

An Intelligent Joystick for Biped Control

Joel Chestnutt[†]

Philipp Michel[†]

Koichi Nishiwaki[‡]

James Kuffner^{†‡}

Satoshi Kagami[‡]

[†]Robotics Institute

[‡]Digital Human Research Center

Carnegie Mellon University

AIST Waterfront 3F

5000 Forbes Avenue

2-41-6, Aomi, Koto-ku, Tokyo

Pittsburgh, PA 15213

135-0064, Japan

{chestnutt, pmichel, kuffner}@cs.cmu.edu {k.nishiwaki, s.kagami}@aist.go.jp

Abstract—We present the concept of an “intelligent” joystick, an architecture which provides simple and intuitive high-level directional control of a legged robot while adjusting the actual foot placements autonomously to avoid stepping in undesirable places. The general concept can be likened to riding a horse: high-level commands are provided, while the “intelligence” of the underlying system selects proper foot placements with respect to the shape and properties of the underlying terrain and overall balance considerations. We demonstrate a prototype system used for realtime control of the humanoid robot HRP-2.

I. INTRODUCTION

Many successful 3D walking humanoids have been developed in recent years. In addition, many of these robots have demonstrated legged capabilities such as climbing stairs, stepping over obstacles, and walking on sloped surfaces. However, very little success has been achieved in autonomous operation in real environments. Furthermore, there is a lack of intuitive manual controls for navigation which can take advantage of the bipedal capabilities of these robots.

In addition to humanoid robots, research groups have developed walking machines capable of carrying humans. For such machines, a simple interface for user control is desired. However, to be more useful than a powered wheelchair, such control should be able to make use of the legs, without overwhelming the user.

We propose a system in which a joystick provides a simple directional interface to the user, while the underlying system incorporates sensor feedback and a model of the legged robot’s stepping capabilities to determine the best foot placement to accomplish the user’s command.

II. RELATED WORK

Since reliable, walking biped robots have been developed only recently, much less research attention has been focused on developing navigation strategies for bipeds. Most research has focused on pre-generating stable walking trajectories (e.g. [1]–[3]), or on dynamic balance and control (e.g. [4], [5]). Recently, techniques have been developed to generate stable walking trajectories online [6], [7], though these results do not account for obstacles. For quadruped robots, adaptive gait generation and control on irregular terrain and among obstacles has been previously studied [8]. This method has not yet been applied to biped robots. In biomechanics, researchers have studied the problem of how humans perform local planning over irregular terrain based on visual feedback [9],



Fig. 1. An Intelligent Joystick: simple control for complex behavior

[10]. Other work has been done to provide manual joystick control for current humanoid robots [11]. While this control works well for positioning the robot and testing walking and balance, it does not take into account any information about the robot’s environment. Due to this restriction, the operator must be careful to keep the robot away from any obstacles. Sensor-based obstacle-avoidance techniques have been developed for bipeds navigating in unknown environments [12], [13]. These approaches could likely be adapted to include joystick input to produce similar results to the approach described in this paper.

Autonomous navigation has achieved some success on several different robots [14]–[18]. However, these involve trying to find a path to a particular goal, and do not afford the user the same kind of simple direct control over the robot’s path. One particular area where a user would want direct control over the robot’s path is when the user is a “pilot” of the robot. Waseda University, Toyota Motor Corporation, and KAIST have all developed walking chairs (shown in Figure 2) capable of carrying a human while walking and balancing [19]–[21]. These robotic chairs have the potential to offer much greater mobility than wheelchairs. For this application, a simple interface to allow the rider to direct the movement of the chair is necessary. Currently, the iFoot and Hubo FX-1 are controlled via a joystick interface mounted on the chair.



Fig. 2. *Left*: Waseda WL-16R [19] *Center*: Toyota iFoot [20] *Right*: KAIST Hubo FX-1 [21]

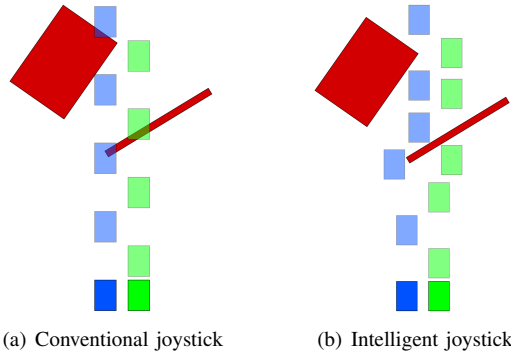


Fig. 3. Comparison of Conventional vs. Intelligent Joystick output given a command to walk forward.

We believe the usability of these devices could benefit from the concepts presented in this paper.

III. THE JOYSTICK CONCEPT

The idea of an intelligent joystick can be compared to riding a horse: the rider provides high-level control inputs about which direction to travel, but the horse handles all of the details of locomotion, including the complexities of selecting suitable foot placements and the overstepping of obstacles along the way. In the case of a legged robot, the joystick controls the overall movement direction of the robot, but the system autonomously selects foot placements and trajectories which best conform to the user's command given the constraints of balance and terrain characteristics. Figure 3 demonstrates how the intelligent joystick modifies the foot locations during a command to walk forward. A naive joystick controller generates the same walking pattern, regardless of obstacles in the path. An intelligent joystick will place the feet at the most suitable locations it can find while still making forward progress as commanded.

For controlling the humanoid robot in our experiments, we use a 3-axis joystick. This provides a simple mechanism to command forward motion, sideways motion, and rotation simultaneously through one interface.

A. What the Joystick Does

The joystick control system must convert a user command into a walking motion appropriate to the environment. This control system can be built on top of existing walking controllers that utilize online techniques for stable online walking pattern generation.

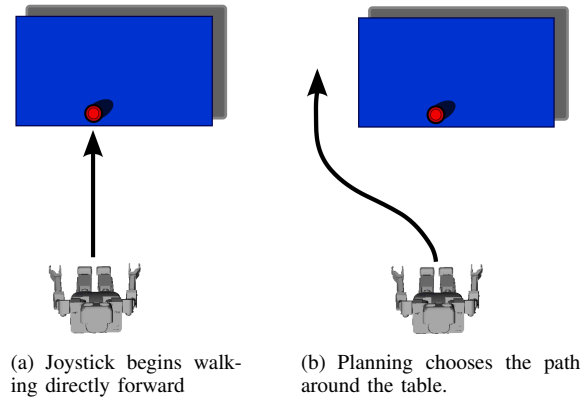


Fig. 5. Comparison of Joystick vs. Planning concepts

Given a robot stance location (x, y, θ) , an environment, e , and a joystick command $(\dot{x}, \dot{y}, \dot{\theta})$, the system's task is to determine the best walking motion to follow the user's command that will still maintain balance when the environment is taken into account. In other words, determine the next stance location (x', y', θ') , which satisfies balance requirements with respect to the environment, e , and brings the robot's velocity as close to the commanded $(\dot{x}, \dot{y}, \dot{\theta})$ as possible. To accomplish this, the system chooses a target location where it would like the next foot to land, and then evaluates that location and the locations nearby to determine the closest location to the desired target location that is most suitable for stepping onto. Figure 4 illustrates the selection of a target foot placement from a joystick command. This location is then sent to the walking control subsystem of the robot, for walking trajectory generation.

B. What the Joystick Does Not Do

The intelligent joystick is not providing an estimated goal to a planning system. The path is left entirely up to the user. Instead, the system attempts to place the feet at the best possible locations along a path in the direction that obeys the joystick commanded velocity as much as possible. This distinction means that the joystick is not providing full navigation autonomy, but rather is allowing the user to drive the overall direction of motion for the biped robot as one would drive a holonomic planar robot.

The example shown in Figure 5 demonstrates the difference in concept between the joystick controlling the direction of motion versus a destination. In Figure 5(a), if the user wants to move up to the table for a manipulation task, the user pushes forward on the joystick, and the robot walks as far forward as possible until it is blocked by the table. In Figure 5(b), if the user pushes forward on the joystick, the planning system may decide that the best way to accomplish that command is to walk around the table. This would make walking up near to the table's edge a potentially difficult task for the user. Furthermore, the direction the robot begins walking can be very different from the direction the user actually commanded via the joystick. This difference results from the fact that a

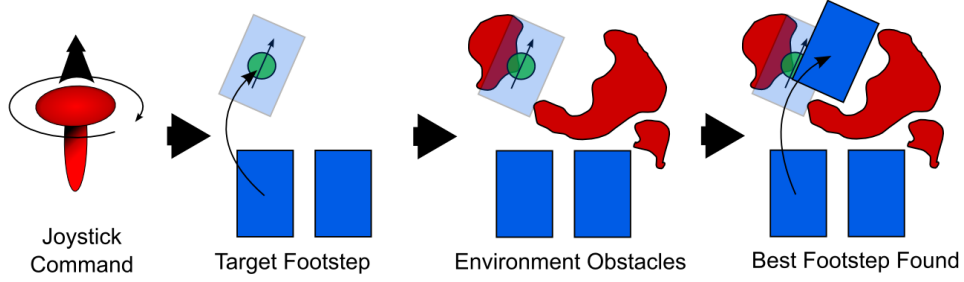


Fig. 4. Foot placement selection for a joystick command of forward while turning to the right

system based on high-level planning to a goal location may be trying to guess the user’s intention about the final destination from a simple joystick command. In the case of our Intelligent Joystick system, we opted to provide the user with control over the desired direction of movement. If the user wants the robot to walk around the table, the user can use the joystick to steer the robot around the table either clockwise or counter-clockwise.

IV. IMPLEMENTATION

A. Representations

A joystick command consists of three inputs: $j = (\dot{x}, \dot{y}, \dot{\theta})$, for forward motion, sideways motion, and rotation. Joystick commands are interpreted to be in the robot’s local coordinate frame.

We represent the environment, e , as a grid of cells. For a level floor in a pure 2D environment, each cell contains a boolean value, indicating whether it is free or contains an obstacle. For uneven terrain (2.5D environment), each cell holds both a boolean value for representing free space, as well as a terrain elevation value. Together, these elevation values describe the shape of the terrain.

The robot’s feet are represented by a footprint shape (in our case, rectangles with a specified width and length). The stepping capabilities of the robot are described by a set of locations, $\mathcal{S}$, at which the robot can place its next foot, relative to the current stance foot location. This is represented as a discrete set of 2D displacements, each corresponding to a potential step that may be taken. Each potential step has an associated cost, allowable obstacle clearance, and allowable height difference from the stance location.

B. Foot Placement Evaluation

Given the above representation of robot capabilities, the problem becomes determining the best safe and convenient valid step $s \in \mathcal{S}$, out of our discrete set of possibilities, that most closely follows the directional command given by the joystick from the current robot state. Valid steps are determined by the area of the terrain covered by a potential footstep location. For the 2D case, all cells under the foot must be free from obstacles. For the 2.5D case the area must be free from obstacles, and in addition must have a shape that the robot can safely step on. This criteria is determined by a set of metrics which assign a cost to a foothold location.

We use the terrain metrics developed in our previous work [15]. For a given step time, Δt , the ideal foot location can be determined from the joystick command (j) to be:

$$p_{next} = p_0 + \Delta t \cdot j$$

where p_0 is the “step-in-place” step location for the robot. We then need to search $\mathcal{S}$ to find the lowest cost step near to p_{next} . We define the cost of a step as:

$$cost(p) = FootCost(p, e) + w \cdot dist(p, p_{next}).$$

FootCost calculates how well a particular location in the environment serves as a foothold as well as the cost of actually taking the particular action, and *dist* is a distance metric determining how closely a step p matches the ideal step p_{next} . Finally, w is a weighting factor which allows us to adjust the tradeoff between closely following the joystick command, and choosing safer locations.

To find the best safe foot location, we precompute the ordering of best-to-worst foot locations from $\mathcal{S}$ for various ideal target footsteps in a location independent way (using only the *dist* part of the cost). Thus for some set of target footsteps, $\mathcal{F}$, each $f_i \in \mathcal{F}$ has a corresponding ordered list L_i containing the steps in $\mathcal{S}$. Once the ordering of closest to farthest for target locations has been precomputed, the online computation to find the best valid step is straightforward. The algorithm determines the nearest $f_i \in \mathcal{F}$ to use for the target footstep, which provides the precomputed list L_i . The locations, $p_j \in L_i$, must be traversed until $cost(p_j) \leq w \cdot dist(p_{j+1}, f_i)$. As *FootCost* will never be less than zero, the list does not need to be further traversed, and p_j can be chosen as the next step. In the 2D case, *FootCost* will either be zero if the step is valid, or infinite if the step is invalid, so the traversal of L_i can stop as soon as the first valid step is found.

For our *dist* metric, we have chosen to weight differences in rotation angle very highly, so that the robot will prefer steps with the commanded orientation, above all other steps, regardless of Euclidean distance to the target step. This decision was made due to the fact that joystick commands are interpreted in the robot’s local coordinate frame, so unexpected rotations make control much more difficult for the user. The robot may still make a step with a rotation different from the target step, but only if none of the steps with the target rotation are valid.

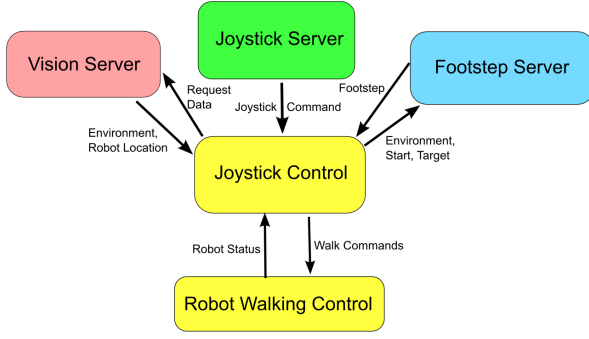


Fig. 6. Layout of the control system components

C. Control System

The intelligent joystick system is made up of several modular components as depicted in Figure 6.

a) *Joystick Server*: Provides the input commands from a joystick or game pad. For this application, we read three joystick axes, and map them to forward motion, side motion, and robot rotation.

b) *Vision Servers*: Provide information about the environment and robot location. This information can come from many sources other than vision, such as motion capture data, range finders, or pre-built models of the environment.

c) *Footstep Server*: Computes the best step to take based on the environment, robot location, and target step location.

d) *Joystick Control*: Communicates with all the servers to gather data, initiate footstep location search, and send commands to the robot.

e) *Robot Walking Control*: Control on the robot which handles all issues of balance and leg movement.

This modular design allows us to easily test individual components of the system, as well as swap in various assortments of sensor systems, joysticks, or robot models without changing any of the other system components.

Many current biped control systems do not have the ability to alter the swing leg trajectory during the execution of a step. Due to this limitation, the robot must know which step it will take next before it shifts support. Therefore, whenever the robot is evaluating the terrain and deciding on a footstep location, it is doing so based on a *future* stance location. For this reason, one of the main purposes of the Joystick Control component is to calculate where the robot will be at the next time it can change its trajectory, and send that information to the Footstep Server. The Footstep Server then performs its search based on this future stance location.

In Algorithm 1, `GetEnvironment`, `GetRobotLocation`, and `GetJoystickCommand` communicate with the various servers to acquire world state information. `ComputeStanceLocation` and `ComputeTargetLocation` perform the conversions necessary to determine where the future stance foot will be, and where the robot should step to follow the joystick command. `SendRobotCmd`, `StopWalking`, and `WaitForNextStep` are robot specific commands that handle the Robot Walking Control component.

Algorithm 1: Joystick Control algorithm

```

previous ← initial_value;
step ← initial_value;
while still walking do
    // Gather information
    env ← GetEnvironment();
    robot ← GetRobotLocation();
    joy_cmd ← GetJoystickCommand();
    // Compute start, target locations
    start ← ComputeStanceLocation(robot, step);
    target ← ComputeTargetLocation(start, joy_cmd);
    // Get step from Footstep Server
    step ← GetNextStep(env, start, previous, target);
    // Update robot command
    if step = NULL then
        StopWalking();
    else
        SendRobotCmd(step);
        WaitForNextStep();
        previous ← start;
    end
end

```

V. ROBOT RESULTS

We performed experiments of this system using both a real and simulated HRP-2 humanoid robot. The simulated tests were performed using the same system as the real robot, but with the Robot Walking Control component of the control system replaced with a simulated robot.

Sensing was performed by two different systems. First, we used an optical motion capture system, provided by Motion Analysis Corporation, comprised of 8 digital cameras operating at up to 240Hz and covering a $16m^2$ floor area. This system supplied us with real-time tracking data for objects in the environment which proved useful for both sensing and system monitoring and debugging. The motion capture system provided reliable information for localizing the robot, and specific obstacles in the environment. By registering 3D models of these objects with the tracking data, we could build appropriate 2D obstacle maps or 2.5D height maps in real-time. Second, by accurately localizing the head-mounted cameras using motion capture data, we were able to reconstruct the ground plane and use color segmentation to build 2D occupancy grids of the floor. Examples of these obstacles can be seen in Figure 8. When using the robot's on-board cameras in this manner, the Joystick Control component is connected to two separate Vision Servers: one that provides obstacle data from the cameras, and the motion capture Vision Server that provides robot position information.

The set of possible steps that we used in these experiments contained 496 different actions per foot. We experimented with larger set sizes, but there was little noticeable difference in the control or capability of the robot in the environments that we tested. The set of actions allowed the robot to step forward or

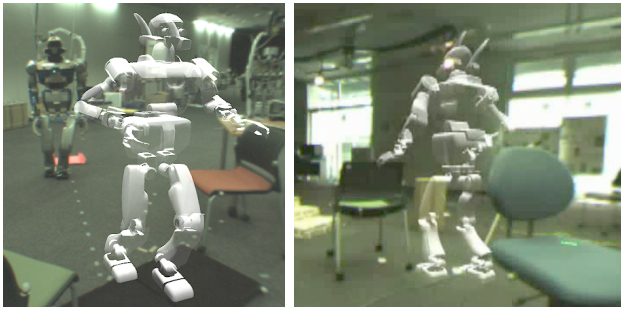


Fig. 7. Simulated robot overlaid on video of a real environment.

backward as much as 20cm. The range of allowed placement to the side was 19-33cm relative to the stance foot. The allowed change in rotation per step was ± 20 degrees. The size of the footprint was 13cm by 24.2cm.

A. Simulation

For the simulated HRP-2, we tested with both 2D maps generated from vision data or motion capture, and with 2.5D height maps generated from motion capture data. The simulation was run in real time and displayed as an overlay on a video of the environment, as shown in Figure 7. The vision data was generated from cameras onboard the real robot. Due to the additional uncertainty in computing obstacle maps from the vision data, we increased the safety margin for valid steps from 2cm (used for mocap data), to 5cm.

1) Specific Experiments:

Blockage in commanded direction: When commanded to walk forward into an obstacle, the robot approaches as close as possible to the obstacle and then steps in place. When facing an obstacle and commanded to move forward and to the side, the robot sidesteps along the blockage, with its movement direction conforming to the shape of the obstacle.

Stepping over an obstacle: For small obstacles, the robot will successfully position its feet to step over the obstacles. Due to the difficulty of placing markers on small objects, we only tested step-over with vision-detected obstacles.

Stepping around obstacles: When commanded to walk toward small obstacles, the controller successfully placed the feet at offset positions around the obstacle perimeters, while continuing in the direction indicated by the joystick.

Stepping onto obstacles: The system found reasonable foot locations for stepping onto obstacles of different heights. However, for these experiments, the swing leg trajectory was not modified from the default trajectory the walking controller creates, resulting in possible collisions with the objects the robot is trying to step onto. This issue will need to be addressed in future experiments.

B. Physical Robot

There were two significant differences between using a simulated robot and the real robot for experiments. First, by having the real robot stationary for the simulated tests, the quality of obstacle data received from vision was noticeably

better. Second, the real robot had small errors in execution that needed to be detected and for which we needed to correct, while the simulated robot had perfect execution of all commands. Due to these increases in the error of the system for the real robot, we increased the safety margin for valid footsteps to 10cm. Because of the lack of swing leg trajectory modification for stepping onto or over obstacles, we did not run these tests on the real robot. For that reason, the tests we performed with the real robot were limited to vision-generated 2D obstacle maps.

Blockage in path of command: The physical robot would safely stop forward motion and walk in place when it reached an obstacle which blocked its path. When commanded forward at an angle, it would successfully walk along the edge of the obstacle, following its shape.

Stepping over an obstacle: The increase in safety margin required for reliable operation unfortunately rendered the system incapable of stepping over obstacles. With a maximum step length of 20cm, the maximum foot travel is 40cm. The foot itself is 24.2cm long, so with a 10cm margin required at both ends, there is no room for an obstacle between the two foot positions. If the error in the system can be reduced, or the maximum step length of the robot increased, then experiments that include stepping over obstacles with the real robot will become possible.

Stepping around obstacles: Even with the increased safety margin, the joystick control system was able to successfully adjust footstep positions when given commands near obstacles, allowing it to remain safe while following the joystick directions, as shown in Figure 8.

VI. DISCUSSION

We have proposed the concept of an “intelligent” joystick, which provides a simple high-level interface for manually controlling a legged robot, while autonomously selecting foot placements that simultaneously consider commanded direction, balance, and terrain characteristics. We have implemented a prototype joystick control system, and demonstrated it on the humanoid robot HRP-2 in both simulation and actual experiments.

This method of operation is very applicable for safe, intuitive manual maneuvering of humanoids and other legged robots, as well as for controlling a robotic biped chair. In the near term, we would like to explore incorporating this method of future step validation into the low-level walking control of a humanoid, providing a level of safety despite potentially unsafe commands under all possible circumstances.

There are several drawbacks to the current implementation which we plan to investigate further. First, the current method of choosing the best nearest step involves a fixed set of samples, which can fail to find a possible step in severely constrained environments. An alternative continuous search strategy that does not rely on a fixed discretization of the possible walking motions would be preferable for this application. Second, the fact that the joystick control only looks at the next immediate step may not be sufficient for some

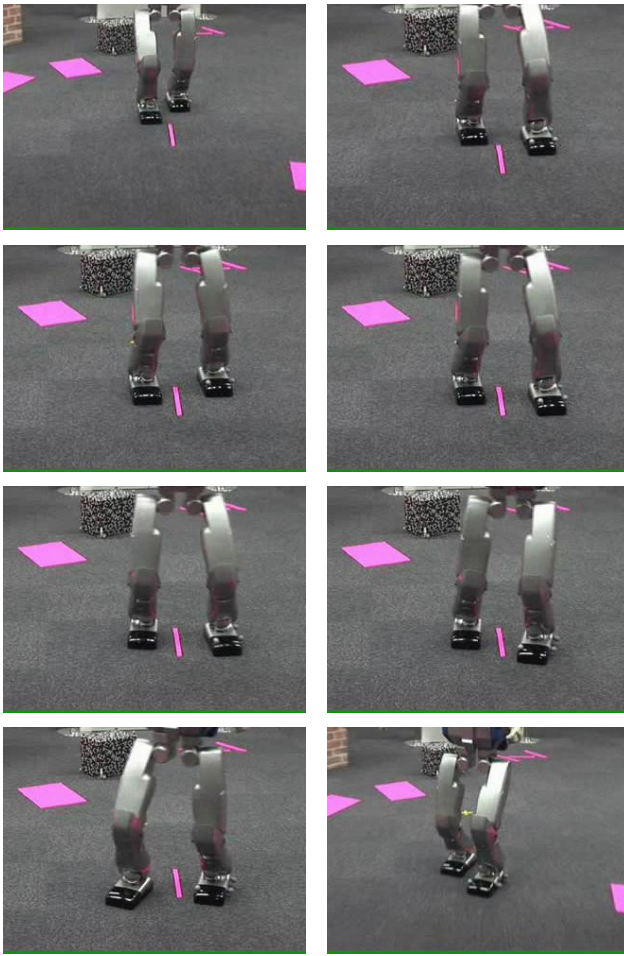


Fig. 8. Robot controlled via intelligent joystick. The given command is “forward and turning to the right”. The intelligent joystick executes the command while splaying its feet outward to avoid the small obstacle.

robots or situations. For example, a running robot may need to look several steps ahead to be certain it can safely avoid upcoming obstacles. Finally, due to the fact that the robot is deciding where to step based on a future stance location, there is a latency between issuing a joystick command, and when the robot takes a step in reaction to that command. This can potentially make the robot difficult or frustrating to control for some tasks. For example, when turning the robot to face a particular direction, it is easy to overshoot the desired orientation.

While using a joystick to control a humanoid may not be the ideal interface for some tasks and situations, it is a simple and intuitive control scheme for circumstances when the user wishes to control the walking direction at an intermediate level. The main drawback to such joystick control methods until now has been the fact that the walking motion did not take environmental information into account. By utilizing this information to select suitable foot placements, we can create intelligent joystick control systems that simultaneously combine ease of use with complex semi-autonomous underlying behaviors.

REFERENCES

- [1] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, “The development of Honda humanoid robot,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1998, pp. 1321–1326.
- [2] J. Yamaguchi, S. Inoue, D. Nishino, and A. Takanishi, “Development of a bipedal humanoid robot having antagonistic driven joints and three dof trunk,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998, pp. 96–101.
- [3] 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.
- [4] M. Vukobratovic, B. Borovac, D. Surla, and D. Stokic, *Biped Locomotion: Dynamics, Stability, Control, and Applications*. Berlin: Springer-Verlag, 1990.
- [5] J. Pratt and G. Pratt, “Exploiting natural dynamics in the control of a 3d bipedal walking simulation,” in *Proceedings of the International Conference on Climbing and Walking Robots (CLAWAR99)*, Portsmouth, UK, September 1999.
- [6] 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 *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001.
- [7] K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, “Online generation of humanoid walking motion based on a fast generation method of motion pattern that follows desired zmp,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, pp. 96–101.
- [8] S. Hirose, “A study of design and control of a quadruped walking vehicle,” *International Journal of Robotics Research*, vol. 3, no. 2, pp. 113–133, Summer 1984.
- [9] A. Patla, A. Adkin, C. Martin, R. Holden, and S. Prentice, “Characteristics of voluntary visual sampling of the environment for safe locomotion over different terrains,” *Exp. Brain Res.*, vol. 112, pp. 513–522, 1996.
- [10] A. Patla, E. Niechwiej, and L. Santos, “Local path planning during human locomotion over irregular terrain,” in *Proc. AMAM2000*, 2000.
- [11] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, “Self-collision detection and prevention for humanoid robots,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, D.C., May 2002.
- [12] M. Yagi and V. Lumelsky, “Biped robot locomotion in scenes with unknown obstacles,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, MI, May 1999, pp. 375–380.
- [13] 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 *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2000.
- [14] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, “Footstep planning among obstacles for biped robots,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 500–505.
- [15] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, “Planning biped navigation strategies in complex environments,” in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Karlsruhe, Germany, October 2003.
- [16] J. Chestnutt, M. Lau, G. Cheng, J. Kuffner, J. Hodgins, and T. Kanade, “Footstep planning for the Honda ASIMO humanoid,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.
- [17] K. Sabe, M. Fukuchi, J.-S. Gutmann, T. Ohashi, K. Kawamoto, and T. Yoshigahara, “Obstacle avoidance and path planning for humanoid robots using stereo vision,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, April 2004.
- [18] P. Michel, J. Chestnutt, J. Kuffner, and T. Kanade, “Vision-guided humanoid footstep planning for dynamic environments,” in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, December 2005.
- [19] Y. Sugahara, T. Hosobata, Y. Mikuriya, H. Sunazuka, H. Ok Lim, and A. Takanishi, “Realization of dynamic human-carrying walking by a biped locomotor,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, New Orleans, April 2004.
- [20] Toyota, “Toyota.co.jp -news release-,” December 2004. [Online]. Available: <http://www.toyota.co.jp/en/news/04/1203.1d.html>
- [21] M. C. L. KAIST, “Hubo fx-1,” November 2005. [Online]. Available: http://www.ohzlab.kaist.ac.kr/robot/fx_intro.html