

Stable Open Loop Walking in Quadruped Robots with Stick Legs

M. Buehler, A. Cocosco, K. Yamazaki, R. Battaglia

Center for Intelligent Machines, McGill University, Montreal, QC H3A 2A7, Canada

<http://www.cim.mcgill.ca/~arlweb>

Abstract

In our previous work we have shown that walking can be implemented on quadrupeds with stiff legs and only one actuated degree of freedom in the hip, based on the concept of controlled momentum transfer. In this paper we show via simulations and experiments on two different quadruped robots that dynamically stable and robust walking can be achieved via an open loop controller which is active only during the back leg support phase.

1 Introduction

Most existing four- or eight-legged walking robots are designed for statically stable operation – stability is assured by keeping the machine’s center of mass above the polygon formed by the supporting feet. While this is the safest mode of locomotion it comes at the cost of mobility and speed. Furthermore each leg needs at least three degrees of freedom to provide body support during forward motion.

Past research in dynamically stable running robots [11] has shown that a dynamic mode of operation permits increased mobility, speed, and much simplified mechanical designs by reducing the number of legs and the actuated degrees of freedom required. We believe that reducing mechanical complexity, and as a consequence, cost, maintenance, and risk of breakdowns is a critical element towards bringing legged robots into practical applications. For this reason we have started studying and building quadrupeds with only one actuator per leg. In [2] we have shown that such a design is still capable of a wide range of useful behaviors, including walking, turning, step/stair climbing and running.

In this paper, we focus on walking control and show that a very simple open loop leg sweeping algorithm can produce stable and robust robot walking. Why is this important? First it is not obvious and of conceptual interest that an inherently unstable dynamic

walking motion can be stabilized via an open loop algorithm. This might lead to interesting parallels to the biological control of locomotion, where spinal CPGs have been shown to provide “open loop” muscle stimulation without input from the brain [9]. In more practical terms, the presented controller needs no other sensors than foot switches to sense stance, and thus could lead to inexpensive walking robots for toys or entertainment applications. Even in more sensory-rich robots, the availability of a basic open loop locomotion algorithm might prove critical when sensors fail, or when adverse conditions hamper visual sensing.

There exists some published research on dynamically stable quadruped walking, which is based, however, on articulated legs. Miura *et al.* [8] developed dynamic walking control for their Collie-1 robot. Fusho *et al.* [5] built a quadruped with articulated legs and developed a controller for a running gait. Hirose *et al.* [6] designed a quadruped walking vehicle for dynamic walking and stair climbing. Dunn and Howe [4] presented a smooth walking controller for their biped robot, but the same approach could be used for a quadruped as well. Motivated by McGeer’s work [7], Smith and Berkemeier [12] showed that quadrupedal walking is possible based on passive dynamics.

2 Robot Model

Quadrupeds with articulated legs are capable of a variety of walking gaits, where the swing leg is retracted to prevent toe stubbing. When the legs cannot be retracted, like in our simple design, a bounding (rocking) type walking gait produces body pitching, which provides ground clearance for the swing legs. Thus at any given instant, the robot is statically unstable, pivoting either on the front leg or the back legs. For our model we assume that the exchange between front and back legs occurs via instantaneous front or back leg impacts. Since the front and back leg pairs move in pairs, a planar model as shown in Fig. 1 suffices.

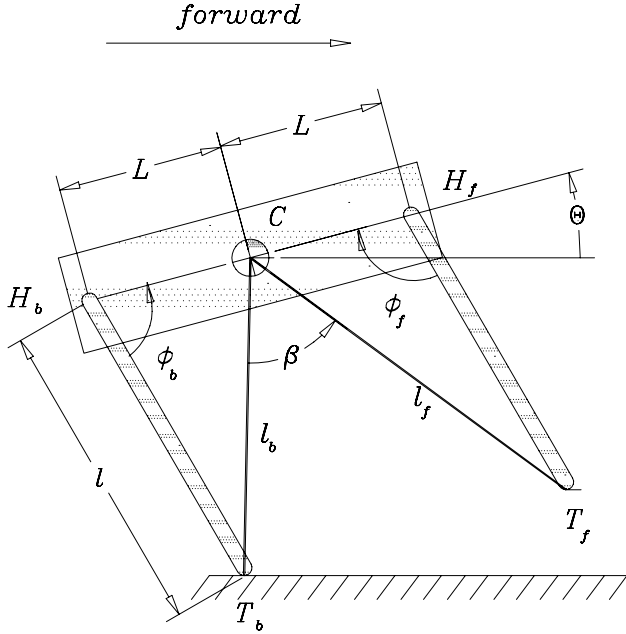


Figure 1: SCOUT model

The legs are connected to the body via the back and front hip joints, H_b and H_f . The two actuators of this planar model are placed at the hips and control the hip angles ϕ_b and ϕ_f . When a toe, T_b or T_f , is in contact with the ground, it is modeled as a pin joint (no friction or slipping). We assume the leg mass to be negligible, therefore only the supporting leg and its hip actuator influence the dynamics of the system. Thus, SCOUT can be modeled as an underactuated, inverted double pendulum with one unactuated joint (stance toe) and one actuated joint (stance hip).

The nomenclature is as follows. Subscripts b, f refer to front and back legs, superscripts B, F denote variables at time of front or back leg impact, and superscripts $+, -$ denote (velocity) variables just after or before impact. Thus $\dot{\theta}^{B+}$ is the body angular velocity just after back leg impact, and ϕ_b^B is the back leg angle at back leg impact.

The equations of motion for each single support phase are derived using the Lagrange Method, and describe the nonlinear dynamics of the body angle, θ , and the hip angle, ϕ , as a function of the hip torque, τ . These equations are used for simulations with Matlab, and are provided in [3].

During front or back leg impact, an instantaneous double support phase occurs, during which we assume that we can control the hip angles and angular velocities. Based on conservation of angular momentum around the impact toe, the change of body velocities

at back leg impact can be calculated as

$$\begin{aligned}
 & [l[-l \cos(\phi_b^B + \phi_f^B) + L \cos \phi_f^B] + r^2 - L^2 + lL \cos \phi_b^B] \dot{\theta}^{B-} \\
 & + l[-l \cos(\phi_b^B + \phi_f^B) + L \cos \phi_f^B] \dot{\phi}_f^{B-} \\
 & = [l[l - L \cos \phi_b^B] + r^2 + L^2 - lL \cos \phi_b^B] \dot{\theta}^{B+} \\
 & - l[l - L \cos \phi_b^B] \dot{\phi}_b^{B+}
 \end{aligned}$$

or, more concisely, as

$$\dot{\theta}^{B+} = c_0 \dot{\theta}^{B-} + c_1 \dot{\phi}_f^{B-} + c_2 \dot{\phi}_b^{B+}$$

where the c_i are configuration dependent factors. A similar relation can be derived for front leg impacts.

3 Control

For purposes of controller design we assume that we can control the hip angles and angular velocities; thus, our inputs are $\phi_f(t)$ and $\phi_b(t)$. However, the simulations (and, of course, the experiments) employ a PD controller to determine the hip torque necessary to track the desired hip angle trajectory.

The robot states to be controlled are the body angle and angular velocity

$$\Theta = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}.$$

One complete step consists of a front and a back leg support phase, and a front and a back leg impact. For purposes of analysis it is convenient to examine the variable of interest, Θ , at only one instant in this step, and we chose the instant immediately following the back leg impact,

$$\Theta_n \triangleq \Theta^{B+} = \begin{bmatrix} \theta^B \\ \dot{\theta}^{B+} \end{bmatrix}. \quad (1)$$

Now we can define a discrete step-to-step return map, \mathcal{S} , which maps the body states just after impact from step n to step $n + 1$, as a function of the inputs $\phi_b(t), \phi_f(t)$,

$$\Theta_{n+1} = \mathcal{S}(\Theta_n, \phi_b(t), \phi_f(t)). \quad (2)$$

For the ramp controller introduced in this paper, the controlled inputs already determine θ^B , so (2) becomes a scalar return map for $\dot{\theta}^{B+}$. The control objective can be stated as finding continuous hip angle trajectories, $\phi_b(t), \phi_f(t)$, which make the desired body states, Θ^* , an asymptotically stable fixed point of the return map \mathcal{S} .

The ramp controller was originally motivated by the desire to find a controller for dynamically stable walking which requires minimal sensing, computation, and control. Such a controller would keep the front legs fixed at $\phi_f(t) = \phi_f$, start with a back leg touchdown angle of ϕ_b^B and sweep the back legs during back leg stance with a constant angular velocity $\dot{\phi}_b(t)$ until the front legs impact again. Thus, strictly speaking, our controller is not completely open loop, since we make use of the leg's ground contact sensors to switch the back leg ramp controller tracking on and off. In addition, the ramp trajectory tracking requires feedback of the back hip angle and angular velocity. However, it is open loop in the sense that there is no measurement or feedback of the body angular position (wrt to horizontal) or velocity. These are the states to be stabilized, and also the states which are most difficult to measure in practice.

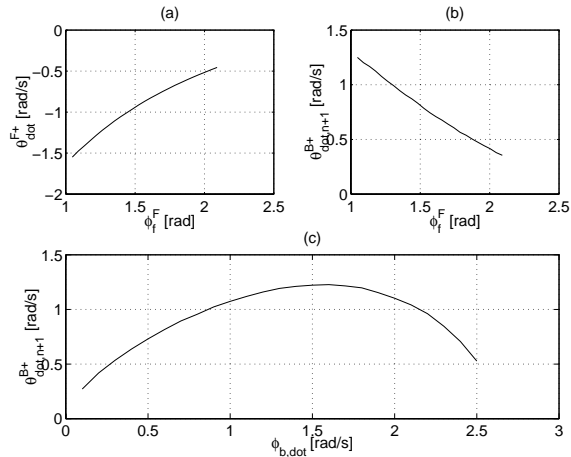


Figure 2: Effect of ϕ_f^F (a) on $\dot{\theta}^{F+}$ and on (b) $\dot{\theta}_{n+1}^{B+}$; (c) effect of $\dot{\phi}_b$ on $\dot{\theta}_{n+1}^{B+}$

A set of MATLAB simulations was used to determine the ramp control design parameters $\phi_b^B, \dot{\phi}_b, \phi_f$ which would make our desired body angular velocity after back leg impact, $\dot{\theta}_b^{B+} = 1 \text{ rad/s}$ a fixed point of the step-to-step return map, \mathcal{S} . To simplify the search, we fixed the first parameter, the back leg angle at back leg impact, $\phi_b^B = 1.05 \text{ rad}$. In order to obtain a large magnitude for the body angular velocity after front leg impact, the front leg angle should be small (Fig. 2(a)), but only a value of $\phi_f = 1.31 \text{ rad}$ will result in the desired $\dot{\theta}_b^{B+} = 1 \text{ rad/s}$ (Fig. 2(b)). Finally, Fig. 2(c) shows that there are two back leg angular velocity settings which achieve the desired $\dot{\theta}_b^{B+} = 1 \text{ rad/s}$ and we chose the smaller value, $\dot{\phi}_b = 0.85 \text{ rad/s}$. Alternatively, gradient type searches can be employed

to search for fixed points of \mathcal{S} in three dimensional controller parameter space.

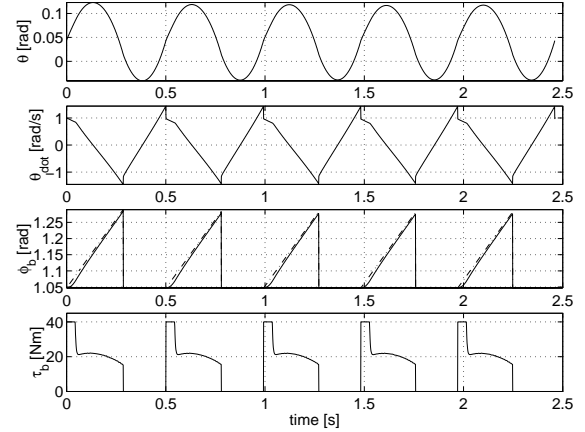


Figure 3: Stable motion produced by the open loop ramp controller. Motor torque limit set to 40 Nm per actuator (80 Nm total for back leg pair).

Simulations (Fig. 3) of the open loop ramp controller suggest that the chosen set point does not only correspond to a fixed point of the step-to-step map, but is at least locally stable. To further investigate the possible stability of this open loop controller around the set point, we added a severe perturbation after the first step, and, to our great surprise, the system converged rapidly back to the desired set point, $\dot{\theta}^{B+} = 1 \text{ rad/s}$ (Fig. 4).

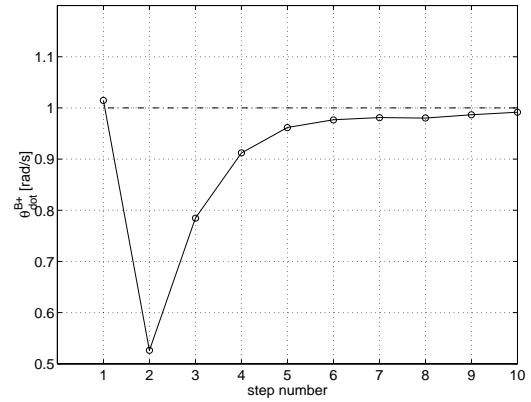


Figure 4: System response with open loop ramp controller after severe disturbance

For a more complete insight into the range of initial body angular velocities which will converge to the desired set point (the domain of attraction of the controller), we plot the numerical evaluation of the step-to-step return map for the open loop ramp controller

in Fig. 5. This plot confirms the unexpected fact that the open loop controller has a domain of attraction which is global for all practical purposes, from almost zero initial body angular velocity, to a maximum body angular velocity of $\dot{\theta}^{B+} = 2.3 \text{ rad/s}$, above which the robot would fall over backwards!

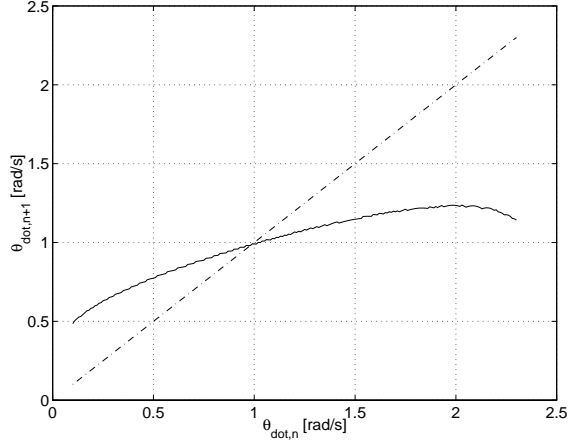


Figure 5: Numerical evaluation of the step-to-step return map for the open loop ramp controller

In order to increase the rate of convergence further, a feedback control mechanism can be added which adjusts the back leg angular velocity as a function of the initial body angular velocity $\dot{\theta}^{B+}$. For a range of after-impact body angular velocities $\dot{\theta}^{B+}$, and a range of back leg hip angular velocities $\dot{\phi}_b$, a look-up table has been generated. The input in this table is the actual (measured) $\dot{\theta}_n^{B+}$ and the output is the required $\dot{\phi}_b$ which, during the back leg support of the $(n+1)$ step, will result in the desired set point $\dot{\theta}_{n+1}^{B+} = 1 \text{ rad/s}$. The look-up table has 180 entries (from $\dot{\theta}_n^{B+} = 0 \text{ rad/s}$ to $\dot{\theta}_n^{B+} = 1.79 \text{ rad/s}$, in steps of 0.01 rad/s). For entries different from the tabulated values, a linear interpolation function is used to generate the outputs. This look-up table is used to simulate walking for several steps, with 50% error in the $\dot{\theta}^{B+}$ (step 2) and 30% error for $\dot{\theta}^{F+}$ (step 7). As it can be seen in Fig. 6 the recovery in both cases is accomplished in just one step. This is confirmed by the closed loop step-to-step return map plot Fig. 7.

4 Experiments

The open loop ramp controller was implemented on two quadruped robots, SCOUT I (Fig. 8) and SCOUT II (Fig. 9). SCOUT I [2] weighs 1.2 kg and has leg

lengths and body length (hip to hip) of 0.2 m . Its body width (left toe to right toe) is 0.145 m at the front and 0.19 m at the back. SCOUT II [1] weighs 27 kg and measures 0.275 m in height, 0.552 m in length and 0.48 m in width. SCOUT I has (position controlled) hobby RC servo motors at the joints, while SCOUT II features geared brush DC servo motors. Both robots are controlled by a PC, running the QNX real time operating system [10].

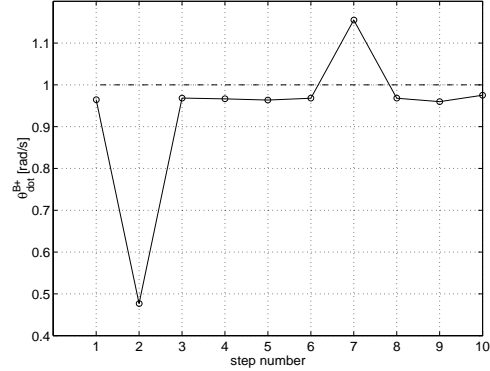


Figure 6: Closed loop control achieves one step recovery from disturbances

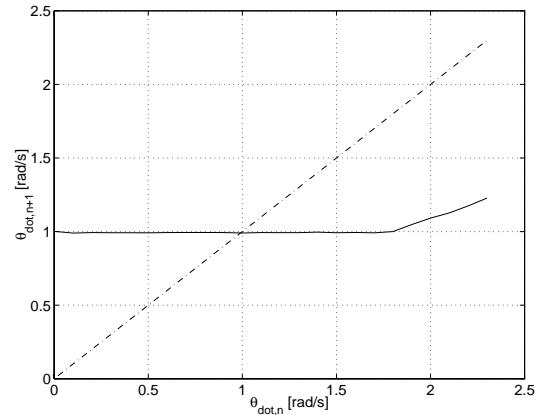


Figure 7: Step-to-step return map for closed loop (LUT) ramp controller

The controller parameter derivation for the open loop ramp controller as described above was done using SCOUT I parameters, based on a higher angular body velocity setpoint. The same strong stability properties were obtained: Fig. 10 shows transient and steady state experimental walking data - the robot converges to the stable fixed point operation within two steps, even after large initial errors. More details of the SCOUT I implementations and model verifica-

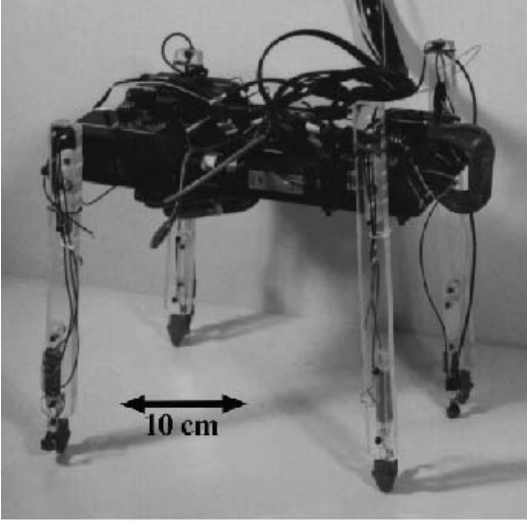


Figure 8: SCOUT I

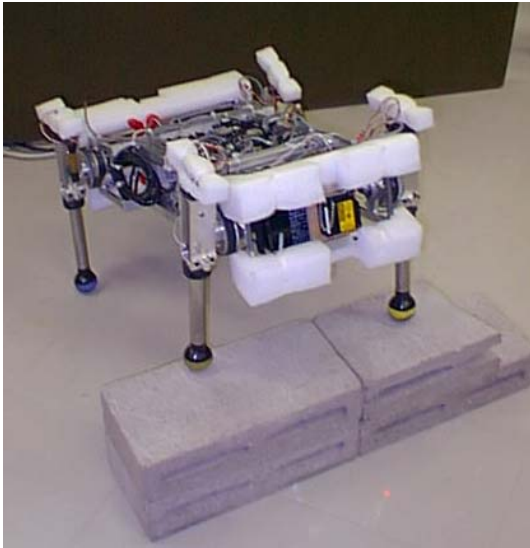


Figure 9: SCOUT II

tion can be found in [13].

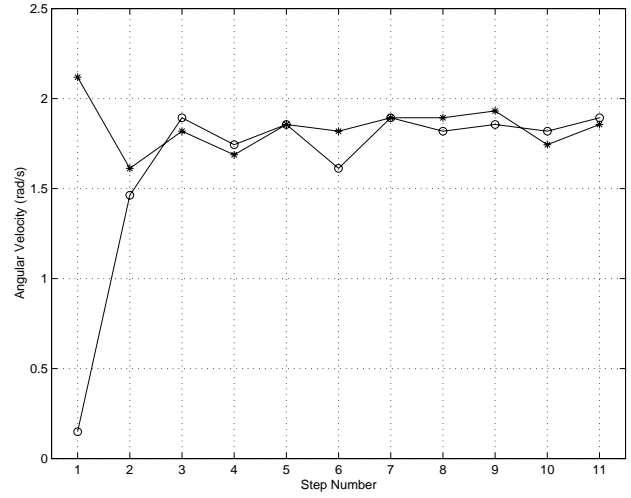


Figure 10: Stable walking in SCOUT I: Transients from different initial conditions; Experimental Data

The ramp controller implementation on SCOUT II uses the control parameter settings derived in the previous section to achieve a stable fixed point for the body angular velocity just after back leg impact of $\dot{\theta}_b^{B+} = 1 \text{ rad/s}$. The configuration at the moments of exchange of support are $\phi_f = 1.31 \text{ rad}$ for the (fixed) front leg angle, $\phi_b^B = 1.05 \text{ rad}$ for the (initial) back leg angle at back leg impact, and $\dot{\phi}_b = 0.85 \text{ rad/s}$ for the back leg stance sweep velocity.

Experimental data of stable SCOUT II walking is shown in Fig. 11. The top two graphs show the body angle oscillations and the angular velocity for six walking steps. The third plot shows the back leg angle and the tracking error for the constant desired velocity during stance. The bottom plot shows one of the two back leg actuator torques, limited to 40 Nm . These experiments confirm the stable and robust walking behavior expected from the earlier simulations. The plot of angular body velocities shows a value close to $\dot{\theta}_b^{B+} = 1 \text{ rad/s}$ at the beginning of the back leg support phase (marked with 'o'). However, a comparison with the simulation Fig. 3 shows some marked differences as well.

The main differences are due to our naive assumption of instantaneous exchange of support. Even with SCOUT II's stiff legs, there is sufficient compliance in the system (toes, transmission belt, gear, motor controller limitations) that we have in fact a double support phase with accounts for approx. 20% of the step time! Since we start sweeping the back legs already during this period, the subsequent back leg sup-

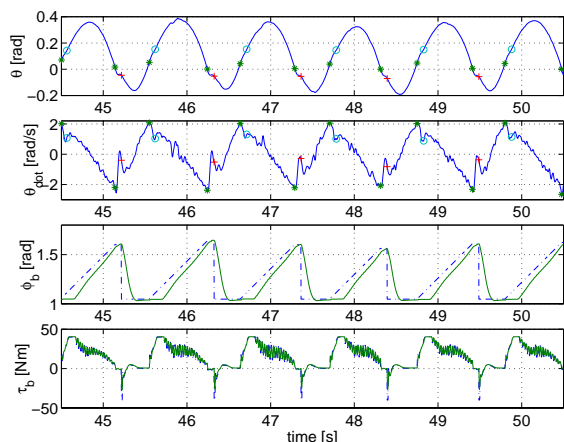


Figure 11: Experimental results for the ramp controller; ‘*’ beginning of the double stance phase; ‘o’ beginning of the back leg support and ‘+’ beginning of front leg support; dashed line: desired; solid line: actual

port phase is longer. The existence of the double support phase, and the resulting increase in back leg support phase explain the large difference in step times between simulation (0.5s) and experiment (1s). The longer step time goes hand in hand with a larger body oscillation amplitude and peak body angular velocity.

Nevertheless, the qualitative results of the simulations and the experiments are very similar. The fact that we obtain stable and robust walking performance despite large modeling errors can be interpreted as another indication of the strong robustness properties of this class of open loop controllers for dynamically stable walking.

5 Conclusion

A new open loop controller has been presented which stabilizes dynamic bounding walking in a class of simple quadruped robots. The strong global stability properties were illustrated via the step-to-step return map using numerical tools. Experimental implementation of this controller on two quadruped robots showed that, despite modeling errors, stable walking was achieved.

Acknowledgments

This project was supported in part by IRIS, a Federal Network of Centers of Excellence, the National

Science and Engineering Research Council of Canada (NSERC), Terra Aerospace Corp., Aromat Inc, and Abacom Technologies. We also acknowledge the generous and talented technical support by D. McMordie and N. El-Fata.

References

- [1] R. Battaglia. Design and control of a four-legged robot, SCOUT II. M. Eng. Thesis, McGill University, Mar 1999.
- [2] M. Buehler, R. Battaglia, A. Cocosco, G. Hawker, J. Sarkis, and K. Yamazaki. Scout: A simple quadruped that walks, climbs and runs. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1707–1712, Leuven, Belgium, May 1998.
- [3] A. E. Cocosco. Control of walking in a quadruped robot with stiff legs. M. Eng. Thesis, McGill University, July 1998.
- [4] E. R. Dunn and R. D. Howe. Foot placement and velocity control in smooth bipedal walking. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 578–583, Minneapolis, MN, Apr 1996.
- [5] J. Furusho, A. Sano, M. Sakaguchi, and E. Koizumi. Realization of bounce gait in a quadruped robot with articular-joint-type legs. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 697–702, May 1995.
- [6] S. Hirose, K. Yoneda, K. Arai, and T. Ibe. Design of a quadruped walking vehicle for dynamic walking and stair climbing. *Advanced Robotics*, 9(2):107–124, 1995.
- [7] T. McGeer. Passive dynamic walking. *Int. J. Robotics Research*, 9(2):62–82, 1990.
- [8] H. Miura, I. Shimoyama, M. Mitsuishi, and H. Kimura. Dynamical walk of quadruped robot (Collie-1). In H. Hanafusa and H. Inoue, editors, *Int. Symp. Robotics Research*, pages 317–324. MIT Press, Cambridge, 1985.
- [9] K. Pearson. The control of walking. *Scientific American*, 235:72–86, 1976.
- [10] QNX Software Systems LTD. *QNX Operating System v4.2*. Ontario, Canada, 1996.
- [11] M. H. Raibert. *Legged Robots That Balance*. MIT Press, Cambridge, MA, 1986.
- [12] A. C. Smith and M. D. Berkemeier. Passive dynamic quadrupedal walking. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 34–39, Nantes, France, April 1997.
- [13] K. S. Yamazaki. The design and control of SCOUT I, a simple quadruped robot. M. Eng. Thesis, McGill University, Dec 1998.