Motion Planning of Ladder Climbing for Humanoid Robots

Yajia Zhang, Jingru Luo, Kris Hauser
School of Informatics & Computing
Indiana University
Bloomington, IN 47408

Robert Ellenberg, Paul Oh
Mechanical Engineering and Mechanics
Drexel University
Philadelphia, PA 19104

H. Andy Park, Manas Paldhe, C.S. George Lee
School of Electrical & Computer Engineering
Purdue University
West Lafayette, IN 47907

***Abstract*—This paper describes preliminary steps toward providing the Hubo-II+ humanoid robot with ladder climbing capabilities. Ladder climbing is an essential mode of locomotion for navigating industrial environments and conducting maintenance tasks in buildings, trees, and other man-made structures (e.g., utility poles). Although seemingly straightforward for humans, this task is quite challenging for humanoid robots due to differences from human kinematics, significant physical stresses, simultaneous coordination of four limbs in contact, and limited motor torques. We present a planning strategy for the Hubo-II+ robot that automatically generates multi-limbed locomotion sequences that satisfy contact, collision, and torque limit constraints for a given ladder specification. This method is used to automatically test climbing strategies on a variety of ladders in simulation. This planner-aided design paradigm allows us to employ extensive simulation in order to rapidly design, test, and verify novel climbing strategies, as well as testing how candidate hardware changes would affect the robot's ladder climbing capabilities.**

I. Introduction

Humans are capable of diverse modes of locomotion such as bipedal walking, stair climbing, crawling, ladder climbing, and brachiated movements, so humanoid robots have a potential advantage over other morphologies in human-made environments like homes, office buildings, or industrial environments. As such humanoids are natural candidates for addressing the DARPA Robotics Challenge (DRC), which poses the ambitious goal of providing robots human-like navigation, manipulation, and perception capabilities in a hazardous disaster site. Using the Hubo-II+ humanoid platform, this paper attacks the problem of ladder climbing. Although existing general-purpose humanoid platforms are capable of walking on even and uneven terrains, crawling, and climbing stairs, it has still proven challenging to realize ladder climbing and other motions that require significant arm strength for stability.

Since the early work of biped gait control and synthesis by Vukobratovic [1] in 1969 and WABOT developed by Wasada University in 1973 [2], existing humanoid robots such as Hubo-II+ robot and ASIMO are more versatile in performing locomotion movements. Kanehiro et al. developed locomotion planning algorithms for humanoid robots in selecting locomotion styles/postures to pass through narrow spaces imposed by different environments [3].

Special purpose robots have successfully climbed vertical surfaces. Kim et al. developed a robot that can climb on a flat and smooth surface using a hand and foot similar to a gecko [4]. Bretl et al. studied multi-step motion planning for a vertical rock climbing robot [5], [6]. Iida et al. developed a ladder-climbing robot called LCR-1 for vertical ladder climbing [7]. Yoneda et al. developed a vertical ladder-climbing humanoid robot that is specially designed for climbing; it uses hook-like hands to maintain balance during the climb, and force sensors to detect whether a rung was successfully grasped [8]. Unlike general-purpose humanoids, these special-purpose climbing robots have limited or nonexistent capabilities to perform manipulation, stair-climbing, and transitioning from flat ground to ladders and vice versa.

For a successful ladder climbing, motion planning for a sequence of hands-feet placement must be performed to detect collisions between the robot's body and the rungs of the ladder as well as gripping forces/torques for the hands and the stepping forces/torques of the feet on the rungs. Early work on motion planning for humanoid robots focuses on footstep planning for biped robots such as Kuffner's footstep planning [9], Kagami's vision-based footstep planning [10], and LaValle and Kuffner's RRT planning algorithm [11]. Recently, Hauser et al. [12], [13], [14], [15] have extended motion planning to humanoid robots with multi-limb contacts to achieve balance.

Although humanoid robots usually appear similar to humans, even seemingly subtle differences in numbers of joints, joint ranges, limb lengths, and actuator strengths can cause dramatic changes in locomotion capabilities. *Human strategies can and do fail when applied directly to humanoids*. So, we approach ladder climbing as an unsolved problem for the robot to solve *de novo*. We use powerful multi-limb motion planning algorithms that generate detailed plans for the robot to climb a specified ladder. A plan includes:

- A sequence of limbs to be placed and removed against the terrain.

†This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) award #N65236-12-1-1005 for the DARPA Robotics Challenge. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

†Yajia Zhang is supported in part by the National Science Foundation under Grant No. 1218534.

†H. Andy Park is also supported in part by the National Science Foundation under Grant CNS-0960061.

978-1-4673-6225-2/13/$31.00 ©2013 IEEE

- Contact points and orientations for those limbs.
- Joint-level trajectories showing the robot's poses that achieve those contacts while avoiding collision and respecting kinematic limits.
- Forces and torques that realize stable balancing at all points along the trajectory.

The planner is quite general; it accepts arbitrary robot models, ladder models, and surrounding obstacles. Although the planner is developed to use the principles of optimization to solve problems "from scratch" if necessary, it is also designed to adapt knowledge from previous plans or from human experts as "suggestions" to help it solve novel ladder climbing problems faster. In this paper we demonstrate how this capability allows us to generate forward climbing motions, and testing hardware variations in simulation. This capability allows our team to make informed and prioritized recommendations before embarking on expensive and time-consuming changes to the hardware platform. Our investigations uncovered three key variables that affect ladder-climbing capabilities: grip strength, hip flexibility, and lower-leg thickness.

II. Hubo-II+ Robot and OpenHubo Simulation Platform

A. Hubo-II+ Humanoid Robot

Hubo-II+ robot is a highly-articulated, full-size humanoid robot designed and built by Rainbow Co., a spin-off company from HuboLab at the Korean Advanced Institute of Science and Technology (KAIST). Hubo-II+ robot (see Fig. 1) is about 130 cm tall and weighs about 42 kg with 38 degrees of freedom (DoFs), 6 in each leg, 6 in each arm, 5 in each hand, 1 in the waist and 3 in the neck. The normal walking speed is 1.8 km/hr and the maximum walking speed is 3.6 km/hr. The robot is equipped with a 3-axis F/T sensor module, a 3-axis accelerometer, and a 1-axis rate gyro sensor at each ankle. A normal force and acceleration along vertical axis, two moments and inclinations about pitch and roll axes, and angular velocity about yaw axis can be measured. Furthermore, it is equipped at each wrist with a 3-axis F/T sensor, which measures one force along and two moments about the same axes. At the hip of the waist position, an inertial sensor is located and it measures accelerations along 3 axes and angular velocity about pitch and roll axes.

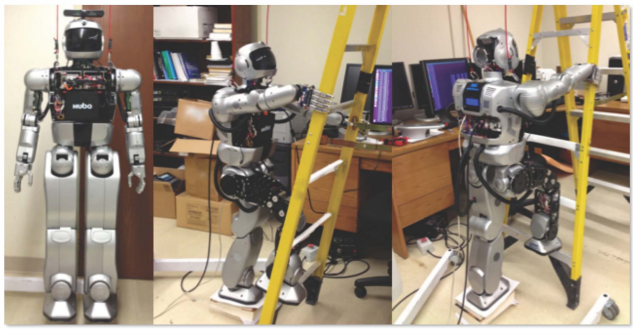

Fig. 1. A Hubo-II+ robot and an experimental setup for the ladder-climbing task.

Two computer modules are mounted at the chest with one unit (body computer) being used for real-time whole-body joint control through Controller Area Network (CAN) and the other unit (head computer) for real-time vision processing. Currently, a Logitech HD webcam (C905) mounted on the head is being used to provide the surrounding information in the form of color images. These color images will be used to detect the presence and identify the specification of a ladder, which will be used for the motion-planning module to generate a sequence of hands-feet (or limbs) placement for ladder climbing.

B. OpenHubo Simulation Platform

The OpenHubo simulation environment (see Fig. 2) is a simulation tool being developed by Drexel University. Its purpose is to provide motion planning tools that include the Hubo-II+ robot dynamics and controllers in a virtual environment, featuring physics-based simulation, simulated sensors, and controller portability to the Hubo-II+ hardware. OpenHubo is built on top of Open Robotics Automation Virtual Environment (OpenRAVE) [16], an open-source planning and simulation framework. OpenRAVE is widely used in the robotics community for robotic manipulator motion planning. It includes a variety of sampling and inverse-kinematics-based planners, as well as a database of common robot models.

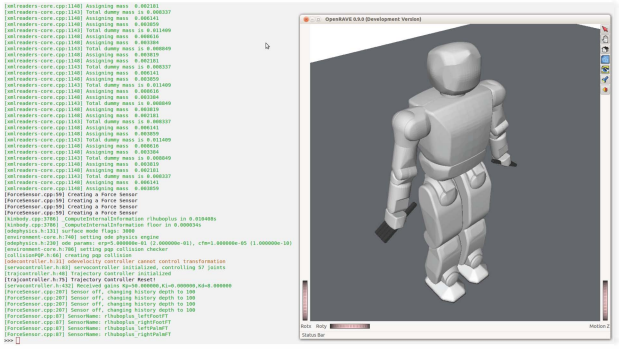

Fig. 2. OpenHubo environment showing python scripting interface and OpenRAVE GUI.

OpenHubo includes collision meshes, mass and inertia properties for Hubo-II+ robots. A servo-control plugin simulates the position control of the Hubo-II+'s servomotors, allowing PID control of joints. Hence, OpenHubo is a simulation tool set, where motions can be planned quickly using Hubo-II+ robot kinematics, verified using physics-based simulation with sensor feedback, and validated directly on the physical Hubo-II+ robot, all within the same software environment. In this paper, OpenHubo will be used to verify the performance of our motion-primitive-based planner for ladder-climbing.

III. Ladder-Climbing Motion Planning

We have designed and developed a motion planner for ladder-climbing motions based on the idea of *motion primitives* [13], which are motions that make or break a single limb contact. The motion planner takes a robot model and a ladder specification as input (see Fig. 3(a)) and outputs a plan that is executed either in simulation or on the physical robot. The ladder specification has two basic entities, the stringer and the rungs, and includes a variety of parameters describing the ladder inclination, rung spacing, stringer width, and cross-sectional geometries. Current cross-sections are allowed to be

either circular or rectangular although this may be extended in future implementations. Each ladder specification defines 3D geometry for collision testing in OpenHubo simulation.

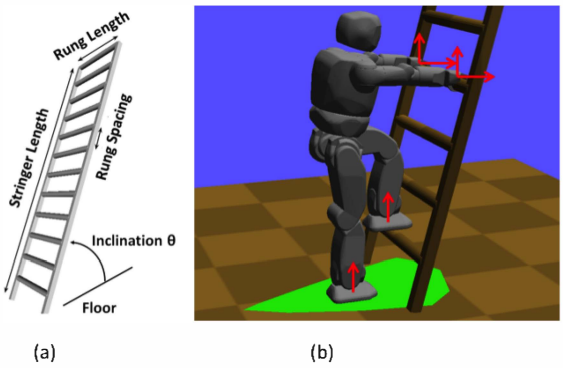

Fig. 3. (a) Parameter specification of a 3D ladder model. (b) Point-contacts and their normals (red arrow). A support polygon (green region) can be calculated based on the contacts to check the stability of the robot.

During climbing, up to four limbs will be in contact with the ground or the ladder to provide support. We define the concept of a *hold* as a region in which the geometry of a single robot link and environment touch. A list of holds yields a *stance*. While on the ground, a stance σ consists of two holds, and during climbing, a stance may consist of three or four holds (i.e., either two feet and a hand, two hands and a foot, or all four limbs in contact). To model a contact region, we use finite number of point-contacts where the points $r_1, \ldots, r_k$ on the robot touch the points $x_1, \ldots, x_k$ in the environment, with contact normals $n_1, \ldots, n_k$. For simplicity, we model hand holds with two point-contacts — one vertically-oriented to allow the hand to push down, and one horizontally-oriented to allow the fingers to pull back — and foot holds on a rung with one vertically-oriented point-contact (see Fig. 3 (b)). Coulomb friction is assumed, with a known coefficient of friction (0.4 is used in our examples).

A feasible robot configuration q at a stance σ must satisfy the following constraints:

1) Contact constraints: the points $r_1(q), \ldots, r_k(q)$ meet the points $x_1, \ldots, x_k$, for all holds in σ.
2) Within joint limits: $q \in [q_{min}, q_{max}]$.
3) Free of collision with the environment.
4) Free of self-collision.
5) Stable under gravity: the center of mass of the robot should lie within the support polygon formed by the supporting limbs. Note that with uneven contacts, a support polygon does not necessarily correspond to the convex hull of the projection of the contacts (see Fig. 3 (b)). We use Bretl's method to compute this polygon [17].

A solution to the ladder-climbing planning problem is a sequence of stances $\sigma_1, \ldots, \sigma_n$ and a continuous sequence of single-step paths of feasible configurations $p_1, \ldots, p_n$ that are feasible at their respective stances. Note that at the transition between stances σ_i and σ_{i+1}, the robot must pass through a configuration that meets the constraints at *both* stances.

We decompose a ladder-climbing motion into a list of motion primitives according to different limb-contact conditions:

1) placeHands: place two hands on a (chosen) rung.
2) placeLFoot: place left foot on the first rung.
3) placeRFoot: place right foot on the first rung.
4) moveLHand: lift left hand to the next higher rung.
5) moveRHand: lift right hand to the next higher rung.
6) moveLFoot: lift left foot to the next higher rung.
7) moveRFoot: lift right foot to the next higher rung.

A motion primitive typically causes a limb to be removed and then placed it at a new location, forming two stance changes. We utilize primitives 1–3 to mount a ladder, and primitives 4–7 are then repeated to climb the ladder for a desired number of rungs.

Each motion primitive is designed to contain prior knowledge for solving that portion of the climbing task. It contains an "ideal" set of point-contacts, robot poses, and intermediate waypoints tailored to the action (see Fig. 5). In our current implementation these are designed by a human expert, but in future work we hope to learn them from experience. The "ideal" values are used as *seeds* to help the planner find natural-looking contacts, poses, and paths for novel ladders. Since our planner is based on optimization, a good starting point is critical to avoid local minima and reduce the cost of optimization. They impose an implicit preference for natural-looking paths, because our planner is designed to make minimal changes from the starting primitive. Our experiments showed that good seeds can greatly speed up the process of finding collision-free and stable paths.

We next describe how to plan and utilize primitives 1–7 in sequence to climb up two rungs. Climbing multiple rungs simply requires repeating primitives 4–7. Our method is a randomized sequential descent, in which each primitive is slightly perturbed from the seed values at random in order to help find successful solutions. To ensure that paths stay close to the seed primitives, the radius of perturbation starts at zero and increases upon subsequent iterations. This procedure is described by the following pseudocode:

1. Repeat until a solution is found, or a time limit is reached:
2. Let $q_0 \leftarrow q_{cur}$ and $\sigma_0 \leftarrow \sigma_{cur}$.
3. For motion primitives, $1, 2, \ldots, 7$, do:
4. Sample a desired hold h_d near the seed hold.
5. Let σ_i replace the current hold in σ_{i-1} with h_d.
6. Sample a feasible destination configuration q_i at σ_i.
7. Find a feasible path connecting q_{i-1} to q_i at σ_{i-1}.

The innermost loop samples holds, configurations, and paths in that order (see Fig. 4). If any step in the innermost loop fails, the planner restarts from step 2. Each innermost sampling step is run for n samples, where n controls the balance of putting more effort on one action or backtracking to get a better start. In our implementation, n is set to 50 after tuning.

Starting from a seed configuration q_{seed}, we use a numerical inverse kinematics (IK) solver to obtain a configuration that satisfies IK and joint limit constraints. Moreover our planning system can retract slightly colliding configurations out of collision by solving a nonlinear constrained optimization process, similar to the Iterative Constraint Enforcement algorithm [12]. If this fails, a perturbation function is used to adjust q_{init} with perturbation drawn uniformly from 0 to some radius c, which is chosen empirically. The perturbation radius increases

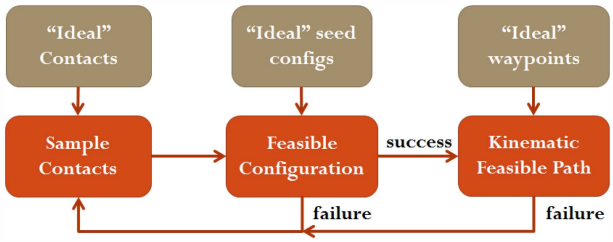

Fig. 4. Three steps in motion planning for ladder climbing based on motion primitives. If one step fails, the planning process will trace back to the previous step. Prior information is being used to assist in each step.

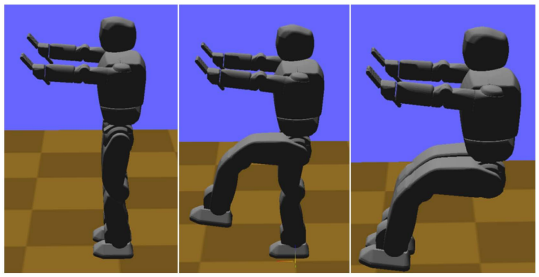

Fig. 5. Seed examples. From left to right: placeHands, liftLFoot, liftRFoot.

as the number of failures increases. The process stops until we find a feasible configuration or it reaches the iteration limit. Algorithm 1 describes a configuration-finding procedure.

Algorithm 1 Finding feasible configuration

```
for i = 0, 1,..., n: do
    q_init = q_seed + perturb(i)
    if find q from IK solver starting from q_init: then
        if no self- and no envr-collision and stable: then
            return q
        if no self-collision and has envr-collision: then
            if retract(q) succeeds and q is stable: then
                return q
```

To generate feasible trajectories that connect the starting and ending configurations of motion primitives, we add intermediate waypoints to avoid collision with the ladder rungs. These waypoints are solved by interpolating the endpoints of the moved limb along an arc in world space. Again, perturbations are used to push waypoints into the feasible space.

We also employ a contact-space interpolation strategy to smoothly connect these waypoints. Simple linear interpolations in joint space does not work because the robot fails to maintain contact at intermediate configurations (a problem known as foot-skate in the animation literature). Instead we use a recursive interpolation to ensure that that the supporting limbs remain in contact up to some user-defined resolution ϵ. Given two endpoint configurations, q_1 and q_2, our strategy first finds the middle point $q = \frac{1}{2}(q_1 + q_2)$ in the joint space, then calculates its projection q' in the contact space. The projection function uses numerical IK to find q', which satisfies the IK constraints that $r_1(q'), \ldots, r_k(q')$ meet $x_1, \ldots, x_k$. If successful, we then recursively solve two sub-problems: interpolation between q_1 and q', and interpolation between q' and q_2. The algorithm terminates the recursion once q_1 and q_2 are closer than ϵ.

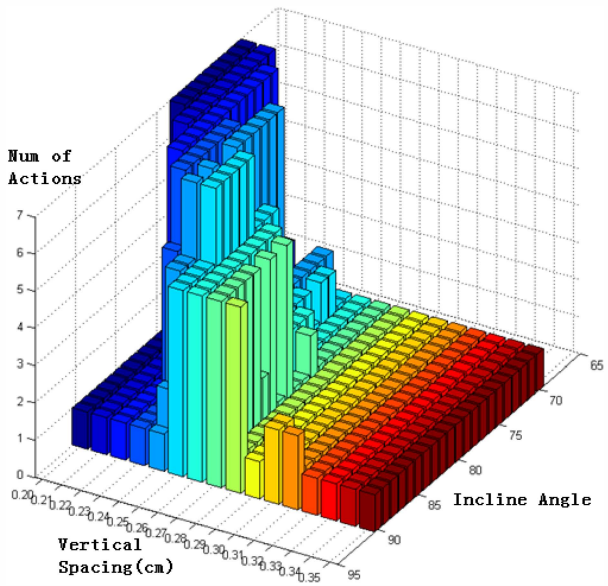

Fig. 6. Computer simulation results on ladders with inclined angle between 70° and 90°, and the rung spacing between 20 cm and 35cm. The height of the bar indicates how many sequential primitive actions were successfully planned.

IV. Computer Simulation Results

Extensive computer simulations were performed to test our ladder-climbing motion planner with Hubo-II+ robot model and a list of ladders of various specifications as input. Two parameters for ladders are considered: the rung spacing ranging from 20 cm to 35 cm with 1 cm increment, and the incline angle ranging from 70° to 90° with 1° increment. For each ladder, we tested our motion planner by utilizing all 7 motion primitives. All the experiments were carried out on an Intel Core i7 2.8 GHz machine with 4GB RAM. The planner was run on each ladder with a 60-second cutoff time.

Figure 6 shows the motion planning results for each ladder, the height of the bar indicates how many sequential motion primitives were successfully planned. Among all 756 ladders, 15.48% could be fully solved by our planner (i.e., succeeded in utilizing all 7 motion primitives) and 25.30% could be solved for all the mounting actions (i.e., the first 3 motion primitives).

By observing the cases that our motion planner failed in finding a solution to a specific ladder, we found that two parameters of Hubo-II+ robot play an important role: the joint limits of the leg pitch and the geometric size of the knee joints. Limited leg pitch prevents the leg lifting higher, which reduces the spacing requirement in a cluttered environment. "Fat" knee joint frequently causes collisions with the ladder with high inclination. We thus designed three experiments to verify our conjecture:

1) Increase the leg pitch joint limits by 10°.
2) Shrink the knee joints by 1.5 cm (In practice, this probably can be done by removing the shells of the knee joints since they are not tightly designed).
3) Apply the above both changes.

Every experiment was carried out with other robot and

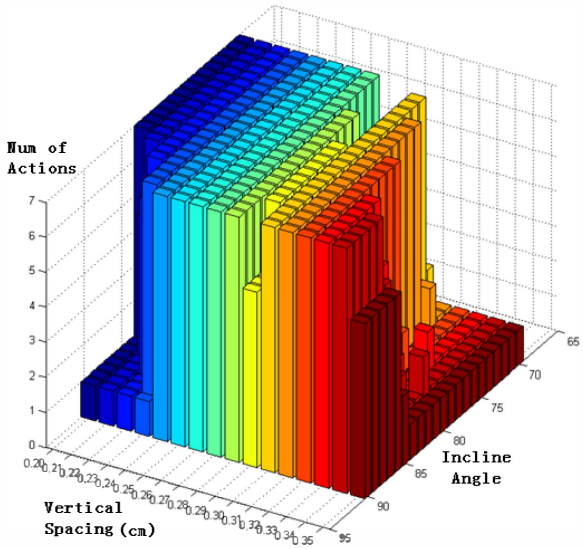

Fig. 7. Batch test result by using modified Hubo-II+ robot model. The fully solved rate is boosted to 70.24% while mounting success rate is boosted to 73.81%.

ladder settings remained the same. The experiment results showed that by increasing the leg-pitch limit alone, the fully solved ladder-climbing ratio changed to 43.45%; by shrinking the knee joints by 1.5 cm alone, the ratio changed to 34.82%; by applying both modifications, the ratio changed to 70.24% (see Fig. 7). This demonstrated that our motion planner can be utilized to effect hardware re-design.

OpenHubo Simulation. The motion planner produces trajectories that satisfy kinematic and posture constraints of Hubo-II+ robot. To validate the planner, the OpenHubo platform was used to simulate a subset of these planned motions. The OpenHubo simulation package considers the dynamic behavior of Hubo-II+ robot as well as the effects of contacts and friction. If a planned motion is executed correctly with full physics considered, then the assumptions and simplifications of the kinematic model and constraints are more trustworthy, and future controller design can be tested and verified in OpenHubo before the controller is ported to the Hubo-II+ robot. Enabling physics in the simulation reveals important difficulties in the ladder climbing motion that are not necessarily apparent from kinematic planning. For example, when multiple contacts are established on the ladder, the hands and feet are not always accurately placed, due to the weight of the robot. Position-controlled joints respond aggressively to small joint errors due to their high gains. Thus, in multi-contact climbing poses, incorrect placement of limbs creates extra force that must be cancelled by the other supporting limbs (see Fig. 8).

Table I shows the set of ladders chosen for full simulation with OpenHubo. For each ladder, there are two independent improvements considered. The first is a 10° increase in each of the three leg pitch joint limits, and the second is the reduction in leg thickness equivalent to the removal of the plastic shells. Grip strength is assumed to be artificially large (10.0 Nm maximum finger joint torque) for this initial validation, while the ladder rungs were assumed to be cylindrical with a diameter of 4 cm. A best-case static friction coefficient of 2.0 was assumed for all contacts due to the use of soft rubber pads

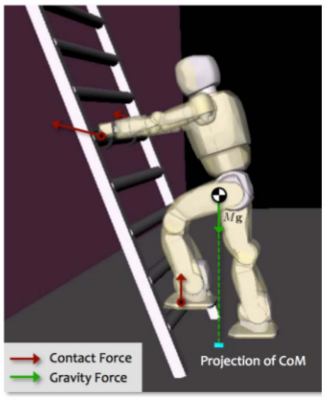

Fig. 8. Contact forces produced by Hubo-II+ robot's hands and feet.

TABLE I. PREDICTED SUCCESSES OF SELECT CASES WITH A STOCK HUBO-II+ AND THE MODIFICATIONS RECOMMENDED IN OUR MOTION PLANNER.

		Prediction		Result	
Angle	Spacing	Stock	Modified	Stock	Modified
70°	20 cm	7	7	1	7
70°	25 cm	1	7	1	4
75°	22 cm	7	7	1	4
75°	25 cm	1	7	1	4
80°	25 cm	7	7	1	4
80°	30 cm	1	7	1	4
85°	25 cm	7	7	1	2
85°	30 cm	1	7	1	2
90°	25 cm	7	7	0	0
90°	32 cm	1	7	0	0

on the hands and feet of the Hubo-II+ robot. During climbing motions, Hubo-II+ robot was capable of reaching for the next rung and completing the motion for the 70° and 75° cases, but not for steeper angles.

The simulated Hubo-II+ robot was able to mount eight of the ten ladders only when using the slim legs and increased range of motion of the shell-less version of the robot. Without both of these improvements, the robot's lower legs collided with the ladder, pushing the robot away and causing it to fall. Contact error was another factor – the large moment created by the robot hanging far from the ladder caused the robot to tilt horizontally away from the ladder, in turn causing the reaching motion moveLHand to fall short (see Fig. 9). The left foot also slipped noticeably backwards in these cases.

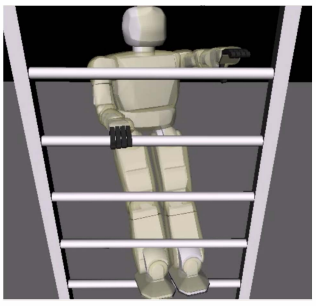

Fig. 9. Joint limits cause the left foot to fall short (left), and climbing onto a 80°, 25 cm-spaced ladder causes a slip, leading to misalignment (right).

For the 90°-inclined ladder case, the fast swing-up motion of the arms added extra torque to the robot, tipping it backwards. Based on these results, the accuracy of hands and feet

placements represents an important challenge for the physical Hubo-II+ robot. The robot must be able to detect an improperly placed hand/foot, or be able to detect when a hand or foot has slipped from the desired location.

For the most successful case, the 70° and 20-cm-spaced ladder, the previous experiment was repeated using a maximum finger strength corresponding to the actual Hubo-II+ robot. The current hands have been shown to support a weight of approximately 4.5 kg. Simulation of a hand grasping a weighted bar (see Fig. 10) showed that the fingers could support the weight of the bar with 1.6 Nm maximum torque on the thumb joints, and a maximum of 0.8 Nm at each finger joint. This measurement is not meant to exactly model each physical finger, rather, it is a simulation reference point that shows how much stronger the hands must be to successfully climb our chosen range of ladders.

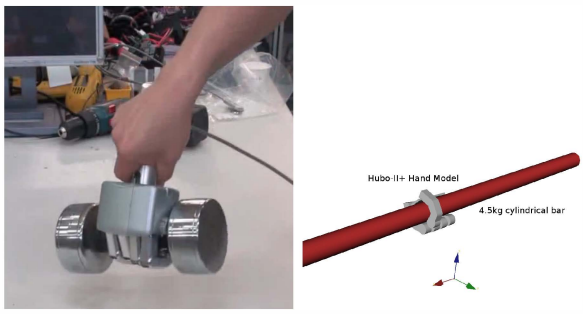

Fig. 10. Physical Hubo-II+ robot hand supporting 4.5 kg (left), and a simulated grasp with finger torque of 1.6 Nm.

For a 70°, 20-cm-spaced ladder, a complete climb was possible with a maximum finger torque of 4.0 Nm (see Fig. 11), or approximately twice the current finger strength. The 4 mounting placements and 3 climbing placements were completed in 31 seconds.

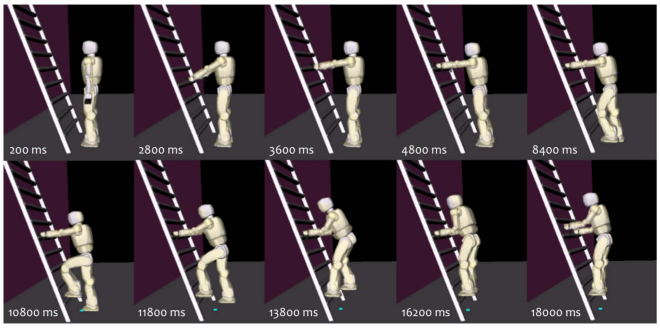

Fig. 11. The Hubo-II+ robot climbed onto a 70°-inclined ladder and took a step, using maximum finger torque of 4.0 Nm.

V. Summary and Conclusion

A motion-primitive-based planner was developed to generate multi-limbed locomotion sequences for ladder climbing for Hubo-II+ robot, which satisfy contact, collision, balance, and torque limit constraints for a given ladder specification. The performance of the planner was validated by extensive computer simulations to verify ladder-climbing strategies on a variety of ladders with different specifications. In addition to verifying the climbing strategies, the proposed motion planner also allows the user to test how hardware design changes would affect the robot's ladder climbing capabilities. OpenHubo simulation package was also used to verify the planner's performance with robot dynamics and controller taken into consideration. Our motion planner also identified two major robot parameters that affect Hubo-II+ robot's capability in ladder climbing. These two parameters are the joint limits of the leg pitch and the geometric size of the knee joints. By slightly increasing the leg pitch joint limit by 10° and shrinking the size of the knee joints by 1.5 cm, Hubo-II+ robot was found to be capable of climbing more ladders of different specifications.

References

[1] M. Vukobratovic and D. Juricic, "Contribution to the synthesis of biped gait," *Biomedical Engineering, IEEE Transactions on*, vol. BME-16, no. 1, pp. 1–6, jan. 1969.

[2] I. Kato, "Development of WABOT 1," *Biomechanism*, vol. 2, pp. 173–214, 1973.

[3] F. Kanehiro, H. Hirukawa, K. Kaneko, S. Kajita, K. Fujiwara, K. Harada, and K. Yokoi, "Locomotion planning of humanoid robots to pass through narrow spaces," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 1, 2004, pp. 604–609.

[4] S. Kim, M. Spenko, S. Trujillo, B. Heyneman, V. Mattoli, and M. Cutkosky, "Whole body adhesion: hierarchical, directional and distributed control of adhesive forces for a climbing robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, April 2007, pp. 1268–1273.

[5] T. Bretl, S. Lall, J.-C. Latombe, and S. Rock, "Multi-step motion planning for free-climbing robots," in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2004.

[6] T. Bretl, "Multi-step motion planning: Application to free-climbing robots," *PhD Thesis, Stanford University*, Stanford, CA, June 2005.

[7] H. lida, H. Hozumi, and R. Nakayama, "Development of Ladder Climbing Robot LCR-1," *Journal of Robotics and Mechatronics*, vol. 1, no. 4, pp. 311–316, 1989.

[8] H. Yoneda, K. Sekiyama, Y. Hasegawa, and T. Fukuda, "Vertical ladder climbing motion with posture control for multi-locomotion robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, sept. 2008, pp. 3579–3584.

[9] J. J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, "Dynamically-stable motion planning for humanoid robots," *Autonomous Robots*, vol. 12, pp. 105–118, 2002.

[10] S. Kagami, K. Nishiwaki, J. Kuffner, K. Okada, M. Inaba, and H. Inoue, "Vision-based 2.5D terrain modeling for humanoid locomotion," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 2, Sept. 2003, pp. 2141–2146.

[11] S. M. LaValle and J. Kuffner, "Randomized kinodynamic planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 1, 1999.

[12] K. Hauser, T. Bretl, and J.-C. Latombe, "Non-gaited humanoid locomotion planning," in *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, Dec. 2005, pp. 7–12.

[13] K. Hauser, T. Bretl, K. Harada, and J.-C. Latombe, "Using motion primitives in probabilistic sample-based planning for humanoid robots," in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2006.

[14] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada, and B. Wilcox, "Motion planning for legged robots on varied terrain," *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1325–1349, December 2008.

[15] K. Hauser, "Motion planning for legged and humanoid robots," *PhD Thesis, Stanford University*, Stanford, CA, September 2008.

[16] R. Diankov and J. Kuffner, "OpenRAVE: A Planning Architecture for Autonomous Robotics," Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-08-34, July 2008.

[17] T. Bretl, "Motion Planning of Multi-Limbed Robots Subject to Equilibrium Constraints: The Free-Climbing Robot Problem," *International Journal of Robotics Research*, no. 4, pp. 317–342, Apr. 2006.