Balancing and Hopping Motion of a Planar Hopper with One Actuator

Morteza Azad[1] and Roy Featherstone[2]

***Abstract*— In this paper, a new control algorithm is presented for implementing hopping and balancing motions on a planar hopping machine with a single actuated revolute joint. Starting with a simple control algorithm for balancing, it is extended to perform trajectory-tracking maneuvers, which enables it to perform the crouching, lift-off and flight phases of a single hop, as well as re-balancing after landing. Simulation results are presented showing that the control system works well, and that it is not significantly affected by small amounts of slipping between the foot and the ground.**

I. Introduction

More than twenty years ago, Raibert introduced a control algorithm for the hopping motion of a one-legged robot [13]. His planar hopper consisted of a torso and a springy telescopic leg, connected to each other by a revolute joint (hip). Raibert's hopper had two actuators: one in the hip joint and the other in the leg. His robot showed good performance at hopping in place, hopping at various speeds, and leaping over small obstacles. After that, many studies and experiments have been done on Raibert-style (telescopic leg) hoppers, which are reviewed in [14].

According to [14], another type of leg that has been used in hopping robots is the *knee leg*, in which the prismatic joint of a Raibert-style hopper is replaced with a revolute joint. Hoppers of this kind can be divided into two groups: those that have two joints in total, and those that have only one. In the former, the two joints represent a hip and a knee. In the latter, the one joint can represent either a hip joining a rigid leg to a torso, or a knee joining the lower and upper parts of a leg.

Berkemeier and Fearing were the first researchers to study hopping in a one-joint knee-leg hopper [3]. Their hopper was modelled on the Acrobot (for acrobatic-robot which was studied by Hauser and Murray [7]). They divided the hopping control problem into two parts: stance phase control and flight phase control. In the stance phase, if the robot is not sliding on the ground, the dynamics of the hopper are identical to those of a two degrees of freedom (DoF) balancer robot with one degree of underactuation. To control the robot in the stance phase, they designed a controller to balance the robot [4] which was able to tolerate a little sliding (it presents small disturbances to the system) and maintain its balance while it accelerates its center of mass (CoM) upward. They used this controller to oscillate the robot to get enough energy for taking off the ground by tracking a prescribed trajectory during stance phase. In the flight phase, they found a desired trajectory for the hop and derived a controller to track that trajectory during the flight. Their control strategy in the flight phase was to rotate the lower leg, an integral number of times, to enable the robot to land in the same configuration as it took off the ground. They were able to show a good performance of sliding and hopping motion of a 2D hopper in simulation.

After Berkemeier and Fearing's studies, there have been a few studies [9], [8], [6] and [15] on knee-leg hoppers with only one actuator. There are also a number of studies that have been done on knee-leg hoppers with two actuators. Ohashi and Ohnishi [10] built a knee-leg hopping robot and demonstrated hopping motion in a plane with desired hopping height. Poulakakis and Grizzle [11] and [12] designed and built a planar knee-leg hopper named Thumper. They experimentally verified their hybrid zero dynamic control algorithm for its hopping motion. Also Grizzle et al. [5] introduced nonlinear control for a general case of underactuated planar mechanical systems during stance and flight phases. They showed ballistic motion of a knee-leg hopper in simulation.

This paper does not consider continuous hopping motion. Instead, it considers a single-hop motion performed by a knee-leg hopper mechanism with only one joint. Starting from a balanced configuration, the hop consists of a crouch-and-launch phase, a flight phase, a landing, and a balance recovery phase that ends with the robot once again in a balanced configuration.

A new simple balancing controller, which was first introduced in [2], is used to balance the robot during the balance recovery phase. An extended version of this controller, which can follow trajectories specifying a deliberate deviation from a balanced configuration, is used to implement the crouch-and-launch phase; and this same controller is used to control the horizontal component of foot motion during the flight phase. The performance of these controllers is demonstrated by simulation, and the video accompanying this paper shows examples of both balancing and hopping motions.

Unlike Berkemeier and Fearing's work, our hopper does not rotate its leg during the flight phase, and it makes no attempt to land in the same configuration as it takes off. Instead, it seeks only to control the hop length during flight, and to land in any configuration from which it can recover its balance.

The proposed controller in this paper is able to show a wider variety of motion behaviours compared with previous similar controllers. It is able to stabilize the robot in a range of unstable balanced configurations, to follow trajectories

[1]M. Azad is with School of Engineering, Australian National University, Canberra, Australia morteza.azad at anu.edu.au

[2]R. Featherstone is a consultant with the Italian Institute of Technology, Genoa, Italy roy.featherstone at ieee.org

specifying deviations from a balanced state, to hop and land on a desired location, and to cope with slipping between the foot and the ground, which can happen during crouching, at lift-off, and after landing.

II. Robot model and motion equations

A planar one-joint knee-leg hopper mechanism consists of two links connected to each other by an actuated revolute joint. Fig. 1 shows a schematic diagram of this hopper. It

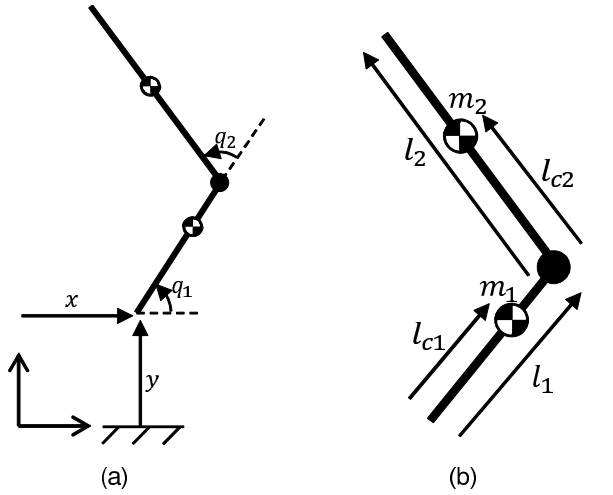

Fig. 1. Planar hopper model, (a) generelized coordinates, (b) parameters

is assumed that the tip of the lower leg (which is called the foot) is the only point that can make contact with the ground. The motion equations for this robot are:

$$\begin{aligned} F_x &= c\ddot{x} - (c_4\sin(q_1) + c_5\sin(q_1+q_2))\ddot{q}_1 \\ &\quad -c_5\sin(q_1+q_2)\ddot{q}_2 - c_4\cos(q_1)\dot{q}_1^2 \\ &\quad -c_5\cos(q_1+q_2)(\dot{q}_1+\dot{q}_2)^2 \end{aligned} \tag{1}$$

$$\begin{aligned} F_y &= c\ddot{y} + (c_4\cos(q_1) + c_5\cos(q_1+q_2))\ddot{q}_1 \\ &\quad +c_5\cos(q_1+q_2)\ddot{q}_2 - c_4\sin(q_1)\dot{q}_1^2 \\ &\quad -c_5\sin(q_1+q_2)(\dot{q}_1+\dot{q}_2)^2 + cg \end{aligned} \tag{2}$$

$$\begin{aligned} 0 &= -(c_4\sin(q_1) + c_5\sin(q_1+q_2))\ddot{x} + (c_4\cos(q_1) + \\ &\quad c_5\cos(q_1+q_2))\ddot{y} + (c_1+c_2+2c_3\cos(q_2))\ddot{q}_1 \\ &\quad +(c_2+c_3\cos(q_2))\ddot{q}_2 - 2c_3\sin(q_2)\dot{q}_1\dot{q}_2 \\ &\quad -c_3\sin(q_2)\dot{q}_2^2 + c_4 g\cos(q_1) + c_5 g\cos(q_1+q_2) \end{aligned} \tag{3}$$

$$\begin{aligned} \tau &= -c_5\sin(q_1+q_2)\ddot{x} + c_5\cos(q_1+q_2)\ddot{y} + \\ &\quad (c_2+c_3\cos(q_2))\ddot{q}_1 + c_2\ddot{q}_2 + c_3\sin(q_2)\dot{q}_1^2 \\ &\quad +c_5 g\cos(q_1+q_2) \end{aligned} \tag{4}$$

where I_1 and I_2 are the moments of inertia of the links about their CoM, F_x and F_y are the contact forces exerted from the ground to the foot, g is the acceleration of gravity, and

$$\begin{aligned} c &= m_1+m_2\,, & c_1 &= m_1 l_{c1}^2 + m_2 l_1^2 + I_1\,, \\ c_2 &= m_2 l_{c2}^2 + I_2\,, & c_3 &= m_2 l_1 l_{c2}\,, \\ c_4 &= m_1 l_{c1} + m_2 l_1\,, & c_5 &= m_2 l_{c2}\,. \end{aligned}$$

When the foot is in contact with the ground and it is not sliding, it is possible to consider the robot as a two degrees of freedom balancer robot. So by neglecting the values of x and y in (1)-(4), the motion equations for this balancer (or hopper during the stance phase) would be:

$$\begin{aligned} 0 &= (c_1+c_2+2c_3\cos(q_2))\ddot{q}_1 + (c_2+c_3\cos(q_2))\ddot{q}_2 \\ &\quad -(2c_3\sin(q_2)\dot{q}_1\dot{q}_2 + c_3\sin(q_2)\dot{q}_2^2) \\ &\quad +c_4 g\cos(q_1) + c_5 g\cos(q_1+q_2) \end{aligned} \tag{5}$$

$$\begin{aligned} \tau &= (c_2+c_3\cos(q_2))\ddot{q}_1 + c_2\ddot{q}_2 \\ &\quad +c_3\sin(q_2)\dot{q}_1^2 + c_5 g\cos(q_1+q_2) \end{aligned} \tag{6}$$

The parameters for the hopper that have been used in the simulations are (Acrobot's parameters in [3]):

$$m_1 = 2\text{kg}\,, \quad m_2 = 14\text{kg}\,, \quad l_1 = 0.5\text{m}\,, \quad l_2 = 0.75\text{m}$$
$$I_1 = l_{c1} = 0\,, \quad l_{c2} = 0.375\,, \quad I_2 = m_2 l_{c2}^2\,.$$

Contact between the robot's foot and the ground is modelled as a nonlinear compliant contact with Coulomb friction, using a planar version of the 3D contact model described in [1]. The forces acting on the foot in the normal and tangent directions are

$$\begin{aligned} F_y &= \max(0, K_n z^{\frac{3}{2}} + D_n z^{\frac{1}{2}}\dot{z})\,, \\ F_x &= \text{clip}(K_t z^{\frac{1}{2}} u + D_t z^{\frac{1}{2}}\dot{u}, -\mu F_y, \mu F_y)\,, \end{aligned} \tag{7}$$

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

$$K_n = 8.5\times10^6\,, \quad K_t = 12.75\times10^6\,,$$
$$D_n = D_t = 3.1\times10^5\,, \quad \mu = 0.4\,.$$

Further details, including how to calculate z and u, are explained in [1].

III. Control strategies

As already mentioned, Berkemeier and Fearing's hopping controller was divided into two parts: stance and flight phases. The objective of the stance-phase controller was to oscillate the robot, by tracking special trajectories, to get enough energy to take off from the ground. Their flight-phase controller aimed to land the robot on the ground in the same configuration as when it took off by tracking a special trajectory. So at the end of its flight phase, the robot is ready to hop again.

This paper studies a hopping gait in which the robot starts from a balanced (upright) configuration and ends in the same balanced configuration. The motion phases of such a hopping gait, after starting from a balanced configuration, are: launching (from the beginning to the instant of take-off), flight, and landing and balancing again at the end. The landing phase starts from the moment the foot touches the

ground, and includes a short period when foot is sliding, which will physically happen in a real robot. Since the proposed controller for balancing the robot in this paper is able to balance the robot from the beginning of the landing phase, without any extra setup, so the hopping motion is divided into three phases: balancing, launching and flight. We now consider each phase in turn.

A. Balancing Controller

We start by assuming that the foot does not slip, which implies that the motion of the robot is given by (5) and (6). Let X denote the horizontal displacement of the CoM relative to the robot's foot (the contact point during the stance phase):

$$X = \frac{1}{c}(c_4 \cos(q_1) + c_5 \cos(q_1 + q_2)) \,, \tag{8}$$

and let L denote the angular momentum of the robot about the contact point:

$$L = (c_1 + c_2 + 2c_3 \cos(q_2))\dot{q}_1 + (c_2 + c_3 \cos(q_2))\dot{q}_2 \,. \tag{9}$$

The conditions for balance are: $X = 0$ and $\dot{q}_1 = \dot{q}_2 = 0$. However, as $\dot{X}$ and L are both linear functions of $\dot{q}_1$ and $\dot{q}_2$, the two velocity constraints can be replaced with $\dot{X} = 0$ and $L = 0$. Now, the rate of change of angular momentum of a multibody system about any fixed point equals the moment about that point of the external forces acting on the system. As L is chosen to be expressed at the contact point, which is the one point about which the moment of the ground reaction force is always zero, it follows that $\dot{L}$ must equal the moment of the gravitational force about the contact point. So

$$\dot{L} = -cgX \,, \tag{10}$$

and therefore also

$$\ddot{L} = -cg\dot{X} \,. \tag{11}$$

Equation (10) could also be derived by differentiating (9) and plugging it into (5). The conditions for balance can now be written $L = \dot{L} = \ddot{L} = 0$; so it is possible to consider the angular momentum as an output function and define a linear feedback controller that drives the output function to zero exponentially. So the control law for the balancing phase would be

$$\tau = k_{dd}\ddot{L} + k_d\dot{L} + k_pL + \tau^d \,, \tag{12}$$

where k_{dd}, k_d and k_p are controller gains, and τ^d is the correct holding torque at the actuated joint for the desired balanced configuration. The effect of τ^d is to make this configuration an equilibrium point of the closed-loop system. If q_1^d and q_2^d are the joint angles in the desired configuration, then (6) gives

$$\tau^d = c_5 g \cos(q_1^d + q_2^d) \,.$$

This controller is equivalent to

$$\tau = -k_v\dot{X} - k_xX + k_pL + \tau^d \tag{13}$$

where $k_v = cgk_{dd}$ and $k_x = cgk_d$. This alternative control law will be used in the launching and flight phases.

Stability Analysis and Gain Calculation

Considering the nonlinear state-space equations for the hopper during the stance phase as

$$\dot{\eta} = f(\eta) + g(\eta)\ \tau$$

where $\eta = (q_1 - q_1^d, q_2 - q_2^d, \dot{q}_1, \dot{q}_2)$, and knowing that τ is a function of η, it follows that $\dot{\eta}$ is a function of η, so the system as a whole can be described by a nonlinear equation of the form $\dot{\eta} = h(\eta)$. Linearizing about $\eta = 0$ gives

$$\dot{\eta} = A\,\eta$$

where $A = \frac{\partial h}{\partial \eta}|_{\eta=0}$ is a 4×4 matrix. To check the asymptotic stability of the system and calculate the controller gains, the eigenvalues of matrix A, which are the roots of its characteristic equation, need to be calculated. The characteristic equation for matrix A has the general form:

$$a\lambda^4 + (gKk_{dd})\lambda^3 + (gKk_d + \alpha)\lambda^2 + (gKk_p)\lambda + b = 0 \tag{14}$$

where

$$\begin{aligned}
a &= c_1c_2 - c_3^2\cos(q_2^d)^2 \,, \\
b &= g^2c_4c_5\sin(q_1^d)\sin(q_1^d + q_2^d) \,, \\
\alpha &= -g(c_1c_5\sin(q_1^d + q_2^d) + c_2c_4\sin(q_1^d)) \,, \\
K &= c_4\sin(q_1^d)(c_2 + c_3\cos(q_2^d)) \\
&\quad -c_5\sin(q_1^d + q_2^d)(c_1 + c_3\cos(q_2^d)) \,.
\end{aligned}$$

This system will be stable if all four eigenvalues have negative real parts. This, in turn, implies that the product of the four eigenvalues must be strictly positive. However, (14) implies that

$$\prod_{i=1}^{4} \lambda_i = \frac{b}{a} \,. \tag{15}$$

So an essential prerequisite for the controller in (12) or (13) to be stable is $\frac{b}{a} > 0$. Observing that a is always positive, we adopt the following restrictions on q_1^d and q_2^d, which ensure that b is always positive:

$$\sin(q_1^d) > 0 \,, \qquad \sin(q_1^d + q_2^d) > 0 \,. \tag{16}$$

Physically, this says that both links must be sloping or pointing upwards.

Given (15), if we define an optimal controller to be one that maximizes the speed of the slowest pole, then the optimal choice of eigenvalues is

$$\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = -p \,,$$

where $p = \sqrt[4]{\frac{b}{a}}$. The gains must then be chosen so that (14) matches the polynomial

$$(\lambda + p)^4 = \lambda^4 + 4p\lambda^3 + 6p^2\lambda^2 + 4p^3\lambda + p^4 = 0 \,.$$

With this choice of gains, all poles of the closed loop system are negative, so the system is asymptotically stable and the controller gains are

$$k_p = \frac{4\,p^3\,a}{gK} \,, \quad k_d = \frac{6\,p^2 a - \alpha}{gK} \,, \quad k_{dd} = \frac{4\,p\,a}{gK} \,. \tag{17}$$

Clearly, this requires $K \neq 0$. Fortunately, it can be proved that for the Acrobot $K \neq 0$ in every configuration that satisfies $c_4 \cos(q_1^d) + c_5 \cos(q_1^d + q_2^d) = 0$ (the condition for balance—see (8)) and also (16). Note, however, that $K = 0$ may be possible for other robots.

Fig. 2 shows how the state variables approach to zero during a balancing motion of the robot when it is in contact with the ground and is not slipping. In this example, the robot starts from an initial stationary crouched position $(\pi - \tan^{-1}(c_4/c_5), -\frac{\pi}{2})$, and moves to the balanced upright position $(\frac{\pi}{2}, 0)$. This motion appears in the accompanying video, followed by a motion in the opposite direction (i.e., from upright to crouched).

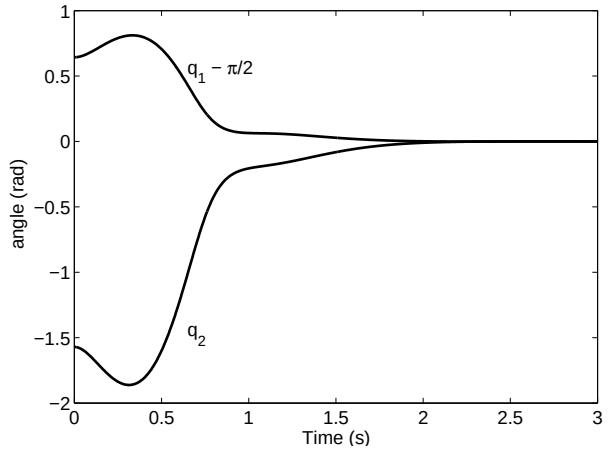

Fig. 2. Robot's balancing motion from crouched to upright position

B. Launching Phase

In the launching phase, the objective is to move the robot from its initial balanced position to a desired dynamic state which is called the launching state in this paper. The launching state is an unstable configuration at the end of the launching phase in which the robot has positive velocity in both the horizontal and vertical directions and is ready to hop. To implement the launching phase, the robot must be able to deliberately lose its balance and pass through a predetermined unstable configuration for an instant which is the take-off instant.

In order to move the robot from its initial state at the balanced position to the desired state at the launching configuration, a trajectory-tracking controller will be used. This controller is a modified version of the controller in (13), and will be described later in the next subsection.

To define a trajectory between the initial and final states of the launching phase, since the controller is based on the angular momentum of the robot, it is necessary to map the state variables to angular momentum (and its derivatives) and then find a suitable trajectory between the initial and final angular momentum. This trajectory must also satisfy the boundary conditions for the derivatives of angular momentum. It is already known from (10) and (11) that there is a linear relation between these derivatives and the horizontal displacement and velocity of the CoM of the robot with respect to the contact point during the stance phase. Therefore, a set of desirable trajectories of L, X and $\dot{X}$ can be used as reference trajectories.

To find a suitable reference trajectory for the launching phase, we employ a time-reversal technique. Starting at the desired lift-off state, but with the launch velocities reversed, we use the balancing controller to bring the robot to its initial balanced configuration. The resulting trajectory, when time-reversed, serves as the reference trajectory for the launching phase. To ensure that the trajectory-tracking controller reaches the lift-off state with a high degree of accuracy, we modify the balancing controller used to compute the reference trajectory so that it begins its balancing maneuver gently (i.e., with a small torque at the actuator). This gives the trajectory-tracking controller a chance to minimize tracking errors at the end of the trajectory.

C. Trajectory Tracking Controller

As already mentioned, during the launching and flight phases, the alternative control law in (13) is used. For the purpose of trajectory tracking, it is modified as

$$\tau = -k_v(\dot{X} - \dot{X}^d) - k_x(X - X^d) + k_p(L - L^d)\,, \quad (18)$$

where L^d is the desired value of angular momentum and X^d is the desired value of X within the desired trajectory. When the robot is in contact with the ground (without slipping), it is possible to calculate L^d from X^d as follows:

$$\dot{L}^d = -cgX^d \implies L^d = -cg \int X^d \mathrm{d}t\,.$$

Although we use this control law to deliberately tip the robot forward, it is also capable of following some trajectories without losing balance. For example, if the desired trajectory for X is a sine-wave function of time, such as $X^d = \rho \sin(\omega t)$, where ρ and ω are constants (with some limits on their values), then the controller can track this trajectory without losing balance, as shown in Fig. 3. The relatively large tracking errors are due to the fact that it is physically impossible for the robot (which is underactuated) to follow this trajectory exactly, so the actual motion follows a physically possible trajectory instead.

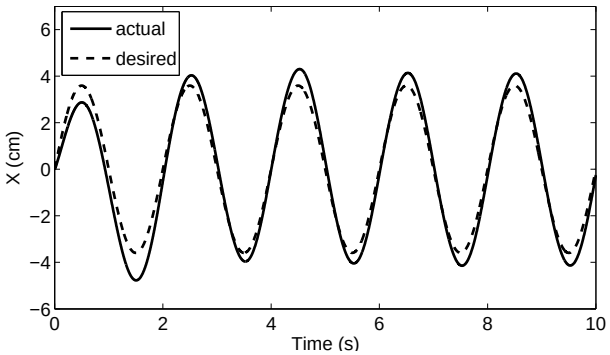

Fig. 3. Trajectory tracking of a sine wave

Fig. 4 shows an example of the trajectory-tracking performance of the robot during the launching phase. In this

example, the robot starts from a stationary balanced upright position and aims to reach the launching state $\eta^d = (2, -1.5, -4.4, 7.5)$. The solid and dashed curves in Fig. 4 show X and X^d as a function of time. As can be seen, the tracking accuracy is high at the beginning and end of the trajectory, although there is significant tracking error beginning at about $t = 1.2$s. The reason for this error can be seen in Fig. 5, which shows the normal component of the ground-reaction force acting on the foot, and also the ratio of the tangent and normal components. The latter is constrained to ± 0.4 by the friction cone. It can be seen from these graphs that the normal force has dropped close to zero, and the foot has started slipping forward as a consequence. The reason for the drop in normal force is that the robot is crouching rapidly at this time, and so its CoM is accelerating rapidly downwards. The slipping ends at approximately $t = 1.4$s, which is when the robot starts to decelerate as it reaches the end of the crouch and begins to launch itself upwards. The tracking controller is back on track about 0.3s after the slipping stops. There is also a short period just before lift-off when the foot is slipping backwards. All three graphs end when lift-off is supposed to happen, which is a few milliseconds before it actually happens. This is why the normal force has not dropped all the way to zero at the end of the graph.

The most important aspect of this motion is the accuracy at the end of the trajectory. The final actual state variables are $\eta = (2.0586, -1.4918, -4.3584, 7.499)$ which are quite close to the desired values. This crouching motion is part of the first hop in the accompanying video.

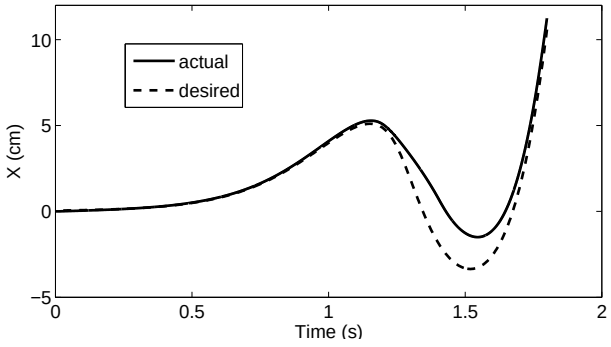

Fig. 4. Trajectory tracking during launching phase

D. Flight Phase

When the robot takes off from the ground, the ability to control its motion significantly decreases. During the flight phase, the angular momentum of the robot is constant, the CoM travels along a given trajectory which cannot be changed by the controller, and the robot has 3 degrees of underactuation. Also, in the flight phase, because of the mass difference between the robot's upper and lower legs (the lower leg is much lighter), applying a torque to the actuator causes much more movement to the lower leg than the upper one. This means that controlling the upper leg's motion is much more difficult than the lower leg's motion. For these reasons, the objective of the flight-phase control is limited to controlling foot placement at landing.

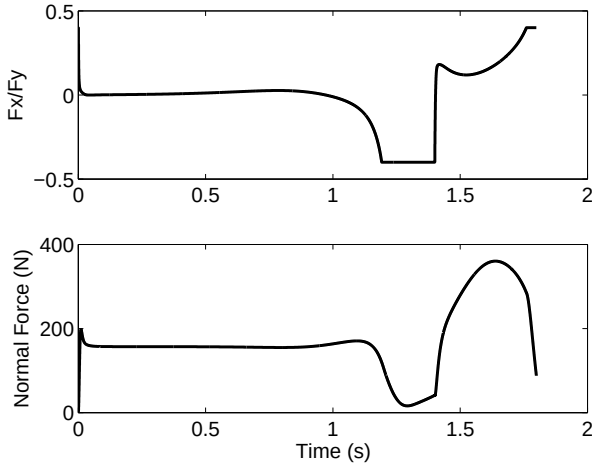

Fig. 5. Normal force and the ratio between the friction force and normal force

To control the robot during the flight phase, the trajectory-tracking controller is used to control the horizontal location of the foot so that it will land on a desired position with a desired velocity. So the first step is to define a reference trajectory for X during the flight phase. If X_G and x are the absolute horizontal displacement of the CoM and foot of the robot, respectively, then

$$X = X_G - x\,. \tag{19}$$

X_G is a given function of time right after the take-off instant, so by defining a desired trajectory for x, the desired value of X can be calculated easily. In this paper, a sigmoid function of the form $f(z) = 1/(1 + \mathrm{e}^{-z})$ is used as a reference function for x so that the forward velocity of the foot at the landing instant will be zero. This helps the robot not to slide (or to slide only a little). The actual trajectory for x is chosen to be

$$x = \frac{L_{\mathrm{H}}}{1 + e^{-(a_1 t + a_0)}}\,, \tag{20}$$

where L_{H} is the desired hop length, $a_0 = -6$, $a_1 = 12/T_{\mathrm{H}}$ and T_{H} is the flight-phase duration time.

Therefore, the desired trajectory for the foot during the flight phase depends on the hop length and the flight duration. Fig. 6 shows the controller's trajectory tracking performance during the flight phase for a hop with a desired length of 50cm. The flight phase duration is 0.280s. The small error at the moment of lift-off ($t = 0$ in this graph) is caused by the slipping of the foot during crouching, so that it is a little forward of where it should be at lift-off. Overall, the trajectory tracking performance is good, but there a small overshoot towards the end of the flight phase.

E. Landing and Rebalancing

When the foot regains contact with the ground, the controller switches from trajectory tracking to balancing, and brings the robot to a halt in the specified balanced

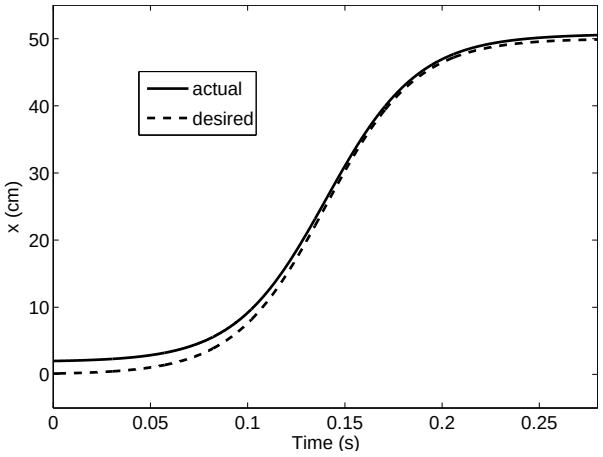

Fig. 6. Flight phase trajectory tracking

configuration, which does not have to be the same as the configuration at the beginning of the hop. Fig. 7 shows the position and velocity of the foot during the flight phase depicted in Fig. 6, the moment of landing, and the first 0.2 seconds of rebalancing. The sharp discontinuities in the $\dot{x}$ and $\dot{y}$ graphs occur at the instant of landing ($t = 0.28$s). It can be seen from these graphs that the forward velocity of the foot at this instant is close to zero, but the downward velocity is more than 2m/s. When the foot hits the ground, much of the lower leg's momentum is absorbed in the impact, but a small portion is converted to forward motion of the foot, which is why the foot slips forward a little immediately after landing. However, no further slipping occurs during the rest of the rebalancing maneuver.

The complete hop depicted in Fig. 4 to Fig. 7 is the first hop shown in the accompanying video (hop length 50cm). The second example in the video is a slightly longer hop (60cm). The slipping at lift-off and landing is too small to be seen in the video, but the slipping during crouch and the slight overshoot in hop length are both visible. (The correct landing spot is marked on the floor.)

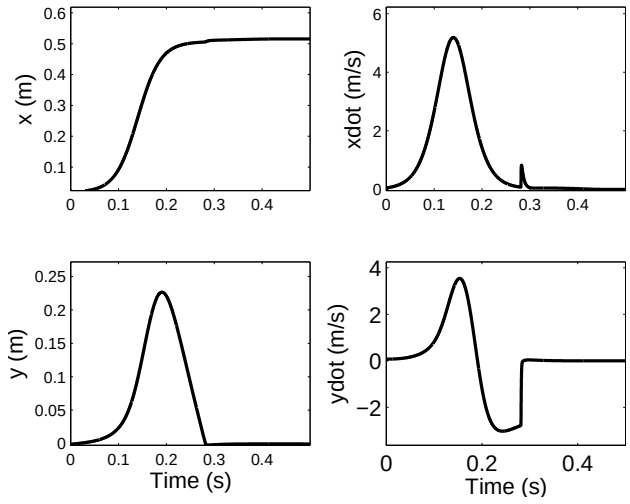

Fig. 7. Translational state variables and their derivatives

IV. Conclusions

In this paper a new and simple control algorithm for the control of a planar one-joint knee-leg hopping robot is introduced. This controller, which is based on the angular momentum of the robot about its contact point, is able to 1) stabilize the robot in any unstable balanced configuration in which both links are pointing or sloping upwards, 2) track a trajectory specifying deliberate deviations from a balanced configuration, and 3) control the robot during a complete hopping motion, starting and ending at a balanced configuration. The controller can also track a restricted class of trajectories without losing its balance, and it can tolerate significant amounts of slipping between the foot and the ground.

Since writing this paper, an improved version of this controller has been tested in simulations with significant sensor signal errors, including errors in sensing the overall orientation of the robot. Work is now under way to test the balancing controller on a real robot, and to extend it to balancing in 3D.

References

[1] M. Azad and R. Featherstone, Modeling the contact between a rolling sphere and a compliant ground plane, Australasian Conf. Robotics and Automation, Brisbane, Australia, 1-3 Dec 2010.

[2] M. Azad and R. Featherstone, Angular momentum based controller for balancing an inverted double pendulum, RoManSy 2012, Paris, France, 12-15 June.

[3] M. D. Berkemeier and R. S. Fearing, Sliding and hopping gaits for the underactuated acrobot, IEEE Trans. Robotics and Automation, Vol. 14(4), Aug 1998.

[4] M. D. Berkemeier and R. S. Fearing, Tracking fast inverted trajectories of the underactuated acrobot, IEEE Trans. Robotics and Automation, 15(4):740–750, Aug 1999.

[5] J. W. Grizzle, C. H. Moog and C. Chevallereau, Nonlinear control of mechanical systems with an unactuated cyclic variable, IEEE Trans. Automatic Control, Vol 50(5):559–576, May 2005.

[6] K. Hattori, H. Yamaura and K. Ono, Torque-based aerial posture control of a two-link acrobat robot, Proc. Institution of Mechanical Engineers Part I: Journal of Systems and Control Engineering, Vol. 218(17):595–601, Dec 2004.

[7] J. Hauser and R. M. Murray, Nonlinear controllers for nonintegrable systems: the acrobot example, Proc. 1990 American Control Conf., pages 669–671, SanDiego, CA, 23–25 May 1990.

[8] T. Mita, S. Hyon and T. Nam, Analytical time optimal control solution for a two-link planar acrobot with initial angular momentum, IEEE Trans. Robotics and Automation, Vol 17(3):361–366, June 2001.

[9] M. Miyazaki, M. Sampei, M. Koga and A. Takahashi, A control of underactuated hopping gait systems: acrobot example, 39th IEEE Conf. Decision and Control, Sydney, Australia, Dec 2000.

[10] E. Ohashi and K. Ohnishi, A hopping height control for hopping robot, Trans. Institute of Electrical Engineers of Japan, 124-D(7):660–665, July 2004.

[11] I. Poulakakis and J. W. Grizzle, The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper, IEEE Trans. Automatic Control, Vol. 54(8), Aug 2009.

[12] I. Poulakakis and J. W. Grizzle, Modeling and control of the monopedal robot thumper, IEEE Int. Conf. Robotics and Automation, Kobe, Japan, 2009.

[13] M. H. Raibert, Legged robots that balance, The MIT Press, Cambridge, Massachusetts, 1986.

[14] A. Sayyad, B. Seth and P. Seshu, Single-legged hopping robotics research–a review, Robotica, Vol. 25:587–613, 2007.

[15] X. Xin, T. Mita and M. Kaneda, The posture control of a two-link free flying acrobot with initial angular momentum, IEEE Trans. Automatic Control, Vol 49(7):1201–1206, July 2004.