In Proc. 2001 IEEE RSJ Int'l Conf. on Intelligent Robots and Systems (IROS 2001)

Footstep Planning Among Obstacles for Biped Robots

James J. Kuffner, Jr. Koichi Nishiwaki Satoshi Kagami
Masayuki Inaba Hirochika Inoue

Department of Mechano-Informatics
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{kuffner,nishi,kagami,inaba,inoue}@jsk.t.u-tokyo.ac.jp

Abstract

We present an algorithm for planning safe navigation strategies for biped robots moving in obstacle-cluttered environments. From a discrete set of plausible statically-stable, single-step motions, a forward dynamic programming approach is used to compute a sequence of feasible footstep locations. In contrast to existing navigation strategies for mobile robots, our method is a global method that takes into account the unique ability of legged robots such as bipedal humanoids to traverse obstacles by stepping over them. Heuristics designed to minimize the number and complexity of the step motions are used to encode cost functions used for searching a footstep transition graph. We show preliminary results of an experimental implementation of the algorithm using a model of the H6 humanoid navigating on an office floor littered with obstacles.

1 Introduction

Humanoid robotics technology has progressed rapidly during the past several years, and the commercial availability of humanoid robot hardware is likely to happen very soon. This will lead to a rising demand for software and algorithms useful to improving the usability and autonomy of humanoids. One important area of need will be software for safe, autonomous navigation in human environments such as homes and offices.

Research on global path planning and navigation strategies for mobile robots has a large and extensive history in the robotics literature. Since the problem can usually be modeled as searching for a collision-free path in a 2D environment, very efficient and complete algorithms can be employed (for an overview, see [6]). However, most of these techniques apply to wheeled robots which must always circumvent obstacles. In contrast, legged robots (including biped humanoids) have the unique ability to traverse obstacles by stepping over or upon them.

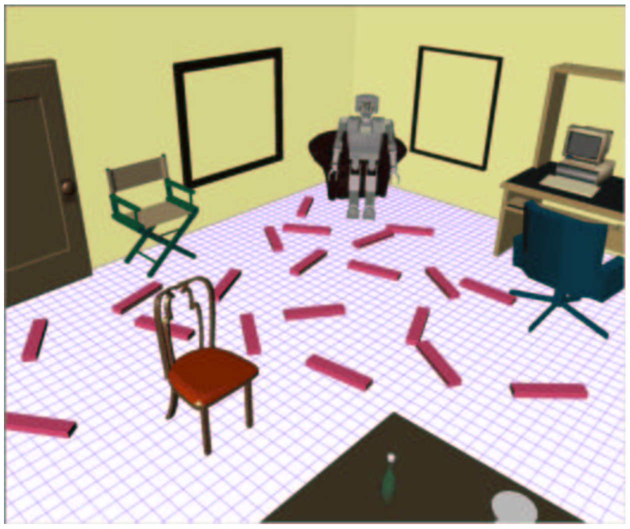
Figure 1: Humanoid robot navigating in a cluttered office.

2 Related Research

Most existing research on humanoid robots has focused on pre-generating stable walking trajectories (e.g. [2, 17, 9]), or on dynamic balance and control (e.g. [15, 12]). Recently, techniques have been developed to generate stable walking trajectories online by mixing pre-generated stepping patterns [11], though these results do not account for obstacles. For quadruped robots, adaptive gait generation and control on irreg-

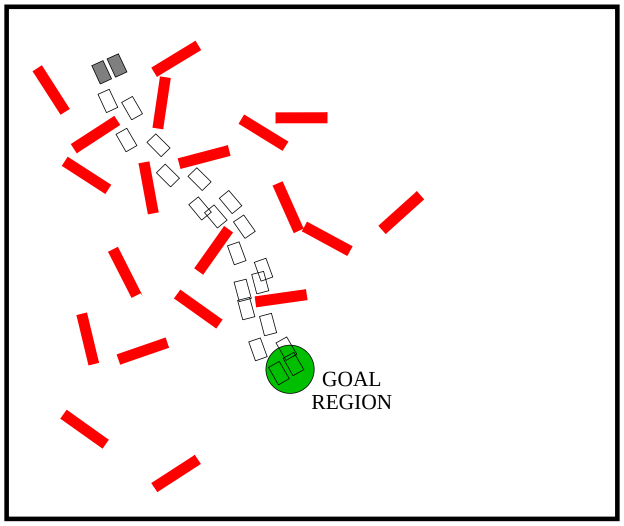

Figure 2: Planned footstep locations (Top view).

ular terrain and among obstacles has been previously studied [3]. This method has not yet been applied to biped robots. Sensor-based obstacle-avoidance techniques have been developed for bipeds navigating in unknown environments [16, 7]. However, such reactive methods can become trapped in local loops or dead-ends, since they do not consider global information. Other related techniques in computer animation that use footprint placement for motion specification have been developed for bipeds[1, 14], and quadrupeds[5, 13].

Our approach is to build a search tree from a discrete set of feasible footstep locations corresponding to statically-stable stepping motions. Using standard dynamic programming techniques, we compute a sequence of footstep placements to reach a goal region in an obstacle-cluttered environment. Encoded heuristics minimize the number and complexity of the steps taken, as well as guide the search. This has the advantage of planning a *global navigation strategy* for bipeds that includes the ability to step over obstacles. Preliminary results have shown that such a strategy can be computed efficiently (from a fraction of a second to a few minutes, depending upon the environment size and complexity) on standard PC hardware.

The rest of the paper is organized as follows: Section 3 gives an overview of the biped stepping model, Section 4 describes the algorithm, Section 5 shows some preliminary results on an experimental implementation, and Section 6 concludes with a summary discussion.

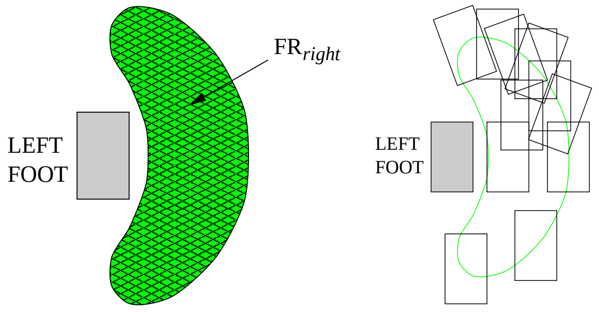

Figure 3: Statically-stable foot placements for the right foot. (*left*: continuous region; *right*: discrete subset)

3 Biped Navigation Model

Although the search technique adopted is very general, we currently make the following simplifying assumptions:

1. The environment floor is flat and cluttered with non-moving obstacles of known position and height.
2. A discrete set of feasible, statically-stable footstep placement positions and associated stepping motions are pre-computed.
3. Only the floor surface is currently allowed for foot placement (not obstacle surfaces).

Statically-stable Footstep Locations: The biped model comes with a pre-determined set of feasible footstep locations for each foot. For example, Figure 3 shows the continuous, feasible footstep range FR_{right} for the right foot while balanced on the left foot, and an example discrete set of foot placements. The placements for the left foot simply mirror the right foot placements, which is possible for symmetric bipeds.

In selecting which footstep placements to include in the discrete set used during the search, we chose a distribution of placements along the edge of the reachable region at different relative foot angles, as well as a few interior placements to allow the robot to maneuver in tight areas. This choice is currently only heuristic, and attempts to strike a balance between maximizing the navigation options, while minimizing the total number of discrete placements. In our implementation, we selected a total of 15 placements for each foot (see Section 5).

The number of placements is particularly important, as it determines the branching factor of the search

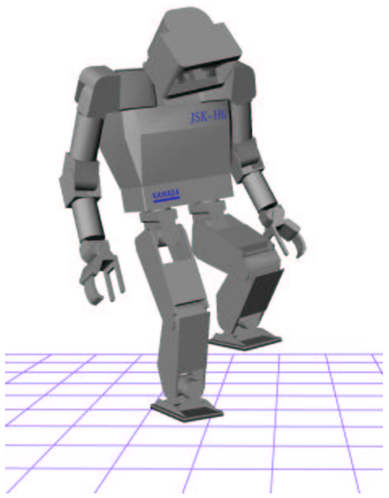

Figure 4: The intermediate posture Q_{right} used for transitioning between left leg footstep placements.

tree. Some relative rotation of the foot sole with respect to the support leg is necessary in order for the robot to alter its global orientation. A few backward foot placements are used to provide additional maneuverability for navigating environments with areas of limited free space. We note that alternatively, one could define multi-step "turn-in-place" motions to explicitly allow for larger changes in orientation at each discrete search step. This has not yet been implemented in our system.

Footstep Transition Trajectories: In addition to the set of feasible footstep locations for each foot, the planner also requires a pre-calculated set of statically-stable motion trajectories for transitioning between them. In general, this implies a quadratic number of transition trajectories between all possible footstep placement combinations.

In order to reduce the number of transition trajectories, we introduce two statically-stable, intermediate postures Q_{right} and Q_{left} that serve as a via points for all footstep transitions. Q_{right} and Q_{left} correspond to default postures in which the robot is balanced entirely on either the right or left foot respectively, with the other foot suspended high above the walking surface. Figure 4 shows Q_{right} for the H6 humanoid robot model (Q_{left} is defined by mirroring Q_{right}).

Using intermediate postures reduces the number of transition trajectories to roughly equal the number of footstep placements considered. An arbitrary transition between two placements of the right foot, Q_1 and

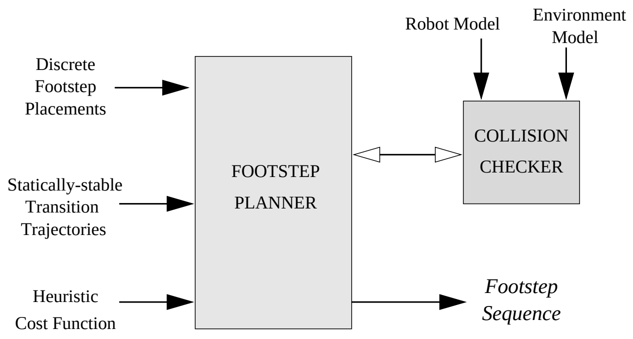

Figure 5: Block diagram of the planning algorithm.

Q_2, is formed by composing the transition from the starting posture Q_1 to Q_{left}, and then from Q_{left} to Q_2. Transitions for the left foot are composed in the same way, except Q_{right} is used as the intermediate posture.

4 Footstep Planning Algorithm

The planner accepts as input a discrete set of robot footprint locations, statically-stable transition trajectories, and a heuristic cost function. The 3D robot model and environment model is used for collision checking. If the planner successfully finds a solution, it outputs a sequence of encoded footstep placements and transitions. An overview of the planner is shown in Figure 5. Each of the main components are described in the following sections.

Dynamic Programming: Given a set of discrete placements, we adopt a *forward dynamic programming* approach to planning navigation strategies. For an overview on dynamic programming, the reader is referred to any textbook on artificial intelligence or game search trees (e.g. [4] or [10]). Since an exhaustive search is too expensive, we employ a greedy heuristic in order to prune the search tree.

Starting from an initial biped configuration Q_{init}, a search tree of possible footstep placements is constructed as in Figure 6. The planner maintains a *priority queue* of search nodes containing a footprint placement configuration and a heuristic *cost value* (see Section 4). The nodes are inserted into the queue based on their cost.

Initially, the queue is initialized with a single node N_{init}, which contains the starting configuration Q_{init}. At each iteration, the planner removes the lowest-cost (higher priority) node N_{min} from the priority queue.

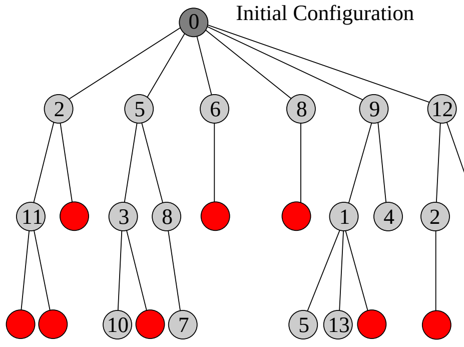

Figure 6: Search tree rooted at the initial configuration. Successor states that resulted in collisions (dark leaf nodes) are pruned from the search.

Successor nodes of N_{min} are generated that correspond to all potential placement locations for the next footstep. Collision checking is used to eliminate successor nodes which result in obstacle collisions (see Section 4).

When a successor node is generated that falls within the predefined goal region, the search terminates and a solution path back to the root node of the tree is returned. The solution path defines a discrete sequence of footprint placements which when executed will take the robot from Q_{init} to the goal region.

The search can fail in one of two ways:

- No more valid successor nodes can be generated. In this case, no collision-free footstep sequence exists using the given discrete set of relative footprint placements.
- The size of the search tree exceeded a pre-determined limit on the maximum number of nodes.

Estimated Cost Heuristic: We encode a simple greedy heuristic by defining a cost function $\mathcal{L}(Q)$ which is used for maintaining the priority queue. $\mathcal{L}(Q)$ combines the cost of the path taken so far along with an estimate of the cost to reach the goal region.

$$\mathcal{L}(Q) = w_d D(N_Q) + w_\rho \rho(N_Q) + w_g \mathcal{X}(Q, Q_g)$$

The first two terms define the cost of the path to configuration Q from Q_{init}. $D(N_Q)$ is the depth of the node N_Q in the tree. $\rho(N_Q)$ is a function that encodes a small penalty for path sequences that include steps which incur orientation changes or backward steps. These terms have the combined effect of favoring paths with fewer steps, as well as slightly favoring paths with long sequences of forward, straight-line steps.

The final term represents an estimated cost from the current configuration to the goal region. $\mathcal{X}(Q, Q_g)$ approximates the minimum number of steps needed to traverse the straight-line distance between the footprint location at Q, and a footprint in the center of the goal region Q_g.

Each of the terms are weighted relative to each other by the factors w_d, w_ρ, and w_g. These values were determined experimentally, and are listed in Section 5. More research is needed to evaluate the effectiveness of different possible values for weighting parameters or alternative heuristic functions (see Section 6).

Obstacle Collision-checking: We require some kind of collision-checking algorithm to test the feasibility of potential footstep placement positions. If a node results in a footstep placement which causes a collision between the robot and an obstacle, that node is pruned from the search tree.

In our implementation, we used a two-level collision-checking strategy. The first phase uses a simple 2D polygon-polygon intersection test between the outline of the obstacle projection to the walking surface, and the outline of the foot geometry at the attempted footstep placement position. If the polygons intersect, the node is discarded. If not, then a 3D polyhedral minimum-distance determination method is used to verify that both the target footstep configuration, *and* the statically-stable trajectory which connects the current configuration to the target configuration are both collision-free. The minimum distance information allows us to enforce conservative safety bounds on the how close the robot geometry is allowed to approach obstacle surfaces. Stepping motions which result in the robot grazing too close to an obstacle surface can be either pruned from the search tree, or assigned a higher cost.

As an optimization, we can skip the 3D collision check for obstacles which have a "negative" height (e.g. holes in the floor, or other gaps in the walking surface). Furthermore, since the minimum distance determination is the most expensive part of the planning process, we adopt a *lazy-evaluation* approach to collision-checking. Instead of testing all successor nodes *prior* to insertion in the queue, we simply insert all successors and perform the minimum distance calculations *after* a node is removed from the priority queue. If there is a collision, we simply discard the node and extract the next node in the queue.

5 Experiments

This section presents some preliminary results using a prototype simulation environment. Figure 1 shows a cluttered office in which a model of the Humanoid robot "H6" must navigate. Figure 2 shows a top view of a footstep sequence computed to reach a circular goal region in the center of the room. There were a total of 15 discrete foot placements considered for each foot, and a total of 20 floor obstacles. The search tree contained approximately 6,700 nodes. Considering that the number of nodes required for a brute-force, breadth-first search on a footstep sequence length of 18 steps is more than 10^{21}, this is quite satisfactory. The path was computed in approximately 12 seconds on an 800 MHz Pentium II running Linux.

We used a polygon-polygon intersection test for the first phase of collision-checking, and the (*V-clip*) library (see [8]) for fast minimum distance determination between the obstacles and the convex hull of each of the leg links.

Figure 7 shows several snapshots while stepping over an obstacle during footstep sequence execution. All transition motions between subsequent footstep locations were accomplished via statically-stable trajectories. This seems to be fairly reasonable given the careful nature in which such motions would have to likely be performed by a real humanoid navigating in such a cluttered environment.

The weighting factors used for the cost function were $w_d = 1.0$, $w_\rho = 0.2$, and $w_g = 1.0$. These values were determined experimentally, and offered reasonable results.

6 Discussion

We propose a dynamic programming approach to planning safe navigation strategies for biped robots moving in obstacle-cluttered environments. By searching from among a discrete set of plausible statically-stable, single-step motions, we can account for the unique ability of legged robots such as humanoids to traverse obstacles by stepping upon or over them. Since it is a global planning approach, it does not suffer from deadlock or local loops like reactive methods. An experimental implementation of the approach has been shown to be reasonably efficient for moderately complex environments using standard PC hardware.

Several improvements can still be made, and these form the basis of our future work:

1. The ability of the robot to step upon the surface of obstacles.
2. Extending the method to handle environments with uneven terrain.
3. Incorporating visual or other sensor feedback during planning.
4. Designing and investigating the effects of different heuristics on the efficiency of the search.
5. Running the algorithm on a real humanoid.
6. Including dynamic stepping motions that increase the range of reachable footstep placements (including hopping or jumping between stable states).

Acknowledgments

This research is supported in part by a Japan Society for the Promotion of Science (JSPS) Postdoctoral Fellowship for Foreign Scholars in Science and Engineering, and by JSPS Grant-in-Aid for Research for the Future: "Research on Micro and Soft-Mechanics Integration for Bio-mimetic Machines (JSPS-RFTF96P00801)", as well as additional Monbusho Grant-in-Aids for Scientific Research.

References

[1] M. Girard. Interactive design of computer-animated legged animal motion. *IEEE Computer Graphics & Applications*, 7(6):39–51, June 1987.

[2] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'98)*, pages 1321–1326, May 1998.

[3] S. Hirose. A study of design and control of a quadruped walking vehicle. *Int. J. Robotics Research.*, 3(2):113–133, Summer 1984.

[4] L. Kanal and V. Kumar, editors. *Search in Artificial Intelligence.* Springer-Verlag, New York, 1988.

[5] E. Kokkevis, D. Metaxas, and N. I. Badler. Autonomous animation and control of four-legged animals. In *Proc. Graphics Interface*, pages 10–17, May 1995. ISBN 0-9695338-4-5.

[6] J. C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, MA, 1991.

[7] O. Lorch, J. Denk, J. F. Seara, M. Buss, F. Freyberger, and G. Schmidt. ViGWaM - an emulation environment for a vision guided virtual walking machine. In *Proc. IEEE Int. Conf. on Humanoid Robotics (Humanoids 2000)*, 2000.

[8] B. Mirtich. VClip: Fast and robust polyhedral collision detection. *ACM Transactions on Graphics*, 17(3):177–208, July 1998.

[9] K. Nagasaka, M. Inaba, and H. Inoue. Walking pattern generation for a humanoid robot based on optimal gradient method. In *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, 1999.

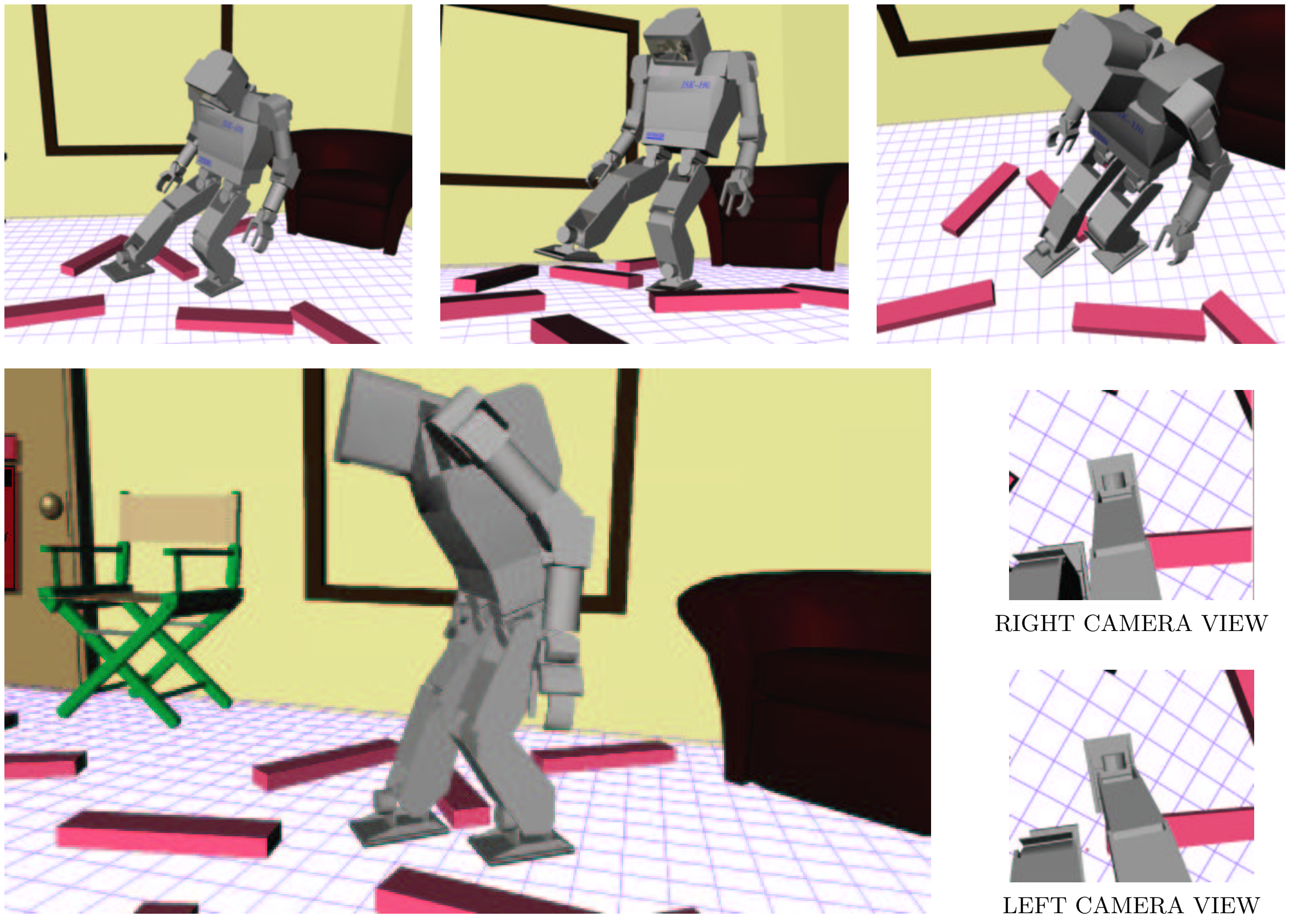

Figure 7: Simulation snapshots during footstep sequence execution.

[10] N.J. Nilsson. *Principles of Artificial Intelligence*. Tioga, Palo Alto, CA, 1980.

[11] K. Nishiwaki, T. Sugihara, S. KAGAMI, M. Inaba, and H. Inoue. Online mixture and connection of basic motions for humanoid walking control by footprint specification. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'01)*, Seoul, Korea, May 2001. *To appear.*

[12] J. Pratt and G. Pratt. Exploiting natural dynamics in the control of a 3d bipedal walking simulation. In *In Proc. of Int. Conf. on Climbing and Walking Robots (CLAWAR99)*, September 1999.

[13] N. Torkos and M. van de Panne. Footprint-based quadruped motion synthesis. In *Proc. Graphics Interface*, pages 151–160, 1998.

[14] M. van de Panne. From footprints to animation. In *Proc. Computer Graphics Forum*, volume 16, pages 211–223, October 1997.

[15] M. Vukobratovic, B. Borovac, D. Surla, and D. Stokic. *Biped Locomotion: Dynamics, Stability, Control, and Applications*. Springer-Verlag, Berlin, 1990.

[16] M. Yagi and V. Lumelsky. Biped robot locomotion in scenes with unknown obstacles. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'99)*, pages 375–380, Detroit, MI, May 1999.

[17] J. Yamaguchi, S. Inoue, D. Nishino, and A. Takanishi. Development of a bipedal humanoid robot having antagonistic driven joints and three dof trunk. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'98)*, pages 96–101, 1998.