performance of the balance controller

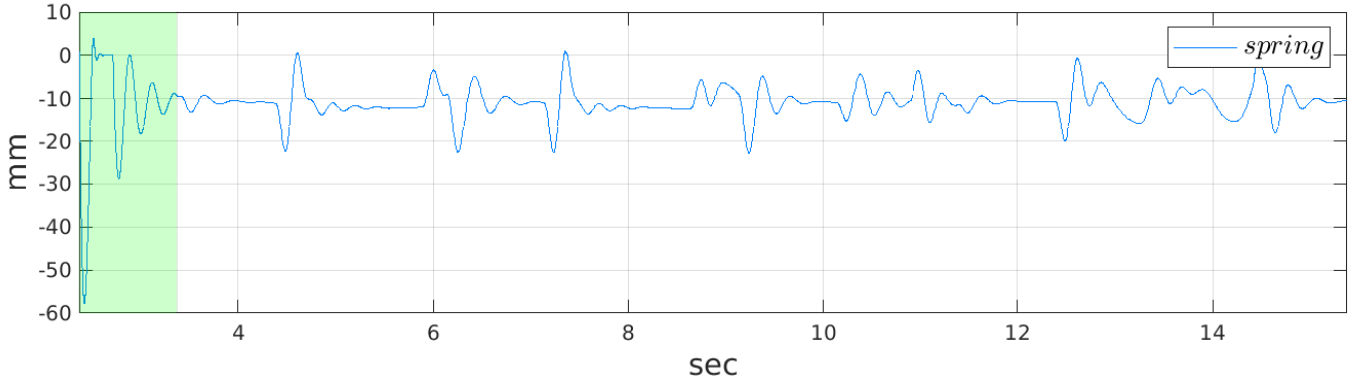

Fig. 5. Motion of the spring from the moment of landing until the end of the motion tracking shown in Fig. 4

At time 3.374 s the command signal (q_c) switches from zero to the sequence of ramps and sine waves mentioned above. This portion of the action sequence is plotted in Fig. 4. The command sequence consists of three ramps between 0 and 1 rad, followed by a long ramp from 1 rad to -1 rad, all at a speed of 4 rad/s, followed by two more ramps at speeds of 2 rad/s and 1 rad/s, followed by two cycles of a sine wave at 1 Hz. Observe that this command sequence asks the robot to make large, fast movements.

During this tracking phase, the poles are set at -20, -20, -20 and $-1/T_c$, all in units of rad/s, and the two introduced zeros are at -20 rad/s. For this robot, T_c varies between 0.2112 s at $q_a = -1.0087$ rad and 0.287 s at $q_a = 1.1343$ rad in this action sequence.

The signals appearing on Fig. 4 are: the original command signal, q_c (i.e., not u as in (19)); the theoretical response of the balance controller if the plant really were linear (labelled q_t); the actual response of the actuated joint on the springy leg robot (labelled $q_a(springy)$); and the actual response on the rigid-leg robot (labelled $q_a(rigid)$). 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 3.374 s. Fig.5 shows the motion of the spring during landing (shaded area) and tracking of the command sequence (after the shaded area). At the initial touch-down, the spring reaches a compression peak of about 6 cm which is 12% of the length of the leg. During the tracking, the spring compression varies in a 2 cm range, which is 4% of the leg length.

It can be seen that there are some significant tracking errors, particularly at the beginnings and ends of the ramps. There are two main contributors to these errors. The greater one is that the balance controller assumes that the plant in Fig. 2 is linear, when in reality it is not. The lesser one is an approximation in the way that the acausal filter works: for simplicity, it uses values of T_c calculated for a balanced configuration at each instant instead of a leaning configuration.

It can also be seen that the springy-leg and rigid-leg responses are very nearly the same. So, the presence of the spring-loaded joint in the leg has had almost no effect on the closed-loop behavior of the robot while following large, fast motion commands; and this has happened even though the the spring is relatively soft and the spring-loaded joint

makes significant movements. This is the main result of this paper. One place where one can see a difference occurs at the end of the second ramp, where the springy-leg response shows a small amount of ringing that dies away in about 0.5 s. There is also a little bit of ringing at the end of the long ramp, and at the end of the sine wave. However, considering the under-damped nature of the spring-damper pair in the leg, there is remarkably little ringing overall, and what little there is dies away quickly. So we can conclude that in this experiment the balance controller has accomplished three tasks simultaneously using only a single actuator: balance the robot, follow the command signal, and suppress vibrations.

VII. Conclusion

This paper investigated the effect of a springy leg on the balance controller described in [2], [3], which aims to achieve and maintain balance in the presence of large, fast motions. The purpose of this study was to investigate the feasibility of a robot that uses a spring in its leg to help it hop efficiently, but which also needs to balance on that leg. It was shown that the balance controller can cope with the large, fast motions that occur during landing, including small bounces, and that it can track large, fast motion commands after landing with almost the same accuracy as could be achieved if the leg were rigid. And all that is necessary in order to achieve this performance is that the spring-loaded joint is instrumented, and that the measured motion of the spring is taken into account when mapping from the robot's state variables to the balance controller's state variables. The controller itself does not need to know about the dynamics introduced by the spring, although these dynamics are taken into account when mapping the output of the controller to a joint torque or acceleration command. It was also shown that the balance controller can suppress vibrations caused by the spring. Future work will investigate the effect of a series elastic element in the actuated joint.

Acknowledgement

The authors wish to thank Dr. Azad for providing his original software and other assistance.

References

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

[2] R. Featherstone, "A New Simple Model of Balancing in the Plane," in *Bicchi A. and Burgard W. (eds) Robotics Research (vol. 2), pp. 167–183, Springer Proceedings in Advanced Robotics, vol. 3, Springer, Cham*, 2018.

[3] 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.

[4] 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), pp. 9459–9465, May 20-24 2019.

[5] 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.

[6] 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.

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

[8] 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), pp. 3722–3727, Oct. 7–12 2012.

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

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

[11] 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.

[12] 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.

[13] M. M. Ankaral, U. Saranl, and A. Saranl, "Control of Underactuated Planar Hexapedal Pronking through a Dynamically Embedded SLIP Monopod," in *2010 IEEE International Conference on Robotics and Automation*, (Anchorage AK, USA), pp. 4721–4727, May 3–7 2010.

[14] A. Hereid, S. Kolathaya, M. S. Jones, J. Van Why, J. W. Hurst, and A. D. Ames, "Dynamic Multi-Domain Bipedal Walking with Atrias through SLIP Based Human-Inspired Control," in *17th International Conference on Hybrid Systems: Computation and Control*, (Berlin, Germany), pp. 263–272, Apr. 15–17 2014.

[15] G. Piovan and K. Byl, "Enforced Symmetry of the Stance Phase for the Spring-Loaded Inverted Pendulum," in *2012 IEEE International Conference on Robotics and Automation*, (St. Paul, MA, USA), pp. 1908–1914, May 14–18 2012.

[16] S. Riese and A. Seyfarth, "Robustness and Efficiency of a Variable-leg-spring Hopper," in *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, (Rome, Italy), pp. 1347–1352, Jun. 24–27 2012.

[17] P. M. Wensing and D. E. Orin, "High-speed Humanoid Running through Control with a 3D-SLIP Model," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Tokyo, Japan), pp. 5134–5140, Nov. 3–7 2013.

[18] I. Poulakakis and J. W. Grizzle, "Formal embedding of the Spring Loaded Inverted Pendulum in an Asymmetric Hopper," in *Proceedings of the 2007 European Control Conference*, (Kos, Greece), pp. 3159–3166, Jul. 2–5 2007.

[19] Z. Batts, J. Kim, and K. Yamane, "Untethered One-Legged Hopping in 3D Using Linear Elastic Actuator in Parallel (LEAP)," in *2016 International Symposium on Experimental Robotics*, (Tokyo, Japan), pp. 103–112, Oct. 3–6 2017.

[20] X. Xiong and A. Ames, "Bipedal Hopping: Reduced-order Model Embedding via Optimization-based Control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Madrid, Spain), pp. 3821–3828, Oct. 1–5 2018.

[21] M. D. Berkemeier and R. S. Fearing, "Sliding and Hopping Gaits for the Underactuated Acrobot," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 629–634, 1998.

[22] 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), pp. 2027–2032, May 6–10 2013.

[23] M. Azad, *Balancing and hopping motion control algorithms for an under-actuated robot*. PhD thesis, Australian National University. Research School of Engineering, 2014.

[24] R. Featherstone, "Quantitative Measures of a Robot's Ability to Balance," in *Proceedings of Robotics: Science and Systems*, (Rome, Italy), Jul. 13–17 2015.

[25] 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.