Control of a Bow Leg Hopping Robot

Garth Zeglin and Ben Brown
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213, USA

Abstract

The bow leg hopper is a new design for a locomoting system with a resilient, flexible leg. It features a passive stance phase and natural pitch stability. It is controlled with actuators that configure the leg angle and stored leg energy during flight. During stance, the actuators are mechanically decoupled from the leg and the stored energy is released. The trajectory is determined by the spring-mass physics and the state of the leg at impact. This design casts the controller as a function mapping three trajectory parameters to two control outputs once every hopping cycle. Our particular solution uses a combination of graph-search planning and feedback control. The planner searches to find sequences of foot placements and computes control outputs using numerical solution of a physical model. Feedback control is computed once per bounce. Experimental data from a planar prototype are included demonstrating navigation of simple artificial terrain.

1 Introduction

The *bow leg hopper* is a novel design for a locomoting robot that bounces passively on a flexible, efficient leg. It is controlled by adjusting the leg angle and stored leg energy during flight in preparation for impact. The body pitch rotation is passively stabilized by locating the center of mass slightly beneath the hip. During stance, the actuators are automatically decoupled and the bounce proceeds passively. The trajectory is determined by the impact state and the spring-mass physics of the robot. This design is energy efficient and moves the energy demand from the stance interval to the longer flight interval, reducing the required peak power. This design also imposes novel requirements on a locomotion controller, since the maximum control rate is one update per hopping cycle.

This paper defines the control problem implied by this mechanism and presents a particular solution with experimental results. The solution includes a simplified physical model, an on-line planner to find trajectory sequences across terrain that plans while in flight, a controller to execute the plans, and an underlying implementation that takes care of real-time chores. The discussion is limited to the case of a planar hopper constrained to a plane, although much of the argument also applies to the 3D case.

We have constructed a prototype planar hopper and implemented the controller presented. The prototype is mounted on a planar constraint boom attached to the floor. A spring connecting the boom to the ceiling reduces the effective gravity. The leg is controlled with two hobby servos: a leg angle motor to set leg position during flight, and a thrust motor to store energy by retracting the *bow string* that runs through the hip to the toe. Power is supplied by an onboard battery. Further details may be found in a companion paper [3]. The prototype has been tested by traversing simple artificial terrain. The results are preliminary, but performance and reliability continue to improve.

2 Related Work

The bow leg hopper has much in common with one leg hoppers built by Matsuoka, Raibert, Papantoniou, Buehler, and several others. It has a rigid body and a compliant leg with an angular positioning axis and an axis that controls thrust. The tasks are to maintain body stability and travel across rough ground. Raibert style three part control [9] can be applied to this hopper with only minor modification: the leg angle controls foot placement, thrust controls hopping height, and the third part, pitch control with hip torque, is not required.

There are several differences. The leg is a flexible bow shaped spring instead of a solenoid [7], telescoping spring [4] [6], or linkage [2] [8]. Pitch is stabilized passively by designing the center of mass below the hip. Most importantly, the hopper is "mechanically programmed" during flight to set initial conditions for impact. Any control must take place once per hopping cycle. High bandwidth control is eliminated along with negative work, and the controller is discrete in the sense that the machine moves between widely separated states between each control cycle.

Philosophically, our emphasis is on planning each step using a physical model instead of controlling a steady state oscillation [1] [5]. Rather than defining specific gaits, behaviors, and transitions between them, the planner finds physically feasible trajectories that satisfy task constraints. The goal is for the planner to be capable of considering any trajectory possible within the physical limits to solve tasks defined as constraints by a human. We expect this will enable automatic generation of a rich variety of performances. We intend to repeat some demonstrations performed by previous machines (running, stairs, etc.), but with a minimum of special programming. Hopefully, this approach should allow solution of more intricate problems in which many terrain elements are composed together, eventually leading to solutions for natural terrain.

In principle, this planning approach could be applied to other hopper designs by treating the closed-loop control of stance as a black box that can be discretely modeled. The model of the low level control would be a discrete function that predicts a takeoff trajectory from a landing trajectory given the parameters supplied to the low level controller during the stance phase. The planner would still search out sequences of trajectories and the output would be low level controller parameters.

3 Properties of the Controller

The bow leg mechanical design permits only one control cycle per bounce and this defines the properties of the controller. The controller function takes the following form:

$$(\phi_{n+1}, \Delta E_{n+1}) = f(x_n, y_n, \dot{x}_n) \tag{1}$$

In this function the variables $(\phi_{n+1}, \Delta E_{n+1})$ are the leg angle and stored leg energy at impact and $(x_n, y_n, \dot{x}_n)$ define the trajectory preceding the impact. This function summarizes the control and comprises the physical model used for feedforward, terrain data, the task being performed, and error feedback. The discrete form can be justified by examining the effect of each actuator and the definition of state.

The leg servomotor determines the angle of the leg prior to impact. During flight, the leg carries no load and can be positioned quickly. This motion only slightly affects body pitch since the leg mass is approximately 1% of the body mass. During stance, the leg positioning motor is physically decoupled from the leg. It is conceivable the leg servo could be repositioned during stance in order to exert horizontal ground forces as the bow string regains tension at liftoff, but we consider this unreliable and ignore this possibility. Thus the leg motion can be entirely described as ϕ_n, the leg angle in world coordinates at impact n.

The thrust motor determines the energy stored in leg tension prior to impact. The leg always begins flight at its lowest energy state. During flight, the motor performs positive work on the leg spring. It is conceivable for it to immediately reverse and dissipate some stored energy but the *net work* during flight is always non-negative. During stance, the thrust motor becomes physically decoupled from the leg as the now-slack string is released. The leg then extends to full length, and all stored energy is released. The thrust action can be entirely described as ΔE, a non-negative potential energy added to the kinetic energy.

The full physical state nominally has ten dimensions: three body DOF, two actuator DOF, and the corresponding velocities. We make several assumptions to define a trajectory using only three dimensions. First, we may neglect pitch and pitch velocity since the body is designed to passively stabilize pitch and rotates like a slow pendulum. This axis is decoupled from the other coordinates since body rotations only slightly affect the direction of leg forces and the leg position is independently defined in world coordinates. Second, on the time scale of the hopping cycle the actuators have insignificant dynamics and may be treated simply as outputs.

The hopper may thus be treated as a point particle with four state variables $(x, y, \dot{x}, \dot{y})$. However, we assume that all constraints are time-invariant and so only the geometry of the trajectory matters. Since the free flight physics is known, each trajectory can be described by only three parameters; we use the set $(x_n, y_n, \dot{x}_n)$, which are the position and velocity at the apex of the trajectory.

Note that the leg and thrust values are a function of time during flight $(\phi(t), \Delta E(t))$, but only the final values $(\phi_n, \Delta E_n)$ affect the impact. The abstract control problem is described with discrete functions but the implementation does require control over time. The abstract control values closely correspond to the mechanical freedoms: the stored energy is a monotonic function of the thrust servo angle, and the leg angle ϕ is the sum of the body attitude θ and the leg servo angle.

The low motor power does impose timing constraints. The minimum time required to store leg energy depends on the magnitude of ΔE and the maximum motor power. In practice, the entire flight time is required to store a large impulse, so energy storage for impact n must typically begin immediately after takeoff $n-1$; that energy will affect the trajectory following impact n. In contrast, the leg servo can typically position the leg shortly before impact since it is moving an unloaded low mass leg.

4 Physical Model

The controller uses a model of the hopper physics for planning paths and for feedforward control. The physics function is a discrete map from one trajectory to the next

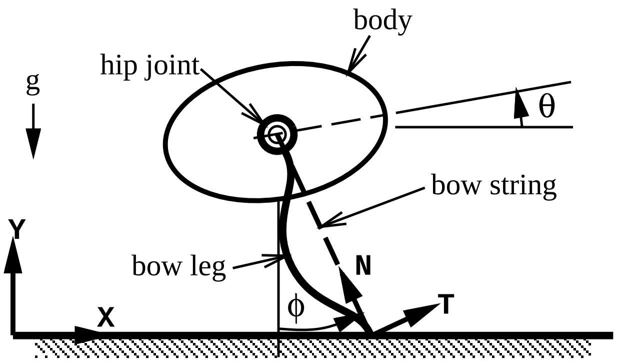

Figure 1: Cartoon of the planar bow leg hopper. The leg is a lightweight fiberglass spring with high restitution. The hip is a ball bearing pivot that exerts minimal body torque. The thrust actuator and leg angle actuator are located on the body but not illustrated. The bow string attaches the thrust actuator to the toe and can be retracted to compress the leg. The center of mass (not illustrated) is located just below the hip for natural pitch stability. The hopper position is defined by the body center of mass position and orientation (x, y, θ) and the leg angle ϕ, which is measured with respect to the world -Y axis. Gravity points in the -Y direction. The impact is analyzed in the leg frame $\left[\hat{T} \quad \hat{N}\right]$.

given the control parameters of the intervening impact. It combines the physics of the hopper and geometric information about the terrain.

Although the controller views the physical model as a discrete function, the physics is a continuous time system and could be modeled using differential equations. However, the hopper is designed to have dynamics similar to idealized models, so we have chosen to use a discrete closed form model based on idealized analysis, combined with ad hoc but physically motivated corrections. This approximation will continue to be refined but is producing encouraging experimental results.

The various parameters in the model are determined by a least squares fit to a set of recorded trajectories. Some parameter values and statistics are shown in Table 1. The errors listed are the residual; i.e., the distribution of the differences between the predicted and actual trajectory parameters on the same data set with which the model was fitted.

The analytic portion of the model is based on the assumption of a massless leg and instantaneous impact. The leg is attached with a pin joint at the hip and an effective pin joint where the foot makes point contact with the ground. With no leg inertia, the free body equilibrium dictates that the ground force applied to the toe lies along the axis of the leg and is balanced by an opposing hip force. The total force on the body is the sum of gravity and the leg spring force. The spring has restitution ϵ that defines the ratio of impulse released to impulse absorbed. The hopper bounces like a ball on a paddle perpendicular to the leg axis. With no thrust, the tangential velocity is unchanged and the normal velocity is mirrored with a loss:

$$\begin{aligned} v_{n1} &= -\epsilon v_{n0} \\ v_{t1} &= v_{t0} \end{aligned} \tag{2}$$

This may be modified to include the effect of thrust. The energy stored in the leg is a function of thrust motor angle and is independent of the impact state. Assuming perfect transfer from spring storage into kinetic energy, the impact may be modeled as follows:

$$\begin{aligned} v_{n1} &= \sqrt{\epsilon^2 v_{n0}^2 + (p_{t1}\theta_t + p_{t2}{\theta_t}^2)} \\ v_{t1} &= v_{t0} \end{aligned} \tag{3}$$

The two terms involving the thrust motor angle θ_t form a quadratic approximation of the energy stored in the leg. Note that the normal impact velocity v_{n0} is always negative and normal takeoff velocity v_{n1} is always positive.

In reality, the stance is not instantaneous and the leg sweeps a small arc while in contact. This angle is a function of stance time and the tangential velocity, but we simply lump the effect into a single parameter and approximate the actual leg sweep as follows:

$$\Delta\phi \simeq p_s \cdot -v_t \tag{4}$$

The leg angle at liftoff is the sum of the angle at impact and the sweep angle ($\phi_n + \Delta\phi$). Since the leg angle is not constant during stance the idealized reflection model is only an approximation. However, if the midpoint of the sweep ($\phi_n + 1/2\Delta\phi$) is used as the effective leg angle in computing the idealized model, the result is good enough to be a useful predictor of takeoff velocity.

The flight model assumes constant gravity and a constant lateral friction force. The effective gravity produced by the constraint boom and gravity compensation spring varies slightly with altitude, but the effect is negligible. The measurable but low horizontal deceleration is presumably due to bearing friction and tether drag.

The terrain is modeled as connected line segments. It is manually measured. Horizontal segments are considered valid footholds.

This is obviously simpleminded, and we only make the claim that it works well enough for us to use in practice. A more detailed model might model the leg potential and state trajectory during stance [10] [11].

Parameters Computed from Training Set `98-02-21`		
Parameter	Value	Definition
g	-2.43 m/sec^2	effective gravity
ϵ	0.82	restitution
ϵ^2	0.68	energy restitution
p_s	0.16	sweep angle coeffecient
p_{t1}	0.45	ΔE vs. thrust, linear term
p_{t2}	-0.07	ΔE vs. thrust, quadratic term

Error Statistics on Training Set `98-02-21`		
Statistic	Value	Definition
σ_x	8 mm	std. dev. of x error
σ_y	7 mm	std. dev. of y error
$\sigma_{\dot{x}}$	17 mm/sec	std. dev. of $\dot{x}$ error
N	442	samples in training set

Table 1: Model Parameters and Fitting Statistics

5 Planning

The task we have defined is to travel to a destination while obeying gait constraints. The basic constraints on this task are the location of footholds, contact friction, and obstacles. The gait constraints might include a desired velocity or hopping height, task constraints such as "land exactly on foothold x," or arbitrary constraints such as "alternate between short and long steps."

The role of the planner in the control system is to plan sequences of steps that attain the goal while satisfying the constraints. It is desirable that the planner operate in real time, be able to use terrain data obtained on-line, and produce plans tolerant of terrain and control uncertainty.

Our prototype planner is a simple attempt to meet these goals. It performs a best-first search of a graph of possible foot placements to explore sequences of trajectories. At every search step, a set of new foot placements (i.e., search nodes) is selected by sampling the continuum of available leg angles at a given impact. For each leg angle chosen, the trajectory that results is computed; the impact point at the end of the trajectory defines the new foot placement. The sampling procedure guarantees at least one choice of leg angle is selected for each reachable terrain segment. The branching factor of the best-first search is thus a function of the number of terrain segments reachable from a given liftoff and the sample spacing of the selection procedure.

The path is defined as a sequence of foot placements rather than a sequence of states or leg angles. This observes the terrain constraints, but a consequence is that adding a new foot placement to a path involves adjusting previous leg angles. This is performed by a numerical optimization that adjusts the leg angles to minimize the sum of absolute distances between the predicted foot contacts and the desired foot placements.

The best-first search is guided by the following heuristic function in which x and $\dot{x}$ are trajectory parameters, p is the number of bounces from the start, k_v and k_l are constant gains, and x_d is the goal position:

$$\begin{aligned} x_{\text{err}} &= x_d - x \\ \dot{x}_d &= \begin{cases} \dot{x}_{\max}, & \text{if } k_v x_{\text{err}} > \dot{x}_{\max} \\ -\dot{x}_{\max}, & \text{if } k_v x_{\text{err}} < \dot{x}_{\max} \\ k_v x_{\text{err}}, & \text{otherwise} \end{cases} \\ \dot{x}_{\text{err}} &= \dot{x}_d - \dot{x} \\ \text{score} &= -|x_{\text{err}}| - |\dot{x}_{\text{err}}| - k_l p \end{aligned} \tag{5}$$

Currently, the energy of the hopper is regulated using a feedback loop that varies thrust to maintain a constant total energy. The hopper is designed so that the dissipation is relatively independent of forward speed. The planner estimates the operation of this controller so that initial energy ramp-up or ramp-down will be correctly treated, but otherwise only needs to plan leg angles.

The toe is assumed to contact the ground with Coulomb friction with coefficient μ. To avoid slip the leg force must lie inside the friction cone within the angle $\phi_\mu = \arctan \mu$ of the surface normal. Since the leg force is always along the leg axis, leg angles within the friction cone satisfy the friction constraint.

6 Plan Execution

The plan is consistent with the model of the physics but is not naturally stable. The sources of uncertainty that lead the hopper off the plan include systematic error in the physical model, mechanical backlash in the leg servo, error in the state estimation, and friction and backlash in the constraint boom. After each impact the controller computes an adjustment to the plan for the next two impacts intended to return to the planned trajectory. If the error is too large, the controller abandons the plan and begins creating a new one from the measured state.

The leg angles $\phi_1..\phi_n$ at n successive impacts may be considered a vector that defines the reachable trajectories. In general, a trajectory is defined by three parameters and three successive impacts may span the trajectory space. However, hopping at constant energy reduces the trajectory space to two dimensions. Thus a deviation from the path can be corrected by adjusting two successive leg angles to reattain the planned trajectory. The correction combines linear feedback and feedforward computed using the physical model.

If the corrected foot placement falls outside a safe region defined around the planned foothold, the controller cannot guarantee the safety of that bounce and a new plan is generated. Planning occurs concurrently with execution; the planning system is an anytime planner and computes usable partial plans immediately. When starting from scratch, the best plan available before impact is used, but is then refined during the remainder of the hopping cycle. Once completed, the plan is used until accumulated error forces a replan.

7 Implementation

The controller views the hopper as a system controlled once each bounce by supplying values for ϕ and ΔE. The physical hardware does require real time attention to implement these commands. The underlying control software reads sensors and computes state estimates, controls the leg and thrust servo positions, and schedules the control computations. The prototype hopper uses hobby servos for the leg and thrust motors, so the lowest level of position control is implemented in hardware.

The leg actuator controls the leg angle relative to the body. Since ϕ is specified in world coordinates the actuator command is actually a function of body pitch. The thrust actuator angle is computed using the inverse of the thrust model presented in section 4.

8 Results

Figure 2 illustrates a successful experimental trial in which the hopper hops to a location, crossing five "obstacles." In this experiment the obstacles are simply designated regions on the floor with which contact must be avoided. The top plot shows the measured path of the body together with cartoons illustrating the body attitude and leg angle at the moments of impact. Below the recorded data are a series of plans generated during the traverse. The long plans are complete plans to the goal and the short plans are the adjustments computed to correct errors and return to the long plan. Ideally, the hopper would compute the complete plan once and execute it all the way to the goal. In this example the errors were too large on three steps and the complete plan was recomputed with a new starting state. The plans are illustrated using cartoons at impact, liftoff, and the apex to emphasize that the planner uses a discrete physical model that computes the transitions between these positions in closed form.

These results should be considered preliminary: the example shown was selected from a number of trials, most of which failed. We are still in the process of refining our mechanical design for reliability and our calibration procedure for repeatability.

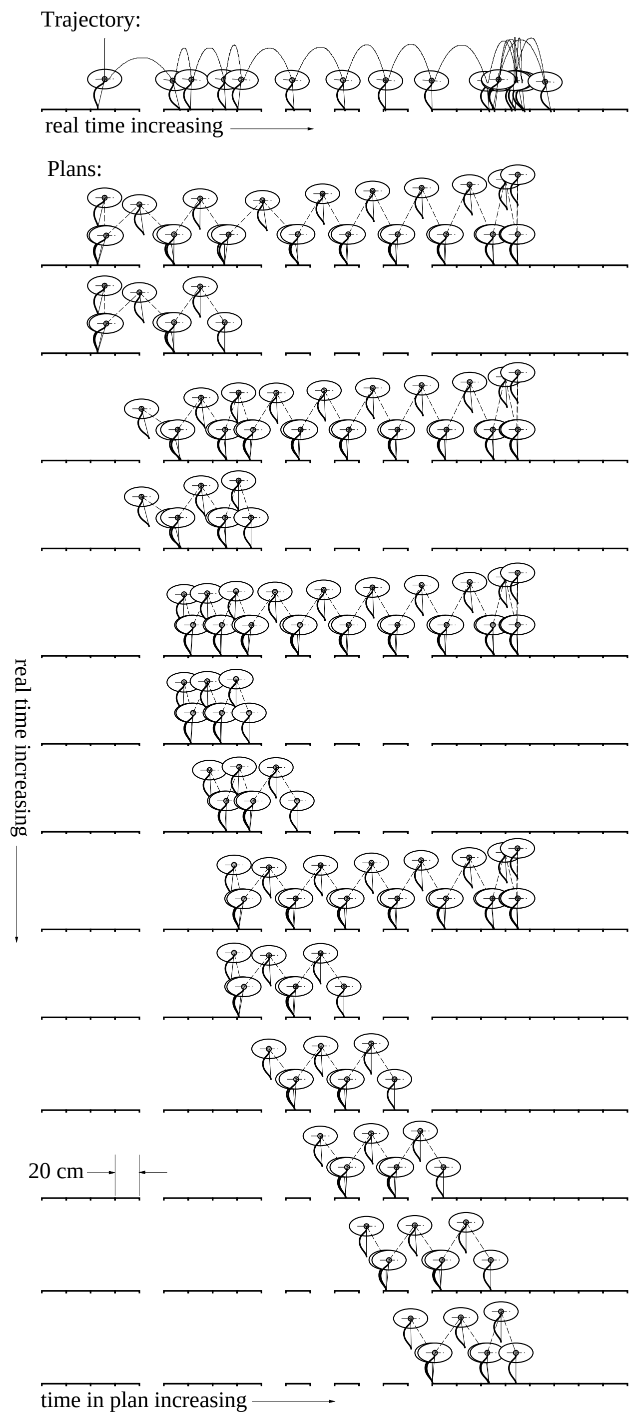

Figure 2: Experimental run and plans. The top plot illustrates the actual trajectory. Below are the succession of plans. Long ones are full plans, short ones are adjustments to correct errors. Real time increases moving down the figure, and planning time increases to the right.

9 Discussion

It is desirable for the control to complement the mechanism in order to take full advantage of every possible motion. The basic message of this work is to choose an unbiased solution method which can produce the best motion for a task from the space of possible motions. This is manageable in the case of the bow-leg hopper since the discrete control opportunities limit the space of possibilities to a continuous valued choice at discrete intervals.

However, the space of possible motions is vast and redundant and the search must be guided by sensible heuristics. It is important to note that at the heart of the planner is a linear controller that guides the search by choosing desired velocities with a linear function. By embedding this in a planning framework the linear control becomes a recommendation. This has several advantages: the terrain model is easily included, obstacles can be anticipated by looking forward in time, and arbitrary constraints can be observed to allow for a richer expression of tasks without specially programming new algorithms.

10 Future Work

Our prototype controller can develop along several avenues. The planning process could include uncertainty estimates to indicate potentially risky plans. The planner could consider energy constraints so plans would anticipate maneuvers that impose bounds on total energy. A maneuver such as crossing a high wall might require several bounces to build up high energy. Similarly, running under an overhang might require several dissipation steps to reduce the total energy. The terrain model will be refined to include obstacles such as walls or overhangs.

The experiments so far have been limited to crossing flat ground with defined "holes." We hope to experimentally demonstrate richer terrain crossing tasks with stairs, walls, and narrow footholds. Another possibility is to define gaits as constraints on the planner, e.g. "alternate short and long steps." An experiment to simulate the use of terrain sensing would be to modify or extend the terrain model during execution. The planner operates on-line so this would be a modest extension.

Designing a controller involves arbitrary choices. The planning methods could be better justified by examining the completeness and efficiency of the planning algorithm and the space of possible planning heuristics. More experimental verification is required to compare this approach to the space of possibilities. But we have high hopes that this combination of good mechanical design with versatile control will produce some exciting performances and potentially practical results.

Acknowledgments

We especially wish to thank Prof. Matthew Mason for his guidance and resources. We thank Dr. Alfred Rizzi and Prof. Illah Nourbakhsh for a number of illuminating discussions about combining control and planning. Prof. Daniel Koditschek, Prof. Martin Buehler, and William Schwind have provided valuable commentary. And we are grateful to Dr. Marc Raibert, who brought us to this work in the first place.

References

[1] Ahmadi, M., Buehler, M. 1997. "Stable Control of a Simulated One-Legged Running Robot with Hip and Leg Compliance." IEEE Transactions on Robotics and Automation, Vol 13, No. 1.

[2] Berkemeier, M.D., Desai, K.V. 1996. "Design of a robot leg with elastic energy storage, comparison to biology, and preliminary experimental results." Proceedings of IEEE International Conference on Robotics and Automation, 1996, Vol 1, pp. 213-18.

[3] Brown, H.B., Zeglin, G.Z. "The Bow Leg Hopping Robot." IEEE International Conference on Robotics and Automation, 1998.

[4] Gregorio, P., Ahmadi, M., Buehler, M. "Experiments with an Electrically Actuated Planar Hopping Robot." International Symposium on Experimental Robotics III, Kyoto, Japan, 1993.

[5] Hodgins, J.K., Raibert, M.N. "Adjusting step length for rough terrain locomotion." IEEE Transactions on Robotics and Automation, Vol. 7, No. 3, pp. 289-98, June 1991.

[6] Lebaudy, A., Prosser, J., Kam, M. 1993. "Control algorithms for a vertically-constrained one-legged hopping machine." Proceedings of 32nd IEEE Conference on Decision and Control, Vol 3, pp. 2688-93.

[7] Matsuoka, K., 1980. "A mechanical model of repetitive hopping movements." Biomechanisms 5:251-258.

[8] Papantoniou, K.V. "Electromechanical design for an electrically powered, actively balanced one leg planar robot." IEEE/RSJ International Workshop on Intelligent Robots and Systems 1991. Vol 3, pp. 1553-60.

[9] Raibert, M.H. 1986. *Legged Robots That Balance*. MIT Press, Cambridge, MA.

[10] Schwind, W.J., Koditschek, D.E. 1995. "Control of forward velocity for a simplifi ed planar hopping robot." Proceedings of 1995 IEEE International Conference on Robotics and Automation. Vol. 1, pp. 691-6.

[11] Schwind, W.J., Koditschek, D.E. 1997. "Characterization of Monoped Equilibrium Gaits." Proceedings of 1997 IEEE International Conference on Robotics and Automation.