Planner-aided Design of Ladder Climbing Capabilities for a DARPA Robotics Challenge Humanoid

Yajia Zhang Jingru Luo Kris Hauser

***Abstract*— This paper describes preliminary steps toward providing humanoid robots with ladder climbing capabilities. Although seemingly straightforward for humans, ladder climbing 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 the Hubo-II+ humanoid that automatically optimizes motion primitives offline and adapts them on-line to novel ladders. The adapted motions are guaranteed to satisfy contact, collision, quasi-static balance, and torque limit constraints. This method is used to automatically test a number of strategies, including forward and backward climbing, on a variety of ladders in simulation. This planner-aided design paradigm has proven useful for hardware design because it permits the rapid testing, verification, and optimization of hardware changes by automatically generating novel climbing strategies.**

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. 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. Ladder climbing is an essential mode of locomotion for navigating industrial environments and performing maintenance tasks in buildings, trees, and other man-made structures (e.g., utility poles).

This paper describes preliminary steps toward tackling the ladder-climbing problem as posed by the DARPA Robotics Challenge (DRC), which aims to develop robots with human-like navigation, manipulation, and perception capabilities in a hazardous disaster site. The type and location of the ladder are unspecified, so the robot must sense, plan, and execute a climbing motion largely autonomously. Beginning with the established Hubo-II+ humanoid platform (Fig. 1), preliminary tests indicate that the stock Hubo is physically unable to climb ladders in a standard human-like fashion. Relative to a human, Hubo-II+ has weaker grip strength, is shorter (130 cm), has shorter arms, has fewer degrees of freedom in each arm (6 instead of 7DOF), and less flexible legs (e.g., upper limit of flexion is 90° rather than 130° and higher). This illustrates the principle that *human strategies usually fail when directly applied to humanoids*; seemingly subtle differences in numbers of joints, joint ranges, limb lengths, and actuator strengths can cause dramatic changes in locomotion capabilities. As a result, we are faced with a *simultaneous hardware and motion design* problem. Overengineering the robot to meet or exceed human capabilities is unrealistic, and furthermore individual design decisions are highly coupled, i.e., stronger grippers require heavier motors that mat affect balance or manipulation capabilities. As a result, we ask the question, *how little can we change the robot so that it can climb a large variety of ladders?*

Yajia Zhang, Jingru Luo, and Kris Hauser are with the Department of Computer Science and Informatics, Indiana University Bloomington, USA {zhangyaj,luojing,hauserk}@indiana.edu

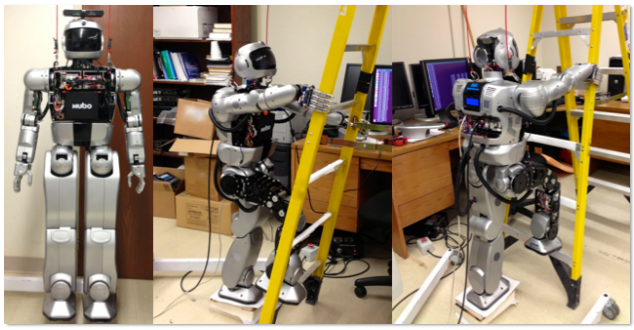


Fig. 1. A Hubo-II+ robot and an experimental setup for the ladder-climbing task.

We employ a *planner-aided design* paradigm to recommend selective upgrades to Hubo hardware by extensively optimizing and simulating motions over a wide range of ladders. The core component of this system is a multi-limb motion planner 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.
- 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. Our method is applied to a variety of ladders in simulation, and has been used to calculate

optimized grip strengths and joint limit changes that will be included in an upcoming upgrade to the Hubo platform. The planner can also be used online to solve for optimized planning motions for novel ladders sensed by the robot's perception system.

II. Related Work

Since the early work of biped gait control and synthesis by Vukobratovic [10] in 1969 and WABOT developed by Wasada University in 1973 [7], 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 [6].

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 [8]. Bretl et al. studied multi-step motion planning for a vertical rock climbing robot [2]. Iida et al. developed a ladder-climbing robot called LCR-1 for vertical ladder climbing [9]. 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 [11]. Unlike general-purpose humanoids, these climbing robots are special-purpose, and as a result have limited or nonexistent capabilities to perform manipulation, stair-climbing, and transitioning from flat ground to ladders and vice versa. Ladder motion planning with general-purpose humanoids has been achieved in simulation by Hauser et al. [5] and Bouyarmane et al [1] with multi-limb contact planning. However, the problem of autonomously sensing, planning, and climbing a novel ladder, particularly with realistic actuator limits, is still an open problem.

III. Hubo-II+ Robot

Hubo-II+ is a full-size humanoid robot designed and built by Rainbow Co., a spin-off company from the Korean Advanced Institute of Science and Technology (KAIST). It 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 F/T sensors, accelerometers, and gyros for sensor feedback. Its five fingers are individually actuated and cable-driven. The gripping force is relatively weak compared to a human.

IV. Planner-Aided Design

The planner-aided design paradigm, depicted in Fig. 2, employs the use of a motion planner to test whether a robot can successfully solve problem instances (use cases) drawn from the space of problems typically expected in practice. The designer inspects the results of these tests and generates design variations to be fed back to the planner. In Sec. VII of this paper we explore the idea of automatically testing design variations in order to minimize some notion of cost.

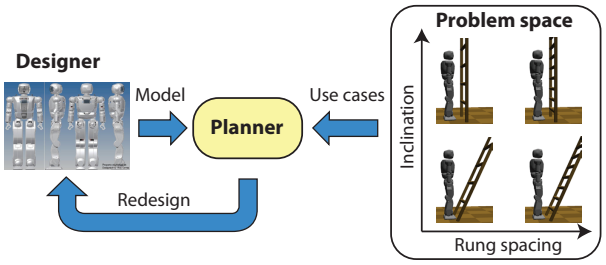


Fig. 2. Workflow of the planner-aided design paradigm.

A. Environment Modeling

Although our current work assumes that a precise ladder model is given, during the DRC we will use stereo and depth sensing to build a model on the fly. We are planning to use the robot's head-mounted sensors to build a 3D point cloud of the ladder and measure its dimensions via a 3D model-based template fitting procedure. Some of the template ladder parameters are depicted in Fig. 3(a). 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 contact selection and collision testing during planning. The planner also supports collision avoidance with other obstacles in the environment that may be perceived by the robot's sensors.

Friction of the hands and feet against the ladder are assumed to follow a Coulomb friction model, with a known coefficient of friction (0.4 is used in our examples). In future work, an initial tactile sensing step may help the robot identify the friction coefficients before starting the climb.

B. Motion Constraints

Keeping the robot's fingers fixed, the robot's configuration space $\mathcal{C}$ is 33-D, including 6 "virtual" DOFs for the rigid transform of the robot's base. 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)).

The feasible space at σ is the set of feasible configurations $F_\sigma = \{q \mid q \text{ is feasible at } \sigma\}$. A feasible configuration q

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. These are represented by an implicit function $C_\sigma(q) = 0$, which restrict motion to a lower-dimensional submanifold of $\mathcal{C}$. During sampling, contact constraints are solved using numerical inverse kinematics techniques.
2) Joint limits: $q_{min} \leq q \leq q_{max}$.
3) Free of collision with the environment and self-collision. We assume the robot's link geometry and environment are represented as triangulated meshes and use the PQP bounding volume hierarchy package [3] to test for collision.
4) Frictional stability 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 [2].
5) Quasistatic equilibrium of internal and external forces, respecting friction, torque, and grip force limits.

More specifically, quasistatic equilibrium requires gravity to be balanced against contact forces $f_1, \ldots, f_k$ and joint torques τ:

$$G(q) = \tau + \sum_{i=1}^{k} J_{p_i}(q)^T f_i \qquad (1)$$

where G is the generalized gravity torque and J_{p_i} is the Jacobian of the i'th contact point. The torque limits must satisfy limits $|\tau| \leq \tau_{max}$, with the inequalities taken elementwise. The 6 components of τ_{max} corresponding to the virtual base DOFs are set to zero. Each contact force must lie in its respective friction cone: $f_i \in FC_i$, which are approximated using polyhedra. Feasible torques and forces $(\tau, f_1, \ldots, f_k)$ are solved via a linear program (LP). The LP is feasible iff the equilibrium constraint is satisfied at q.

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 $p_1, \ldots, p_n$, with $p_i : [0, 1] \to F_{\sigma_i}$ for each $i = 1, \ldots, n$. 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.

V. Ladder-Climbing Motion Planner

We designed and developed a motion planner for ladder-climbing motions based on the idea of *motion primitives* [4], 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 solution path. Besides providing feedback to the robot designer, the plan can also be executed either in simulation or on the physical robot.

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.

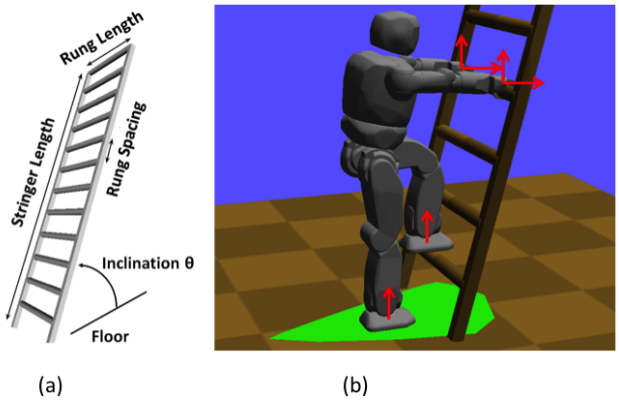


Fig. 3. (a) Parameter specification of a 3D ladder model. (b) Point-contacts and their normals (red arrow). A support polygon (green region) can be calculated based on the contacts to check the stability of the robot.

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.

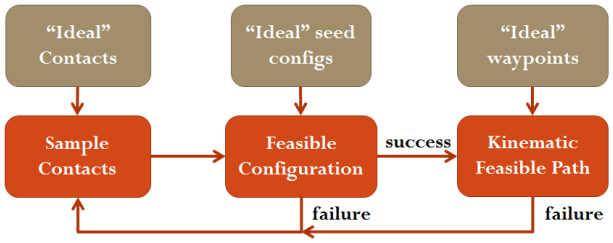


Fig. 4. Three steps in motion planning for ladder climbing based on motion primitives. If one step fails, the planning process will trace back to the previous step. Prior information is being used to assist in each step.

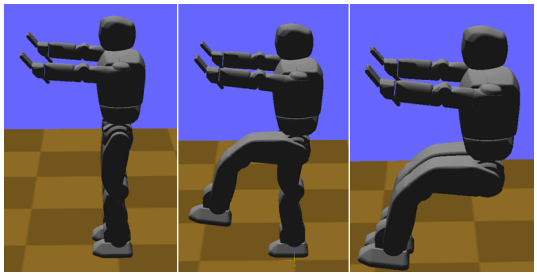


Fig. 5. Seed examples. From left to right: placeHands, liftLFoot, liftRFoot.

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 [5]. 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 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.

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

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
```

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.

VI. Computer Simulation Results

Extensive computer simulations were performed to test our ladder-climbing motion planner with Hubo-II+ robot model and a list of various ladders 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).

Preliminary observations found that two parameters of Hubo-II+ robot play an important role in the feasibility of ladder climbing: 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. The size of the lower leg geometry 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 lower-leg geometry by 1.5 cm. In practice, this can be done by shrinking or removing the leg's plastic covers).

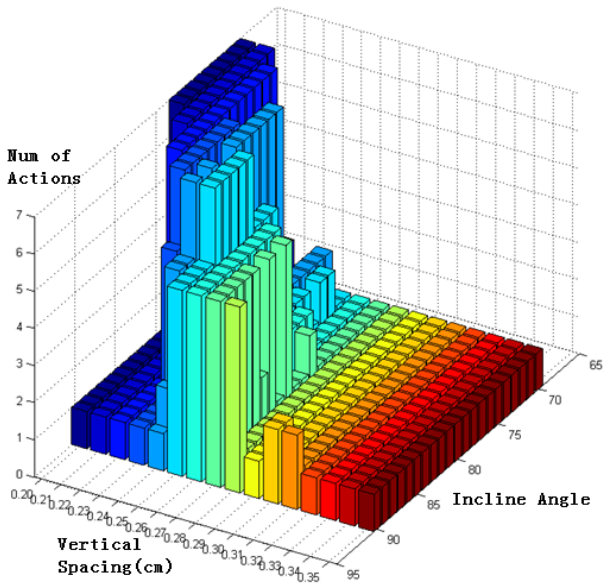


Fig. 6. Simulation results on ladders with inclined angle between 70° and 90°, and the rung spacing between 20 cm and 35cm. The height of the bar indicates how many sequential primitive actions were successfully planned.

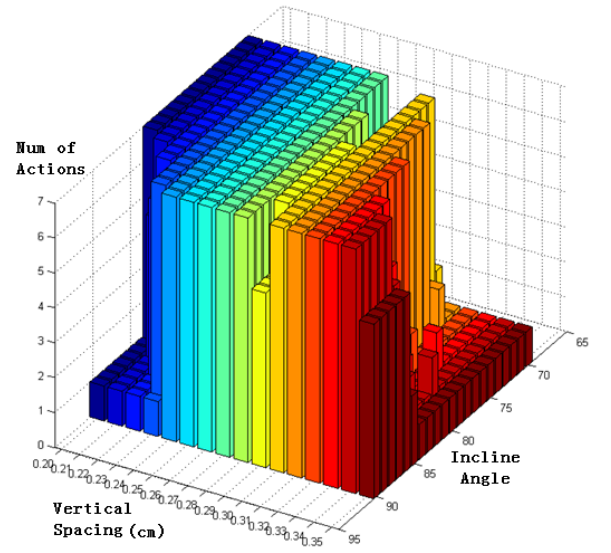


Fig. 7. Simulation results indicate that the suggested modifications to Hubo-II+ boosts solution rate to 70.24%.

3) Apply both of the above changes.

In these experiments other robot and ladder settings remained the same. The results show that by increasing the leg-pitch limit alone, the fully solved ladder-climbing ratio increased to 43.45%; by shrinking the knee joints by 1.5 cm alone, the ratio increased to 34.82%; by applying both modifications, the ratio increased to 70.24% (see Fig. 7).

VII. Automatic Planner-Aided Cost Minimization

Next we consider the problem of *automatically* optimizing design parameters for a planning problem, which reduces the burden on the designer to inspect planning results and

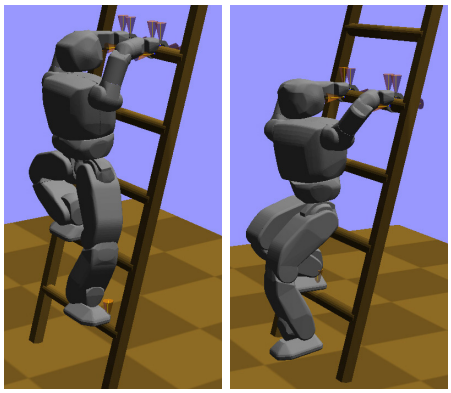


Fig. 8. The forward and backward climbing strategies compared in this paper can successfully climb a standard ladder from the point of view of kinematic constraints. However, strong grip forces are required in these two configurations.

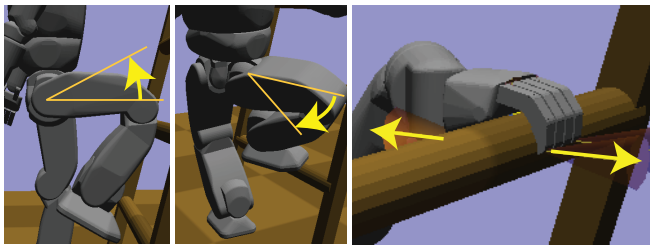


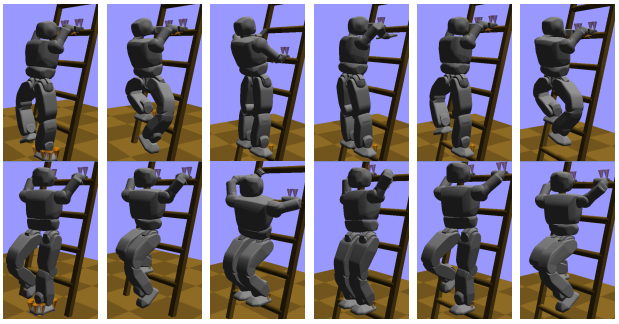
Fig. 9. The three types of constraint displacement considered: hip pitch limit, hip yaw limits, and finger/thumb force limits.

propose candidate design modifications. Note that the problem is not as straightforward as planning a path and then determining the requirements for executing the path; alternate paths may lead to cheaper requirements. Moreover, the system must also consider how multiple design parameters trade off in terms of cost. To do so, we present a new Minimum Constraint Displacement (MCD) motion planning formulation where the design parameters affect how motion constraints are specified. The objective is to optimize a cost function $C(d)$ over design parameters d *and* a path $q(s)$ such that the path satisfies all motion constraints.

We consider three candidate constraint displacements that would require a tractable amount of mechanical engineering effort to change (Fig. 9):

- Increasing hip pitch limits (1 parameter each for L/R). Cost is weighted by 1/5°.
- Increasing hip yaw limits (1 parameter each for L/R). Cost is weighted by 1/5°.
- Increasing finger and thumb strength (2 parameters each for L/R). Cost is weighted by $1\,\mathrm{kg}^{-1}$.

These adjustments correspond to *design parameters* $d_1, \ldots, d_m$ with $m = 8$, which define certain displacements to a stance's constraints. Now, the joint limit and static equilibrium constraints now depends on one the parameter vector $d = (d_1, \ldots, d_m)$, so that $F_\sigma(d)$ is a family of feasible



Strat.	Method	1	2	3	4	5	Max
FW	RRT	0	7.2	2.70	**0.98**	**0.23**	7.2
FW	MCD	0	**2.22**	2.50	0.99	**0.23**	2.50
BW	RRT	0	3.16	**2.37**	1.48	0.51	3.16
BW	MCD	0	2.37	**2.37**	0.15	0.45	**2.37**

Fig. 10. Snapshots along the two climbing strategies and grip forces of each stage of the climb, in kg.

spaces. Specifically, the joint limits become

$$q_{min}(d) \leq q \leq q_{min}(d) \tag{2}$$

and the quasistatic equilibrium condition tests whether there exists a solution τ, f to

$$\left| G(q) - \sum_{i=1}^{k} J_{p_i}(q)^T f_i \right| \leq \tau_{max}(d). \tag{3}$$

The MCD motion planning problem is now:

$$\min_{d \geq 0, q} C(d) \text{ such that } q(s) \in F_\sigma(d) \text{ for all } s \in [0,1] \tag{4}$$

The cost function is simply a weighted sum of displacement values. We use a sample-based approach to simultaneously optimize the path $q(s)$ and design parameters d. (Further details of our algorithm are currently under blind review in another publication.)

Along a ladder climbing path, we solve successive MCD problems as follows:

1) Using the planner defined in Sec. V generate stances $\sigma_1, \ldots, \sigma_m$ and *transition configurations* $q_1, \ldots, q_{m-1}$ where q_i satisfies the constraints of both σ_{i-1} and σ_i.
2) Solve an MCD problem between each pair of configurations q_{i-1}, q_i among the configuration space $\mathcal{F}_{\sigma_i}$ and up to 8 displacement parameters.
3) Take the maxima of each optimized displacement.

Our experiments are performed on a 75° inclined ladder with cylindrical rungs spaced 30 cm apart, on which standard forward climbing is unsuccessful because the knees of the robot collide with the rungs. We designed two kinematically-feasible strategies: a forward-facing strategy with splayed feet (FW), and a backward climbing strategy (BW) (Fig 10). Both strategies climb one rung at a time without skipping, and run over 12 stances. We apply MCD to each stance. Running time is between 10 s and 3 min for each stance, and is dominated by collision and equilibrium checking. We also compare a standard sampling-based planner (RRT), with grip force and hip joint displacements set to hypothetical upper limits (10kg and 20° respectively).

Results indicate that grip strength is the bottleneck in climbing, and no other joint limit displacements are needed. Fig. 10 gives maximum grip forces for each strategy during each three-contact stage (we do not list four-contact stages because they exert lower forces on the hands). The RRT solutions exhibit large grip forces, particularly in stage 2. Using MCD, we can conclude with relative certainty that both the FW and BW strategies are actually limited during the one-handed support stage 3, with approximately 2.5kg grip force sufficient to carry out the optimized motion.

VIII. Summary and Conclusion

A planner-aided design approach was developed to study and improve the feasibility of ladder climbing for the Hubo-II+ robot across a variety of ladders. The planner is able to generate paths that satisfy contact, collision, balance, and torque limit constraints, and was used to test the effects of how joint limits of the leg pitch and the geometric size of the knee joints affected the range of ladders successfully climbable. A new technique for simultaneously optimizing costs of design parameter choices and feasible motions was applied to determine minimum grip force requirements for an upgraded Hubo. The new version of Hubo is currently under construction.

References

[1] K. Bouyarmane, J. Vaillant, F. Keith, and A. Kheddar. Exploring humanoid robot locomotion capabilities in virtual disaster response scenarios. In *Proc. 2012 IEEE-RAS Int. Conf. on Humanoid Robots*, pages 337–342, Osaka, Japan, Dec. 2012.

[2] T. Bretl. Motion Planning of Multi-Limbed Robots Subject to Equilibrium Constraints: The Free-Climbing Robot Problem. *International Journal of Robotics Research*, 4:317–342, Apr. 2006.

[3] S. Gottschalk, M. Lin, and D. Manocha. OBB-tree: A hierarchical structure for rapid interference detection. In *ACM SIGGRAPH*, pages 171–180, 1996.

[4] 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.

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

[6] 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)*, volume 1, pages 604–609, 2004.

[7] I. Kato. Development of WABOT 1. *Biomechanism*, 2:173–214, 1973.

[8] 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)*, pages 1268–1273, April 2007.

[9] H. Iida, H. Hozumi, and R. Nakayama. Development of Ladder Climbing Robot LCR-1. *Journal of Robotics and Mechatronics*, 1(4):311–316, 1989.

[10] M. Vukobratovic and D. Juricic. Contribution to the synthesis of biped gait. *Biomedical Engineering, IEEE Transactions on*, BME-16(1):1–6, jan. 1969.

[11] 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)*, pages 3579–3584, sept. 2008.