FAST AND ADAPTIVE HOPPING HEIGHT CONTROL OF SINGLE-LEGGED ROBOT

Jawaad Bhatti, Andrew R. Plummer, M. Necip Sahinkaya and Pejman Iravani
Department of Mechanical Engineering
University of Bath
Bath, UK

Emanuele Guglielmino and Darwin G. Caldwell
Department of Advanced Robotics
Istituto Italiano di Tecnologia
via Morego, 30
16163 Genova, Italy

KEYWORDS
Adaptive hopping height control; running; hopping; legged robot

ABSTRACT
Research on running robots has generally focussed on the steady-state. When the ground has limited foot placement surfaces or there are sudden changes in height then steady-state running is not possible. It becomes necessary to make step-by-step adjustments to place the foot.

In this paper a mass-spring-damper model of a robot’s leg is used to develop a hopping controller capable of meeting rapid changes in demand height or flight time. Analysis of the model provides a simple method to select control parameters for effective height control without tuning or iteration. Additionally, a simple adaptive algorithm is introduced and demonstrated in simulation. The adaptive control algorithm allows rapid changes of height even when ground characteristics change. Experimental validation is ongoing and some preliminary results are provided.

INTRODUCTION
Mobility in robots is mostly achieved using wheels or tracks. However if a mobile robot has to be designed for locomotion over rough and unstructured terrains these technologies suffer from inherent limitations. This has motivated research into bio-inspired legged locomotion [1], [2].

Robots such as Boston Dynamics’ BigDog are now famous for their somewhat unsettling ability to traverse rough terrain and maintain balance [3]. These successful robots are the direct result of an approach to the problem of legged robot locomotion pioneered by Marc Raibert at MIT in the 1980s [4]. Raibert began by tackling the problem of a single-legged hopping robot [5]. His solution was to decouple the problem into a three part controller: height, speed and body angle. The single-legged controller was then successfully extended for bipeds and quadrupeds. The single-legged case is an important one as it provides a relatively simple platform to develop control techniques for application on multi-legged robots.

Most research published on single-legged robot control has however focussed on the steady-state running case with variations in leg design and actuation methods [6]. Steady-state running is not possible when the ground has limited foot placement surfaces or sudden changes in height. In that case it becomes necessary for a single-legged hopping robot to land its foot from one spot to the next. The focus of this paper is controlling the hopping height, one of the three parts of a single-legged robot controller, to meet sudden changes in demand. The problem of foot placement has previously been addressed in [7]. Two methods for varying the ‘step length’, the length from one foot placement spot to the next, were investigated using a 2D planar hopping model:

- Maintaining a constant forward speed and varying the hopping height from one step to the next.
- Maintaining a constant hopping height and varying the forward speed.

The height variation method used in [7] was found to produce less accurate foot placement than their alternative forward speed variation method. In this paper we contribute a height control algorithm based on an analysis of a mass-spring-damper model of the hopping leg. This provides an insight into
which variables may be more appropriate for control purposes. Additionally, the height controller developed in this paper is adaptive. Other adaptive hopping height control methods have been published but are more complex and aim to eliminate only steady state errors [8–10].

NOMENCLATURE

c  Leg damping coefficient.
\( c_{gr} \)  Ground damping coefficient.
\( C_r \)  Coefficient of restitution.
F  Ground force.
\( F_0 \)  Reference ground force.
g  Gravitational acceleration.
h  Height of mass (Figure 1).
h_b  Body height.
h_f  Foot height.
k  Leg spring stiffness.
\( K_1, K_2 \)  Controller gains.
\( K_L \)  Controller gain for losses.
\( K_\Delta \)  Controller gain for changes.
m_p  Body mass.
m_f  Foot mass.
p  Lift-off or touch-down speed.
t  Time.
t_{td}  Time at touch-down.
T_s  Stance time – time spent on ground in each hop.
T_f  Flight time – time of flight phase of hop.
t_{cd}  Time when leg contacts ground.
v  Actuator extension velocity.
y  Actuator displacement (Figure 1).
\( \delta_0 \)  Reference ground penetration.
\( \zeta \)  Damping ratio.
\( \omega_d \)  Damped frequency, \( \omega_d = \omega_n \sqrt{1 - \zeta^2} \).
\( \omega_n \)  Natural frequency.

HEIGHT CONTROL ANALYSIS

A simplified model of a hopping leg is shown in Figure 1. The model consists of a mass, actuator and spring-damper. The unstrained length of the spring \( L \) does not affect the dynamics of hopping so can be set to \( L = 0 \) in order to simplify the equation of motion giving:

\[
\ddot{h} = -g - \left( 2\zeta \omega_n (\dot{h} - \dot{y}) + \omega_n^2 (h - y) \right)
\]

for \( h - y < 0 \):
\[
\dot{h} = -g
\]

otherwise:
\[
\dot{h} = -g
\]

If the actuator is kept stationary \( y(t) = 0 \) and the mass is dropped from an initial height the mass will bounce and will lose energy on each hop. In order to maintain hopping the actuator needs to put energy into the system. A simple way to deliver energy on each hop is to extend the actuator at a constant velocity \( v \) during the stance phase. The actuator then retracts during the flight phase back to its starting position ready for the next hop as shown in Figure 2. The actuator has to maintain its motion throughout the stance phase which can be estimated to be half the period of the mass-spring-damper: \( T_s \approx T_s' = \pi / \omega_d \). The estimated stance time \( T_s' \) may be less than the true stance time \( T_s \) so the actuator motion is maintained for 1.5 times longer than the estimate \( 1.5T_s' \) to ensure motion throughout the stance phase. The resulting motion in the stance phase can be analysed by neglecting gravity \( g = 0 \). Equation (1) for the stance phase then becomes:

\[
\ddot{h} + 2\zeta \omega_n (\dot{h} - \dot{y}) + \omega_n^2 (h - y) = 0
\]
Beginning at touch-down with an impact speed $p$ gives the following:

- $y = vt$
- $h(0) = 0$
- $h(0) = \Delta p$

Resulting in the solution:

$$h = vt - e^{-\frac{-\pi t}{\omega_d}}\left(\frac{p + v}{\omega_d}\right)\sin\omega_d t \tag{3}$$

Lift-off occurs at $t = T'_L$. This means that the lift-off speed, which is the same as the next touch-down speed, will be:

$$\dot{h}(T'_L) = p + \Delta p = C_R(p + v) + v \tag{4}$$

Where $C_R$ is a coefficient of restitution. It is the proportion of the impact speed to which the take-off speed is reduced when there is no actuator motion $v = 0$ and is given by:

$$C_R = \frac{1 - \Delta}{\sqrt{1 - \xi}} \tag{5}$$

Equation (4) can be rearranged for $v$ to give:

$$v = K_1p + K_2\Delta p \tag{6}$$

Where:

$$K_1 = \frac{1 - C_R}{1 + C_R} \tag{7}$$
$$K_2 = \frac{1}{1 + C_R}$$

Equation (6) describes the actuator velocity $v$ required to produce a given change in vertical speed $\Delta p$ upon lift-off, given the touch-down speed $p$. It should be noted that $v$ can take a negative value. This results in the leg contracting to remove energy from the system to bring the robot to a lower height than would be otherwise possible with damping alone. During a flight phase if the most recent lift-off speed $p_0$ is known and the next desired lift-off speed is $p_d$ then the control logic becomes:

$$v = K_1p_0 + K_2(p_d - p_0) \tag{8}$$

The constants in equation (8) can be set:

- By using the coefficient of restitution so $K_1 = K_L$ and $K_2 = K_3$.
- By varying $v$ and finding the best-fit values of $K_1$ and $K_2$.
- By an adaptive process.

Motion during the flight phase is simply parabolic so the relationships $p_0 = \sqrt{2gh_0}$ and $p_0 = \frac{1}{2}gT_{f0}$ may be substituted into (8) giving the controller in terms of apex height or flight time if desired:

$$v = K_1\sqrt{h_0} + K_2\left(\sqrt{h_d} - \sqrt{h_0}\right) \tag{9}$$

$$v = K_1T_{f0} + K_2(T_{fd} - T_{f0}) \tag{10}$$

**STEADY-STATE CONTROLLER**

The steady-state relationship between $v$ and $p$ can be found from (6) by substituting $\Delta p = 0$:

$$p_{ss} = \frac{v}{K_L} \tag{11}$$

The control logic then becomes:

$$v = K_1p_d \tag{12}$$

As before, (12) may be written in terms of $\sqrt{h_d}$ or $T_{fd}$. It can also be seen that (12) contains no feedback of any variables. This type of height controller is almost open-loop, requiring only the synchronisation of actuator outputs to touch-down events.

The above analysis neglects gravity during the stance phase. The effect of doing so can be seen in Figure 3 which compares analytical results (line) against simulations which include gravity (crosses). Simulations of (1) with the parameters in Table 1 are run for different constantly held values of actuator velocity $v$ to find the corresponding steady-state lift-off/touch-down speeds $p_{ss}$. The closeness of the simulation and analytical results suggests that the assumption of negligible gravity during stance was reasonable.

<table>
<thead>
<tr>
<th>Table 1: Parameters for model 1 simulations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1 Parameter</td>
</tr>
<tr>
<td>$\omega_n$</td>
</tr>
<tr>
<td>$\xi$</td>
</tr>
<tr>
<td>Initial $h$</td>
</tr>
<tr>
<td>Initial $\dot{h}$</td>
</tr>
</tbody>
</table>
Figure 3: Model 1 steady-state $p_{SS}$ and $v$ relationship. Crosses are simulation including gravity; line is analytically derived ignoring gravity $p_{SS} = \frac{v}{k_L} K_L = 0.157$.

Figure 4 shows the results of applying the steady-state control logic (12) to a varying height demand. The apex height demand $h_d$ is held constant at 0.1 m for the first 7 hops, increased to 0.12 m for hops 8 to 12, varied randomly between 0.04 m and 0.14 m for hops 13 to 22 and then kept constant at 0.12 m for the rest. It can be seen that during periods of constantly held demand the hopping height is converging to a constant value although with steady-state error because the controller gain $K_1 = K_L$ was selected based on an analysis ignoring gravity. It can clearly be seen that this controller is not able to track a rapidly changing height demand so would not be suitable for controlling foot placement. It does nevertheless have the advantage of requiring no height feedback so is almost open-loop, requiring only the synchronisation of actuator motion with touch-down.

**ADAPTIVE DYNAMIC CONTROLLER**

Figure 5 shows the results of simulating model 1 using the control logic (9). Gain values have been selected using (7) giving $K_1 = K_L \sqrt{2 g}$ and $K_2 = K_d \sqrt{2 g}$. In comparison with the steady-state controller results seen in Figure 4, the dynamic controller has better performance tracking changes in demand but also during steady-state. The error that remains may be reduced by introducing an adaptive logic.

If the apex heights, flight periods or lift-off speeds for previous hops are known then these can be used to determine controller gains. Writing (8) for the previous two hops in matrix form gives:

$$
\begin{bmatrix}
    p_1 \\
    p_2
\end{bmatrix}
- \begin{bmatrix}
    \Delta p_1 \\
    \Delta p_2
\end{bmatrix} \begin{bmatrix}
    K_1 \\
    K_2
\end{bmatrix} = \begin{bmatrix}
    v_0 \\
    v_1
\end{bmatrix}
$$

(13)

Where $\Delta p_1 = p_0 - p_1$ and $\Delta p_2 = p_1 - p_2$. The most recent lift-off speed is $p_0$ and $p_1$ the one before that and so on. Solving for $K_1$ and $K_2$ requires matrix inversion which is not possible to do accurately if the determinant is close to zero:

$$
\Rightarrow \left| \frac{\Delta p_1}{p_1} - \frac{\Delta p_2}{p_2} \right| \approx 0
$$

(14)

Ill-conditioned inversion can be avoided by setting a threshold for inversions $\left| \frac{\Delta p_1}{p_1} - \frac{\Delta p_2}{p_2} \right| > 0.01$. If below the threshold then the control parameters are kept the same as before. In simulation it was also found that the controller can fail if solving (13) results in $K_2 \approx 0$. A solution then is to keep the previous value of $K_2$. Figure 6 shows results when the adaptive logic is added to the dynamic controller. Initial values of controller gains are selected to be poor. This results in hops 2 and 3 with large errors. The adaptive logic then acts to determine the correct controller gains and reduce height tracking error even with a rapidly changing demand.
The performance of the adaptive dynamic controller on changing ground properties was investigated using model 2 shown in Figure 7. Model 2 is similar to model 1 but includes a foot mass $m_f$ in addition to the body mass $m_b$ and a non-rigid ground. Here the ground which is a hard material is modelled as a non-linear spring-damper [11]. The equations of motion are:

$$F(h_f, h_f) = F_0 \left( \frac{h_f}{\delta_0} \right)^2 + c_{gf}(-h_f)$$

$$x = h_b - y - h_f \quad (15)$$

$$m_b \ddot{h}_b = -m_b g + c \ddot{x} - kx$$

$$m_f \ddot{h}_f = -m_f g + c \ddot{x} + kx + F(h_f, h_f) [h_f < 0]$$

The results of simulating model 2 using the parameters in Table 2 with a constant height demand are shown in Figure 8. It can be seen that after each change in ground properties the controller adapts within a few hops. This is the case even if demand height is randomly varied between 0.05 m and 0.15 m as shown in Figure 9.
PRELIMINARY EXPERIMENTAL WORK

Work is ongoing to experimentally test the controllers developed in this paper. A suitable rig for doing so has been constructed and is shown in Figure 10. The test rig consists of a 2-link hydraulic leg from the HyQ robot [2] mounted at the end of a pivoting beam. Joints at the hip and knee are actuated by hydraulic actuators controlled through proportional valves. A real-time industrial controller outputs two control voltages to an amplifier which supplies a proportional current to the hydraulic valves.

Sensors include an accelerometer at the top of the leg and position encoders at the hip and knee joints. A springy foot has been mounted to the end of the leg.

Control of the leg is achieved using the following procedure:

- Ordinarily proportional control is used to bring the leg to a nominal zero-position.
- When the measured acceleration exceeds a set threshold value, the stance mode is triggered for a fixed time.
- Throughout stance mode the voltage signals to the valves are held at fixed values. These voltages are determined upon initiation of stance mode.

The voltages applied during stance mode are in a ratio that results in a downward velocity of the actuator from the zero-position. A control input $\alpha$ is then used to determine their magnitude. For example:

$$\begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \alpha \begin{pmatrix} 0.48 \\ 1 \end{pmatrix}$$

Control is now possible by varying the control input $\alpha$ as a function of the flight times and demanded flight times. This is sufficient to implement an adaptive controller as in (10). Work is ongoing to obtain results with the adaptive controller but some preliminary results for steady-state hopping are included. Figure 11 shows some initial results when the control input $\alpha$ is kept constant while the hopping leg is allowed to reach steady-state. Two sets of data were obtained, one on a concrete floor and one on a cushioned floor.

[These results show a linear relationship between the control input and the flight time similar to analytical and simulation work (Figure 3).]

Comparing simulation (Figure 3) with experiment (Figure 11) it is can be seen that the relationship between the control input and steady-state hopping is not linear for the experimental results. There could be a number of reasons for this:

- Non-linear leg elasticity and/or damping.
- Dynamics of hydraulics.
- Dynamics of articulated leg and rig.

These remain to be investigated. It can also be seen that the hopping flight time and hence height is affected by ground properties.

Table 2: Model 2 simulation parameters.

<table>
<thead>
<tr>
<th>Model 2 Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_b$</td>
<td>10 kg</td>
</tr>
<tr>
<td>$m_f$</td>
<td>1 kg</td>
</tr>
<tr>
<td>$k$</td>
<td>8000 N/m</td>
</tr>
<tr>
<td>$c$</td>
<td>30 N m/s</td>
</tr>
<tr>
<td>Initial $h_b$</td>
<td>0.15 m</td>
</tr>
<tr>
<td>Initial $h_f$</td>
<td>0.15 m</td>
</tr>
<tr>
<td>High stiffness $F_0$, $\delta_0$, $c_{gr}$</td>
<td>10000 N, 0.01 m, 10 N m/s</td>
</tr>
<tr>
<td>Low stiffness $F_0$, $\delta_0$, $c_{gr}$</td>
<td>100 N, 0.01 m, 10 N m/s</td>
</tr>
</tbody>
</table>

Figure 10: Experimental hopping leg rig.

Figure 11: Steady-state experimental results.
CONCLUSION

Previous work on height control has focused on steady-state running. In this paper a mass-spring-damper model was used to provide insight into the problem of controlling a hopping robot to meet rapid changes in demand apex height or flight time. A relatively simple adaptive control algorithm based on analysis of the model was developed. Simulations testing the control algorithm along with preliminary experimental work show promise that the adaptive controller can be implemented. Further experimental work will investigate the effectiveness of the adaptive controller using an articulated hydraulic robot leg.

Although in this paper the velocity of the actuator extension was used as the control input, it may be the case that the algorithms would also work for other control inputs. For example one method of controlling vertical motion is to have a leg design where a thrust force can be applied during stance for different amounts of times in order to impart a desired impulse in the vertical direction. If controlling the vertical impulse is analogous to extending an actuator at a fixed velocity then the control algorithms developed in this paper may be used in several different robot designs. This remains to be investigated.

Future work will involve validating the hopping height controller with experimental results and extension of the controller to take into account the effect of a non-vertical leg angle which is necessary when running forwards. In the longer term, it is intended to make progress towards a robot capable of navigating across limited stepping surfaces quickly. This will require extending the controller to function in the context of a 3D robot with multiple legs.

REFERENCES


