In *Proceedings of the IEEE Intl. Conf. on Robotics and Automation*
vol. 4, pp. 3998-4003, Washington, D.C., May 2002

Controlling Hopping Height of a Pneumatic Monopod

Kale Harbick and Gaurav S. 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

We describe a model-based height controller for a hopping robot with a pneumatically powered leg. The controller explicitly models variation in the leg angle and height. Using an explicit model of the physics of the pneumatic spring and some symmetry assumptions, we derive the desired leg-length setting to regulate apex hopping height using a PD controller. Simulation experiments of hopping in the sagittal plane show reasonable height regulation. For low-speed running, we take advantage of the small variations in the leg angle about the vertical, and demonstrate that the original symmetry assumptions may be relaxed by restricting the leg angle to $\pi/2$. *Simulations show that the restricted model outperforms the original model by a small but significant amount.*

I. Introduction

The benefits of legged systems over other forms of terrestrial locomotion are obvious; they can navigate obstacles which wheeled and treaded vehicles cannot. Dynamically-stable legged systems have advantages over statically-stable legged systems. Running machines can travel faster and can navigate terrain which has points of support that are spaced too far apart for walking machines to reach. Even for the simplest running machine, a one-legged hopper, research problems have not been completely explored.

Most running machines constructed thus far have served as little more than existence proofs of different forms of locomotion. We seek to contruct a running machine that builds on basic locomotion with additional behaviors that would be useful in a more functional robotic system. To this end we have studied ways to improve the Raibert three-part control system [1].

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 force curve for a mechanical spring is different from a pneumatic spring; therefore the model must take that into account. The machine analyzed here is a planar hopper, i.e. its motion is restricted to the sagittal plane.

The controller is based on Raibert's three-part control system [1]. This method decomposes the control into three separate control loops: forward velocity control, body attitude control, and height control. Raibert controlled hopping height by delivering a specified thrust value during stance. The hopper would repeatedly reach a specific apex height, where the energy injected during stance would equal the energy lost due to friction and air resistance. The experiments in this paper utilize the Raibert forward velocity and body attitude controllers; only the height controller is changed.

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.

The height controller is the most difficult of the three. The forward velocity controller acts only during flight to position the foot in preparation for landing. The body attitude controller acts only during stance to try and keep the body level. The height controller tries to make the body reach a specified apex height (halfway through the flight phase), but the control inputs to do this must be finished by the end of stance. Control inputs during flight before the apex is reached will have little effect on the apex height achieved.

This paper develops two model-based height controllers. The first models flight and stance physics in the sagittal plane, and the second takes advantage of the small variation in leg angle during stance to form a simplified model. We report results of height regulation experiments with both controllers and evaluate their performance. Simulations show that the second, simplified, model runs without certain symmetry assumptions needed in the first model, and marginally outperforms it.

The paper is organized as follows. Related work is discussed in Section II. Section III describes the model-based height controllers. Experimental results are presented in Section IV. Conclusions and future work are discussed in Section V.

II. Related Work

A three-part control system has been used to control monopod hopping robots [1]. One of the controllers used a fixed duration thrust to regulate hopping 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.

The controllers presented in this paper use trajectory models that utilize the differential equations of motion rather than discrete functions. The equations of motion are numerically integrated to produce appropriate actuator commands. This approach doesn't require experimental determination of parameters, and unlike tabular control doesn't require storage of a large table of values.

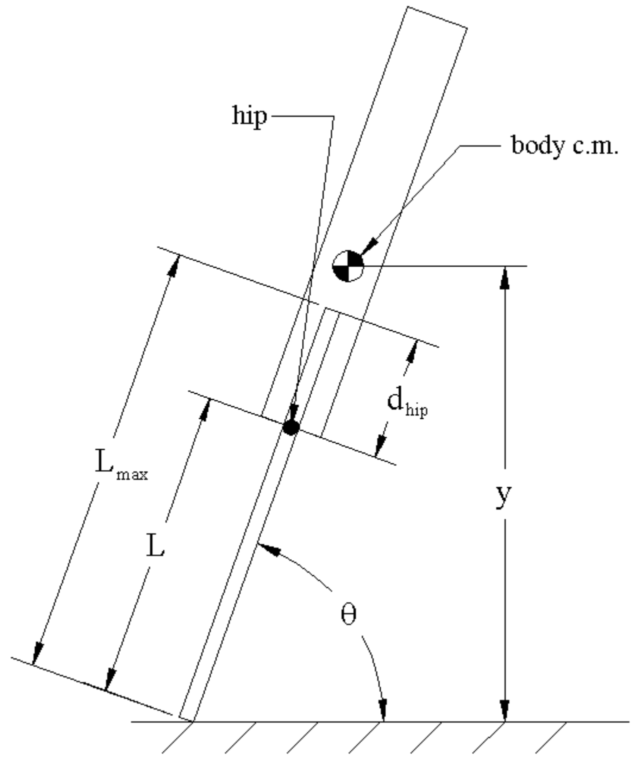


Fig. 2. 2D hopping machine in stance. The body is not shown.

td	touch-down
lo	lift-off
$+$	immediately after
$-$	immediately before
d	desired value

TABLE I
SUBSCRIPT DEFINITIONS.

III. Height Controllers

A diagram of the hopper model is shown in Figure 2. Table I defines the subscripts used for the state variables, Table II defines the state variables themselves, and Table III defines the physical parameters of the hopper. Superscripts of k denote the kth hopping cycle. The leg stroke, hip offset, and mass parameters are similar to those used by Raibert for his planar hopper [1].

The detailed derivations for the equations presented in this section can be found in [6] and [7].

The control loop functions as follows. At lift-off, the controller solves equations of motion for the next five phases: initial ascent, descent, compression, extension, and final ascent. The goal is to control the apex height y_{max}^{k+1} reached at the end of the final ascent phase. The controller uses state information at the beginning of initial ascent (lift-off) and the goal apex height to calculate the proper leg length setting. As the hopper transitions from initial ascent to descent, the leg setting is realized with a PD controller

$$\tau = -K_P\,(L - L_d) - K_D\dot{L}$$

where τ 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.

This leg position is maintained until touch-down, at which point the actuator releases and the passive dynamics of stance take over. At lift-off the cycle begins again.

A. Two-Dimensional Model

This model incorporates the leg angle and lateral velocity effects. First we define the spring constant K_s

$$K_s = A_p P_s L_{max}. \tag{1}$$

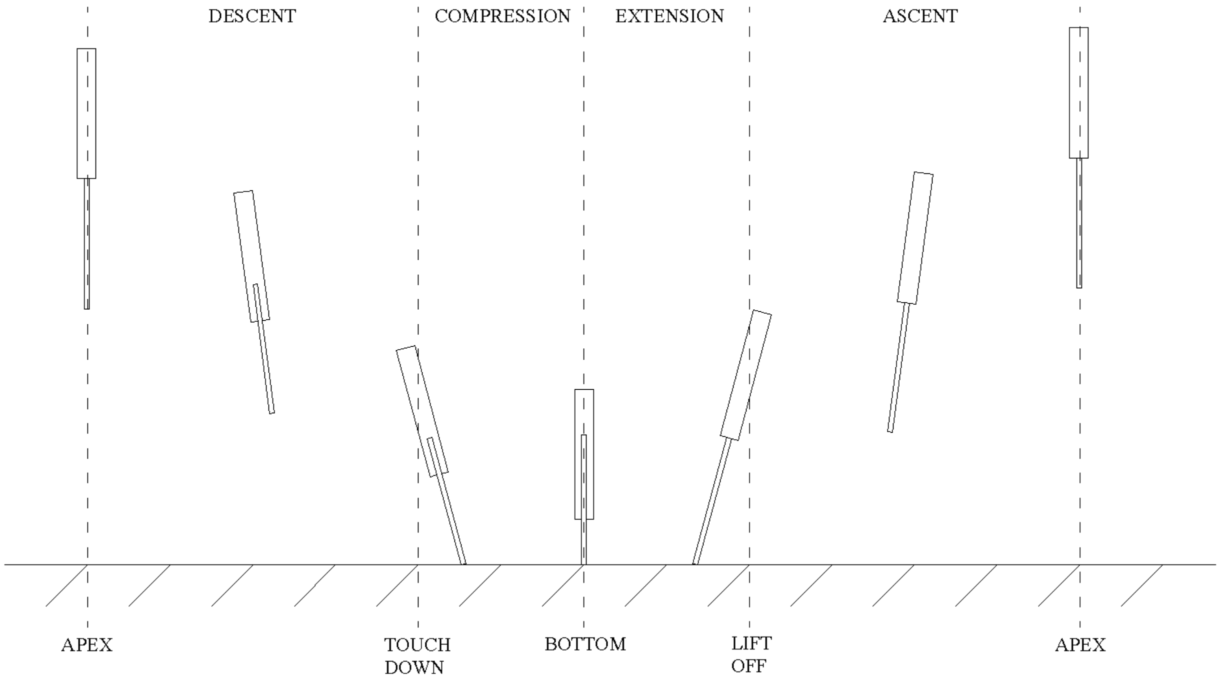


Fig. 1. Phase diagram of a hopping cycle.

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 ($N{\cdot}m$)
θ	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 ($kg{\cdot}m^2$)

TABLE II
VARIABLE DEFINITIONS.

g	gravitational acceleration	$-9.81\ m/s^2$
μ_k	viscous leg friction	$5.0\ N/s$
d_{hip}	distance from hip to body c.m.	$0.05\ m$
L_{max}	max leg length	$0.285\ m$
m_s	sprung mass	$8.375\ kg$
m_u	unsprung mass	$0.225\ kg$
A_p	area of piston	$4.91 \times 10^{-4}\ m^2$
P_0	nominal pressure of upper leg chamber	$4.0 \times 10^2\ kPa$

TABLE III
PHYSICAL PARAMETERS OF THE HOPPER.

There are two equations of motion for compression and extension

$$\frac{dL}{d\dot{y}} = \frac{\dot{y}}{\left(\frac{K_s \sin\theta}{m_s L} - \frac{\mu_k \dot{y}}{m_s} + g\right)\sin\theta} \tag{2}$$

$$\frac{d\theta}{d\dot{y}} = \frac{\dot{\theta}_{lo}^{k+1} L_{max}^2}{\left(\frac{K_s \sin\theta}{m_s L} - \frac{\mu_k \dot{y}}{m_s} + g\right) L^2} \tag{3}$$

where

$$\dot{\theta}_{lo}^{k+1} = \frac{\dot{x}_{lo^-}^{k+1} \sin(\pi - \theta_{lo}^{k+1}) + \dot{y}_{lo^-}^{k+1} \cos(\pi - \theta_{lo}^{k+1})}{L_{max}}.$$

The vertical velocity $\dot{y}_{lo^-}^{k+1}$ is calculated below. Because of the simplifications made in this model, we cannot determine $\dot{x}_{lo^-}^{k+1}$ and θ_{lo}^{k+1}, so we invoke symmetry considerations and use the measured values from the previous hop

$$\dot{\theta}_{lo}^{k+1} \approx \frac{\dot{x}_{lo^-}^{k} \sin(\pi - \theta_{lo}^{k}) + \dot{y}_{lo^-}^{k+1} \cos(\pi - \theta_{lo}^{k})}{L_{max}}.$$

Equations 2 and 3 are integrated backwards in time from lift-off to touch-down. Initial conditions at lift-off are given by

$$L_{lo} = L_{max}$$

$$\dot{y}_{lo^-}^{k+1} = \left(\frac{m_s + m_u}{m_s}\right) \dot{y}_{lo^+}^{k+1}$$

where

$$\dot{y}_{lo^+}^{k+1} = \sqrt{2g(y_{lo}^{k+1} - y_{max}^{k+1})}$$

and

$$y_{lo}^{k+1} = (L_{max} + d_{hip}) \sin\theta_{lo}^{k+1}.$$

We also need a final condition for $\dot{y}$ at touch-down

$$\dot{y}_{td}^{k+1} = -\sqrt{2g(y_{td}^{k+1} - y_{max}^{k})}$$

where

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

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

and

$$y_{lo}^{k} = (L_{max} + d_{hip}) \sin\theta_{lo}^{k}.$$

As described in [7], the desired height at touch-down y_{td}^{k+1} is approximated by

$$y_{td}^{k+1} \approx \left(L_{td}^{k} + d_{hip}\right) \sin\theta_{td}^{k}$$

where L_{td}^{k} is the leg length of the previous touch-down, and θ_{td}^{k} is the touch-down angle in the previous cycle.

Equations 2 and 3 are integrated over $\dot{y}$ to find L_{td}^{k+1}, our desired leg length setting. A Runge-Kutta algorithm described in [8] is used to perform the numerical integration, and converges in six or seven steps to a solution within a 0.01% relative error tolerance.

At the top lateral velocities studied in these experiments, the leg sweeps through a rather small angle during stance, approximately ± 15 degrees. The next section makes use of this fact, producing a simpler model. This is accomplished by setting $\theta = \pi/2$.

B. *Reduced One-Dimensional Model*

We now model only the vertical motion of the hopper, and ignore the effects of the leg angle and lateral velocity. The spring constant K_s is defined in Equation 1.

The equation of motion for compression and extension is given by

$$\frac{dL}{d\dot{y}} = \frac{\dot{y}}{\frac{K_s}{m_s L} - \frac{\mu_k \dot{y}}{m_s} + g}. \tag{4}$$

This equation is integrated backwards in time from lift-off to touch-down. Initial conditions at lift-off are given by

$$L_{lo}^{k+1} = L_{max}$$

$$\dot{y}_{lo^-}^{k+1} = \left(\frac{m_s + m_u}{m_s}\right) \dot{y}_{lo^+}^{k+1}$$

where

$$\dot{y}_{lo^+}^{k+1} = \sqrt{2g(y_{lo} - y_{max}^{k+1})}$$

and

$$y_{lo} = L_{max} + d_{hip}. \tag{5}$$

We also need a final condition for $\dot{y}$ at touch-down

$$\dot{y}_{td}^{k+1} = -\sqrt{2g(y_{td}^{k+1} - y_{max}^{k})}$$

where

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

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

and y_{lo} is given in Equation 5.

As described in [6], the desired height at touch-down y_{td}^{k+1} is approximated by

$$y_{td}^{k+1} \approx L_{td}^{k} + d_{hip}$$

where L_{td}^{k} is the leg length of the previous touch-down.

	Desired Apex Height (m)	Avg. Abs. Error (cm)	Avg. Rel. Error (%)
2D Model	0.5	0.62	1.24
	0.45	0.40	0.89
1D Model	0.5	0.21	0.42
	0.45	0.13	0.28

TABLE IV

ABSOLUTE AND RELATIVE ERRORS IN APEX HEIGHT.

Equation 4 is integrated over $\dot{y}$ to find L_{td}^{k+1}, our desired leg length setting. A Runge-Kutta algorithm described in [8] is used to perform the numerical integration, and converges in six or seven steps to a solution within a 0.01% relative error tolerance.

IV. EXPERIMENTS

Two sets of experiments were performed in simulation. Each experiment was run twice, once for each height controller.

The first set involves only vertical hopping. The hopper was dropped from a height of $0.75\ m$ and given desired apex heights in a piece-wise constant fashion. Figure 3 shows the results of these experiments. Note that going from a high to low height takes many more hops than going from low to high. This is a consequence of our actuator model. Note that having the leg at maximum extension will produce the smallest possible spring constant, assuming the quantity of air in the cylinder remains constant. If the hopper continues to hop with this leg extension, energy will be gradually lost to friction. This is what happens when we go from high to low apex heights. If friction were greater, the system would lose energy faster. Increasing energy is easier to do, by simply shortening the leg, thereby creating a larger spring constant. In a real robot, exhausting air from the actuator to provide a smaller range of spring constants would allow faster convergence from high to low apex heights.

For the second set, the hopper was given a constant desired apex height while the desired lateral velocity was slowly ramped up. The results of trials with two different desired apex heights are shown in Figure 4. Average absolute errors and average relative errors in apex height are shown in Table IV.

V. CONCLUSIONS AND FUTURE WORK

Table IV shows that the restricted 1D model actually performs better than the 2D model. Although the 2D model is closer to the actual physics of the system than the 1D model, there are more unknown boundary conditions for the stance phase in the 2D model. This leads to making more assumptions than we had to do for the 1D model. We hypothesize that the error introduced in these assumptions outweighs the benefit gained by additional fidelity to the true physics in the model itself.

We have shown that both the 2D model and simplifed 1D model can accurately regulate apex height on the simulated hopper with a variety of lateral speeds. Since the model solutions both converge so quickly, implementation on a real robot is feasible.

Future work in simulation includes 3D simulations and the introduction of sensor and actuator noise. We will also test the height controllers on a real hopping robot, which is being constructed.

Acknowledgments

The authors would like to thank Michael Poole for his work on designing and constructing the physical robot frame, presently under development. The Vortex physics simulation toolkit developed by CM Labs was used for the experiments reported in this paper.

REFERENCES

[1] Marc Raibert, *Legged Robots that Balance*, MIT Press, Cambridge, MA, 1986.

[2] M. Ahmadi and M. Buehler, "Stable control of a simulated one-legged running robot with hip and leg compliance," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 1, pp. 96–104, February 1997.

[3] M. Ahmadi and M. Buehler, "The ARL Monopod II Running Robot: Control and Energetics," in *IEEE Intl. Conf. on Robotics and Automation*, 1999, pp. 1689–1694.

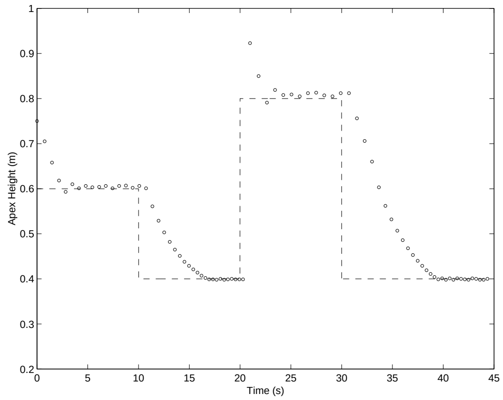
[4] G.Z. Zeglin and H.B. Brown, "Control of a bow leg hopping robot," in *IEEE Intl. Conf. on Robotics and Automation*, 1998.

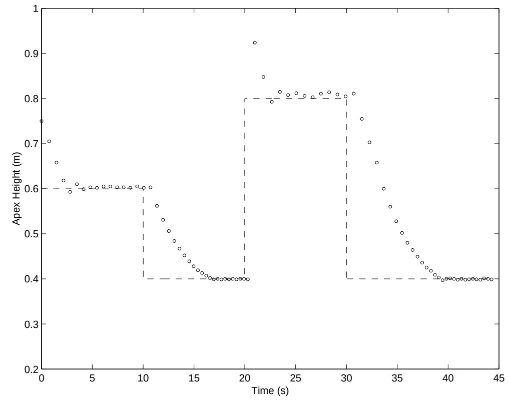
[5] Joseph Prosser and Moshe Kam, "Control of hopping height for a one-legged hopping machine," *Mobile Robots VII*, vol. 1831, pp. 604–612, November 1992.

[6] Kale Harbick and Gaurav S. Sukhatme, "Height control for a one-legged hopping robot using a one-dimensional model," Tech. Rep. IRIS-01-405, Institute for Robotics and Intelligent Systems, University of Southern California, 2001.

[7] Kale Harbick and Gaurav S. Sukhatme, "Height control for a one-legged hopping robot using a two-dimensional model," Tech. Rep. IRIS-01-406, Institute for Robotics and Intelligent Systems, University of Southern California, 2001.

[8] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, UK, 2nd edition, 1999.

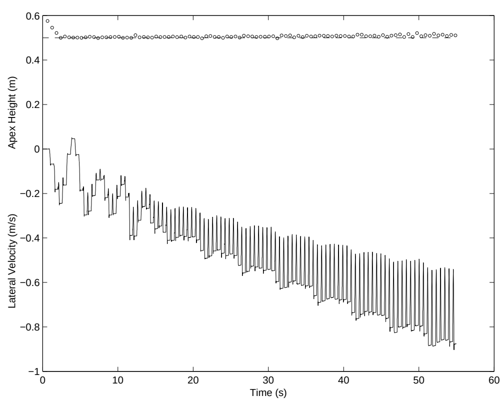


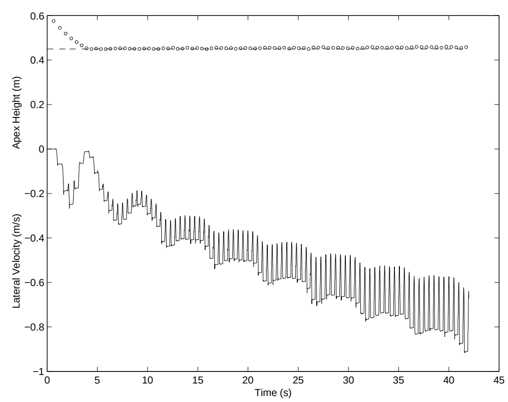


(a) 2D model

(b) 1D model

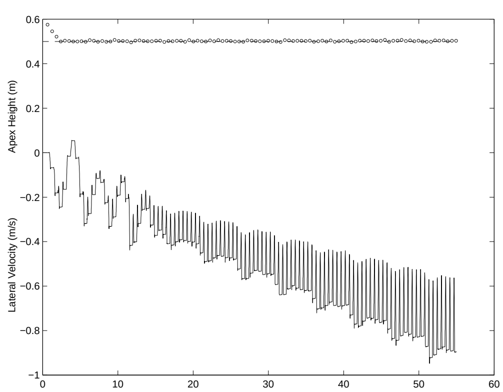
Fig. 3. Vertical hopping. Each circle denotes an apex height, and the dashed line represents the desired apex height profile.

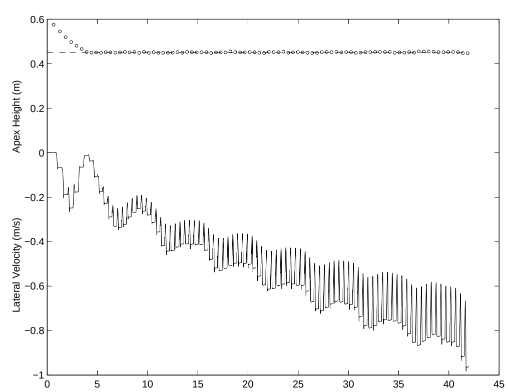




(a) 2D model, 0.5 m desired apex height

(b) 2D model, 0.45 m desired apex height





(c) 1D model, 0.5 m desired apex height

(d) 1D model, 0.45 m desired apex height

Fig. 4. Lateral hopping.