Height Control for a One-Legged Hopping Robot using a Two-Dimensional Model

Kale Harbick and Gaurav Sukhatme

kale@gaurav@robotics.usc.edu

Robotic Embedded Systems Laboratory
Robotics Research Laboratories
Department of Computer Science
University of Southern California
Los Angeles, CA 90089-0781

Abstract

This paper describes a model-based height controller for a one-leg hopping robot. Equations of motion in two dimensions (i.e. the sagittal plane) are developed. These equations are solved and numerically integrated to produce an actuator command that will allow the robot to regulate its apex height.

I. INTRODUCTION

The hopping machine considered in this paper is like a pogo-stick; it has a small foot and a leg spring. Unlike a pogo-stick which uses a mechanical spring, the spring for the hopper is pneumatic.

The controller is based on Raibert’s three-part control system [1]. Raibert controlled hopping height by delivering a specified thrust value during stance. This paper describes a model-based height controller that allows a wide range of apex heights to be selected and achieved.

As shown in Figure 1, a hopping cycle consists of four phases. There are two flight phases, ascent and descent, and two stance phases, compression and extension. The height can be changed by setting the leg to an appropriate length during descent. Changing the leg length effectively changes the air spring constant. In this way, energy can be added to or removed from the system to change the apex height.

Proportional control has been used to regulate hopping height in simulation [2]. For the Monopod II robot, an adaptive energy feedback controller was used to control the apex energy and thus the apex height [3].

A discrete closed-form trajectory model was used to control a bow leg hopper [4]. Model parameters were experimentally determined with a least squares fit to a set of recorded trajectories.

A simulation of a vertical hopping machine was controlled using a near-inverse discrete-time model [5]. Time-varying or unknown parameters were estimated with a recursive least-squares parameter estimator. Because the model was able to update itself in this way, the controller was much less sensitive to modeling errors and even abrupt changes in physical parameters.

II. CONTROLLER

Table I defines the physical parameters of the hopper, Table II defines state variables of the system, and Table III defines the subscript notation. Superscripts of \(k\) denote the \(k\)th hopping cycle. A diagram of the hopper is shown in Figure 2. The leg stroke, hip offset, and mass parameters are similar to those used by Raibert for his planar hopper.

After the hopper transitions from extension to ascent (lift-off), the height controller uses the measured velocity of the body just before lift-off (\(\dot{y}_{k}^{\text{ext}}\)) to calculate the appropriate actuator setting for the leg. At apex, the leg length is set to this value, and held at this value until touch-down. During stance, no active actuator commands are used; it behaves as a passive air spring. If the controller functioned properly, the hopper should lift-off again and reach the desired apex height.

A. Initial Ascent

The flight phases can be described with simple ballistic equations. First the initial apex height must be calculated using the velocity just after lift-off and the lift-off height.

At lift-off the leg is at full extension, so the lift-off height of the body is

\[
y_{k, o}^{\text{ext}} = (L_{\text{max}} + d_{\text{hip}}) \sin \theta_{l_{o}}^{k}.
\]  

(1)
Fig. 1. Phase diagram of a hopping cycle.

Fig. 2. 2d hopping machine in stance. The body is not shown.
| $g$ | gravitational acceleration | $-9.81 \text{ m/s}^2$ |
| $\mu_k$ | viscous leg friction | $5.0 \text{ N/s}$ |
| $d_{\text{hip}}$ | distance from hip to body c.m. | $0.05 \text{ m}$ |
| $L_{\text{max}}$ | max leg length | $0.285 \text{ m}$ |
| $m_s$ | sprung mass | $8.375 \text{ kg}$ |
| $m_u$ | unsprung mass | $0.225 \text{ kg}$ |
| $A_p$ | area of piston | $4.91 \times 10^{-4} \text{ m}^2$ |
| $P_0$ | nominal pressure of upper leg chamber | $4.0 \times 10^2 \text{ kPa}$ |

**TABLE I**

**PHYSICAL PARAMETERS OF THE HOPPER.**

| $L$ | leg length (m) |
| $y$ | height of body c.m. (m) |
| $\dot{y}$ | body vertical velocity (m/s) |
| $\dot{x}$ | body horizontal velocity (m/s) |
| $K_s$ | spring constant (N·m) |
| $\theta$ | angle of leg measured from ground (rad) |
| $\dot{\theta}$ | angular velocity of leg (rad/s) |
| $I$ | moment of inertia of body w.r.t. foot during stance (kg·m$^2$) |

**TABLE II**

**VARIABLE DEFINITIONS.**

Now the apex is calculated to be

$$y_{\text{max}}^k = -\frac{(\dot{y}_{\text{lo}}^k)^2}{2g} + y_{\text{lo}}^k.$$

However, the velocity just before lift-off can be measured more accurately than $\dot{y}_{\text{lo}}^k$ on the real machine. This is because of the particular sensors chosen. The leg length is measured by a linear potentiometer, and the vertical acceleration of the body is measured by an accelerometer. During stance, the linear potentiometer measurement can be differentiated to obtain $\dot{y}$. During flight or stance, the accelerometer measurement can be integrated to obtain $\dot{y}$. Experiments with the sensors have shown that the linear potentiometer does not exhibit high-frequency noise, while the accelerometer is subject to bias and drift.

Conservation of linear momentum allows us to account for the inelastic collision between the upper leg's mechanical stop and the piston attached to the lower leg

$$\dot{y}_{\text{lo}}^k = \left(\frac{m_s}{m_s + m_u}\right) \dot{y}_{\text{lo}}^{k-}.$$

(2)

**B. Initial Descent**

The desired touch-down velocity depends on the apex height calculated in the previous section and the touch-down height

$$y_{\text{td}}^{k+1} = -\sqrt{2g(y_{\text{td}}^{k+1} - y_{\text{max}}^k)}.$$

The problem now arises that $y_{\text{td}}^{k+1}$ cannot be determined exactly. The touch-down height depends on the leg length at touch-down, which is exactly what we are trying to calculate. In order to approximate $y_{\text{td}}^{k+1}$, the leg length setting and touch-down angle for the previous cycle are used

$$y_{\text{td}}^{k+1} \approx (L_{\text{td}}^k + d_{\text{hip}}) \sin \theta_{\text{td}}^k.$$
C. Compression and Extension

The stance phase is more complex, because three forces affect the motion: gravity, friction in the leg, and the leg spring force

\[ m_s \frac{d\dot{y}}{dt} = \left( \frac{K_s}{L} - \mu \frac{dL}{dt} \right) \sin \theta + m_s g \]  \hspace{1cm} (3)

where the spring constant \( K_s = A \mu P_s L_{\text{max}}. \)

Since the foot does not move relative to the ground during stance, the velocity of the body center of mass is equal to the vertical component of the rate of change of leg length

\[ \dot{y} = \frac{dL}{dt} \sin \theta. \]  \hspace{1cm} (4)

Equations 3 and 4 can be combined to produce an equation of motion for compression and extension

\[ \frac{dL}{dy} = \frac{\dot{y}}{\left( \frac{K_s \sin \theta}{m_s L} - \mu \frac{dL}{m_s} + g \right) \sin \theta}. \]  \hspace{1cm} (5)

Because the previous equation also depends on \( \theta, \) we need one more to describe how \( \theta \) changes with respect to \( \dot{y}. \) At touch-down, the body velocity can be resolved into two components: one along the axis of the leg and one perpendicular to it. The perpendicular component causes the body to rotate around the foot with some angular velocity \( \dot{\theta} \)

\[ \dot{\theta}_{td}^{k+1} = \frac{s_{td}^{k+1} \sin (\pi - \theta_{td}^{k+1}) + y_{td}^{k+1} \cos (\pi - \theta_{td}^{k+1})}{L_{td}^{k+1}}. \]  \hspace{1cm} (6)

During stance, the angular momentum of the system varies less than 5%, so we can consider it to be conserved

\[ I \frac{d\theta}{dt} = m_s L^2 \frac{d\theta}{dt} \]  \hspace{1cm} (7)

\[ I_{td}^{k+1} \dot{\theta}_{td}^{k+1} = m_s \left( r_{td}^{k+1} \right)^2 \dot{\theta}_{td}^{k+1} \]  \hspace{1cm} (8)

\[ I \frac{d\theta}{dt} = I_{td}^{k+1} \dot{\theta}_{td}^{k+1}. \]  \hspace{1cm} (9)

Now combining Equations 7, 8, and 9

\[ \frac{d\theta}{dt} = \frac{\dot{\theta}_{td}^{k+1} \left( r_{td}^{k+1} \right)^2}{L^2}. \]  \hspace{1cm} (10)

Finally we combine Equations 4, 5, and 10

\[ \frac{d\theta}{dy} = \frac{\dot{\theta}_{td}^{k+1} \left( r_{td}^{k+1} \right)^2}{\left( \frac{K_s \sin \theta}{m_s L} - \mu \frac{dL}{m_s} + g \right) L^2}. \]  \hspace{1cm} (11)

Since the desired condition we are searching for is the leg length at touch-down, the integrations for compression and extension proceed backward in time. The initial velocity is \( \dot{y}_{lo} \), and the final velocity is \( \dot{y}_{td}^{k+1}. \) The initial leg length is
Since we cannot determine what \( \theta_{io} \) will be exactly for this final lift-off, we just use the value that was used in Equation 1. Integrating over velocity produces \( L_{td} \).

Since our initial condition for \( \theta \) was at lift-off, we need to use Equation 11 in a slightly different form

\[
\frac{d\dot{\theta}}{dt} = \frac{\dot{\theta}_{io}^{k+1} L_{max}^2}{\left(\frac{K_s \sin \theta}{m_v L} - \frac{u_k y}{m_v} + g\right) L^2}
\]

The expression for \( \dot{\theta}_{io}^{k+1} \) is similar to Equation 6:

\[
\dot{\theta}_{io}^{k+1} = \frac{\dot{x}_{io}^{k+1} \sin (\pi - \theta_{io}^{k+1}) + \dot{y}_{io}^{k+1} \cos (\pi - \dot{\theta}_{io}^{k+1})}{L_{max}}
\]

The vertical velocity \( \dot{y}_{io}^{k+1} \) is derived in the next section. The horizontal velocity \( \dot{x}_{io}^{k+1} \) cannot be determined exactly, so to approximate we just use the measured value from the previous cycle. Again the previous cycle’s lift-off angle is used to approximate \( \dot{\theta}_{io}^{k+1} \)

\[
\dot{\theta}_{io}^{k+1} \approx \frac{\dot{x}_{io}^{k+1} \sin (\pi - \theta_{io}^{k+1}) + \dot{y}_{io}^{k+1} \cos (\pi - \theta_{io}^{k+1})}{L_{max}}
\]

**D. Final Ascent**

The desired velocity after the final lift-off can be calculated from the lift-off height and desired apex

\[
\dot{y}_{io}^{k+1} = \sqrt{2g(\dot{y}_{io}^{k+1} - \dot{y}_{max}^{k+1})}
\]

where

\[
\dot{y}_{io}^{k+1} = (L_{max} + d_{hip}) \sin \dot{\theta}_{io}^{k+1}
\]

By rearranging Equation 2, we can again compensate for the inelastic collision at lift-off, this time going backwards in time to find \( \dot{y}_{io}^{k+1} \)

\[
\dot{y}_{io}^{k+1} = \frac{L_{max} + d_{hip}}{m_v} \dot{y}_{io}^{k+1}
\]

Now that we have the initial conditions, the integration can proceed. This integration yields the leg length setting to be used.

**E. PD Controller**

In the simulation, a PD controller is used to command the leg length to the desired position:

\[
\tau = -K_P (L - L_d) - K_D \dot{L}
\]

where \( \tau \) is the leg force command, and \( K_P \) and \( K_D \) are proportional and derivative gains. Values for \( K_P \) and \( K_D \) used in the simulations are 1300.0 N/m and 50.0 N/(m/s), respectively.

**Acknowledgments**

The Vortex physics simulation toolkit developed by CM Labs was used for the simulations discussed in this paper.

**References**