Non-Gaited Humanoid Locomotion Planning

Kris Hauser, Tim Bretl, and Jean-Claude Latombe
Stanford University
Stanford, CA 94307, USA
khauser@cs.stanford.edu, tbretl@stanford.edu, latombe@cs.stanford.edu

***Abstract*— This paper presents a non-gaited motion planner for humanoid robots navigating very uneven and sloped terrain. The planner allows contact with any pre-designated part of the robot's body, since the use of hands or knees (in addition to feet) may be required to balance. It uses a probabilistic, sample-based approach to compute each step. One challenge of this approach is that most randomly sampled configurations do not satisfy all motion constraints (closed-chain, equilibrium, collision). To address this problem, a method of iterative constraint enforcement is presented that samples feasible configurations much more quickly. Example motions planned for the humanoid robot HRP-2 are shown in simulation.**

***Index Terms*— Humanoid robots, motion planning, non-gaited locomotion, multi-step planning, equilibrium.**

I. Introduction

A key constraint on the motion of a legged robot—in particular a humanoid robot—is maintenance of equilibrium. At each instant, the reaction forces at contacts with the environment must exactly compensate for the other forces acting on the robot. In flat horizontal terrain, a gaited motion can be pre-computed to satisfy this constraint. However, in highly irregular terrain (e.g., rocky outdoor terrain, broken urban environment after an earthquake, lunar surface), the contacts that a robot can possibly make are arbitrarily distributed, and sometimes sparse. As a result, each motion step may be unique and its planning requires deliberate reasoning about contacts, equilibrium, and collision.

In this paper, we present a motion planner for humanoid robots navigating rigid, severely uneven, possibly sloped terrain. In such terrain, foot contacts do not always suffice to achieve equilibrium. So, our planner can make use of any pre-designated part of the robot's body (e.g., feet, hands, knees) to achieve balance. We have successfully tested our planner on a computer model of the humanoid robot HRP-2 of AIST [1] operating in various virtual environments. Though still preliminary, our results demonstrate the ability of our planner to compute complex motions in difficult terrain.

We assume the motion of a humanoid robot consists of a sequence of steps, each of which maintains a fixed set of contacts with the environment. A transition from one step to the next either breaks a contact or makes a new one. As in dexterous manipulation [2], planning the motion of a humanoid requires computing both a sequence of contacts and trajectories to achieve them.

One way to make planning faster is to solve these two sub-problems separately. For example, if the terrain is nearly flat, then it does not matter much where a humanoid robot places its feet. So, it is appropriate first to worry about the robot's overall motion (e.g., by modeling the robot as a vertical cylinder sliding among obstacles) and then to follow this motion with a regular gait. Alternatively, if the terrain is very rough, useful contacts are sparse and arbitrarily distributed. So, it makes more sense to choose the contacts before computing the trajectories [3].

The second approach ("contact-before-motion") is the one we use here. We begin by sampling a number of useful *contacts*—associations between points on the robot and points on the terrain. Each set of contacts forms a potential *stance*, at which the robot can take a step. A distinct *feasible space* corresponds to each stance: it is the set of robot configurations at which the stance contacts are achieved, the equilibrium constraint is satisfied[1], and the robot is collision-free (except at the intended contacts). We search a *stance-adjacency graph* for a sequence of stances to reach a goal. Two nodes (stances) in this graph are connected if they differ by a single contact and contain a feasible configuration in common. We convert a sequence of stances into a continuous motion by planning a one-step trajectory in the feasible space of each stance. Upon failure, graph search resumes.

Overall, our work extends this contact-before-motion planning approach, which was previously developed for free-climbing robots [4], [5], to humanoid robots with more degrees of freedom. We also make two additional contributions:

- Our planner allows the robot to make contact with any pre-designated point on its body. So, it allows the robot to walk on relatively easy terrain, use its hands as needed in severely broken and steep terrain, and crawl when obstacles prevent the robot to move standing up.
- Our planner, which embeds a Probabilistic Roadmap (PRM) technique [6] to compute one-step trajectories, employs numerical techniques to sample the feasible space at each stance more quickly by reducing the sampling rejection rate.

This paper gives an overview of our planner, focusing primarily on the second contribution listed above. It is organized in two parts. The first part discusses our adaptation of the planner of [3] to humanoid motion planning. The second part describes our sampling method. We demonstrate in simulation the ability of our planner to generate motions for HRP-2 in environments that require the use of hands for mobility (in particular, rough terrain, high ledges, and ladders).

[1] In this paper, we only consider quasi-static equilibrium.

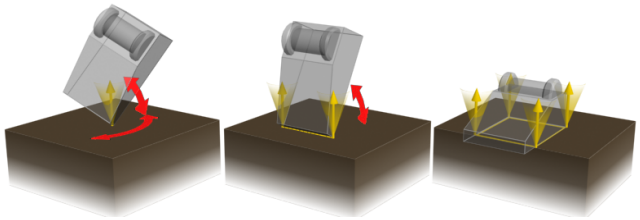

Fig. 1. Three types of contact, showing contact normals and friction cones.

II. Humanoid Locomotion Planning

A basic method for navigating around obstacles on flat ground approximates the humanoid as a cylinder (or other bounding volume), plans a 2D collision-free path of this cylinder, and follows the path with a fixed gait [7], [8]. This method can generate motions very quickly, since it takes advantage of the substantial body of research on both 2D path planning and on creating stable, dynamic walking gaits for humanoid robots.

Other approaches have been developed for piecewise-flat terrain (e.g., with stairs), where the robot must step over, onto, or under obstacles. These methods consider each step individually. One method searches a grid-based terrain decomposition using bounds on step length and height, placing individual footsteps with kinematic planning of the lower body [9]. Another method uses a library of feasible steps [10], [11].

Our work addresses humanoid locomotion in very uneven or sloped terrain. It is an extension of previous work on motion planning for a four-limbed free-climbing robot [3]. This robot climbed an inclined, planar wall by making frictional point contacts at scattered "climbing holds." Because the location of each hold was arbitrary, the climbing motion was non-gaited, and each "step" was often unique. Therefore, a PRM planner was used to compute each step. We extend this planner to humanoid robots with an enhanced model of contact (Section III) and an improved method of sampling (Section VI).

III. Contact Modeling

We assume that all robot links and the environment are perfectly rigid. When a point p^R on the surface of a robot link touches a point p^E on the surface of the environment, we call this association a *point contact*. We only consider points p^E where there is a unique plane tangent to the environment surface. The point contact is then characterized by the outer normal to this plane (pointing away from the environment) and a coefficient of friction μ. The reaction forces that the environment can generate span a friction cone of half-angle $\tan^{-1}\mu$, whose apex is $p^R = p^E$ and whose axis is oriented along the outer normal. (In fact, in the rest of this paper, it is only necessary that the range of possible reaction forces be a convex set.) A point contact allows the robot link to rotate freely about the fixed p^R.

We model other types of contact between the robot and the environment by groups of $k > 1$ point contacts $(p_i^R, p_i^E), i = 1, ..., k$, occurring simultaneously, where all points p_i^R belong to the same robot link. When $k = 2$, we have an *edge contact* that allows the robot link to rotate freely about the fixed line passing through p_1^R and p_2^R. When $k > 2$, we have a *face contact* that fully constraints the link's orientation. By convention, each point p_i^R in a face contact is located at a vertex of the convex hull of the overlap between the robot link and the environment. Each type of contact is illustrated in Figure 1.

In advance, we designate a set of features (points, edges, and faces) on the robot at which contact is allowed. There is no restriction on this set, but once it has been chosen, no other points on the robot's surface may touch the environment. A contact is generated by picking a feature and placing it against a point in the environment. This process is repeated many times to construct a set of candidate contacts.

Ideally, the set of candidate contacts should be large enough to allow the robot to reach a goal, but not too large to overwhelm the planner. Hence, picking the "right" contacts is critical. In our current implementation, we do this either at random or by hand. We are currently investigating better techniques, based either on local properties such as the range of reaction forces at a contact or on other considerations such as the distribution of contacts.

IV. Static Equilibrium Test

A configuration q of the humanoid robot is specified by the translation and orientation of an arbitrarily selected root link and a set of joint angles. This parameterization is 36-dimensional for the HRP-2 robot.

Consider a configuration q where the robot makes k point contacts with the environment. Three kinds of forces act on the robot: joint torques τ, reaction forces $f_i (i = 1, ..., k)$ at contact points p_i, and gravity g. The robot is in static equilibrium if these forces sum to zero, the joint torques are within their limits τ_{max}, and the contact forces f_i lie in their respective friction cones $\mathcal{FC}_i$. This yields the equilibrium conditions

$$\begin{aligned} &|\tau| \leq \tau_{max} \\ &G(q) = \tau + \sum_i J_i^T(q) f_i \\ &f_i \in \mathcal{FC}_i \text{ for all } i \end{aligned} \tag{1}$$

where $G(q)$ is the generalized gravity vector, and J_i is the contact point Jacobian.

For a given configuration q, conditions (1) can conservatively approximated as a linear program (LP) by inscribing a polyhedral pyramid inside each friction cone. If the LP is infeasible, the configuration q is not in equilibrium. If any feasible solution $(\tau, f_1, ..., f_k)$ can be found then these torques and contact forces keep the robot in equilibrium at q.

A simpler test can be derived by assuming the robot to be a rigid body, or equivalently assuming infinite torque limits. Denoting the robot's center of mass by $CM(q)$ and the gravity

force by mg, we get the rigid-body equilibrium conditions [4]:

$$\begin{aligned} &\sum_i f_i + mg = 0 \\ &\sum_i p_i \times f_i + CM(q) \times mg = 0 \\ &f_i \in \mathcal{FC}_i \text{ for all } i \end{aligned} \tag{2}$$

These conditions are necessary (but not sufficient) for the robot to be in equilibrium at q. In [4] a method is presented that transforms them into a convex *support polygon* in the horizontal plane over which $CM(q)$ must lie. This polygon is fully determined by the contacts. For sufficiently actuated robots, torque limits are rarely violated. Then checking if $CM(q)$ lies over the support polygon can quickly reject many configurations before testing the more expensive conditions (1). In Section VI, the support polygon will be used to sample feasible configurations with higher success rate.

V. Overview of Planner

The structure of our planner is very similar to the one described in [3]. It consists of a multi-step and a single-step planner.

The multi-step planner is given a start and a goal configuration of the robot, as well a set CONTACT of candidate contacts. We define a *stance* σ to be any set of contacts from CONTACT such that no point on either the robot or terrain participates in more than one contact. The feasible space $\mathcal{F}_\sigma$ of the robot is the set of all configurations q where the robot achieves all contacts in σ and is in static equilibrium and collision-free . Two stances σ_1 and σ_2 are *adjacent* if $\sigma_1 \subset \sigma_2$ and if $\sigma_2 \backslash \sigma_1$ contains a single contact from CONTACT. Hence, a motion that switches from σ_1 to σ_2 reaches a new contact, while a motion that switches from σ_2 to σ_1 breaks an existing contact. This definition can be generalized to allow motions that switch from a face contact to an edge contact on the face's boundary (e.g., moving from contact with the entire foot to just the toes), or similarly move from an edge to a point.

Let σ_{start} and σ_{goal} be the stances at the start and goal configurations, respectively. The multi-step planner must first create a sequence of adjacent stances $(\sigma_0 = \sigma_{start}, \sigma_1, ..., \sigma_n = \sigma_{goal})$ likely to contain a feasible motion. It does so by searching a *stance graph* Γ, whose nodes are stances, and two adjacent stances σ_1 and σ_2 are connected with an edge if $\mathcal{F}_{\sigma_1} \cap \mathcal{F}_{\sigma_2}$ is non-empty. The stance graph is defined in this way because experiments show that for free-climbing robots [3], if $\mathcal{F}_{\sigma_{i-1}} \cap \mathcal{F}_{\sigma_i}$ and $\mathcal{F}_{\sigma_i} \cap \mathcal{F}_{\sigma_{i+1}}$ are non-empty, then most of the time, a feasible path exists between them. This is true for humanoid robots as well. Therefore, a sequence of stances in Γ is likely to admit a feasible motion.

The stance graph is constructed incrementally by expanding adjacent stances σ_{i+1} only if it can be shown that $\mathcal{F}_{\sigma_i} \cap \mathcal{F}_{\sigma_{i+1}}$ is non-empty. This is done by attempting to sample a *transition* configuration $q_i \in \mathcal{F}_{\sigma_i} \cap \mathcal{F}_{\sigma_{i+1}}$. At q_i, a feasible motion can change stance from σ_i to σ_{i+1} or vice versa. Despite the fact that most infeasible stances can be quickly recognized (e.g. contacts are too far apart) and discarded, many go undetected and must be sampled. Therefore, a critical factor in the speed of multi-step planning is discerning feasible steps from infeasible ones by sampling feasible transition configurations as quickly as possible. We present our approach in the next section.

After the multi-step search produces a sequence of stances, it tries to transform this sequence into a continuous feasible path. In each $\mathcal{F}_{\sigma_i}, i = 0, ..., n$, the single-step planner is called to connect the transition configurations q_{i-1} and q_i (where q_{-1} is the start configuration and q_n is the goal). Because feasible spaces are too complex to represent exactly, while all feasibility constraints can be efficiently tested at any configuration, our one-step planner is a sampling-based PRM planner (see Section VII). Upon failure of the one-step planner to transform the sequence of stances into a feasible motion, the multi-step planner is resumed to produce another candidate sequence of stances.

VI. Sampling Transition Configurations

Our planner should efficiently sample a transition configuration in the intersection of two feasible spaces $\mathcal{F}_\sigma$ and $\mathcal{F}_{\sigma'}$ at σ and σ'. This operation is challenging for two reasons:

- First, due to the contact constraints, each feasible space has fewer dimensions than the robot's configuration space $\mathcal{C}$, hence has zero measure in $\mathcal{C}$. So, pure rejection sampling, which consists of sampling $\mathcal{C}$ and then testing if the generated sample lies in $\mathcal{F}_\sigma \cap \mathcal{F}_{\sigma'}$, does not work.
- The equilibrium and collision constraints reduce the size of $\mathcal{F}_\sigma \cap \mathcal{F}_{\sigma'}$ (but not its dimensionality). In our tests, it is often the case that less than 1% of the configurations satisfying the contact constraints also satisfy the equilibrium and collision constraints.

Below we will compare two sampling methods that differ in their sampling philosophy. The first method (direct parameterization) directly generates samples to satisfy the contact constraints, but allows samples to be rejected by equilibrium and collision constraints. The second (iterative constraint enforcement, or ICE) uses an iterative, numerical technique that spends more time per sample in an attempt to reduce the rejection rate. Our experiments show that ICE samples transition configurations faster.

A. Direct Submanifold Parameterization

Assume without loss of generality that $\sigma' \subset \sigma$. For the robot to be at a configuration in $\mathcal{F}_\sigma \cap \mathcal{F}_{\sigma'}$, it must achieve all the contacts in the larger stance σ, which imposes a number of closed-loop kinematic constraints. The set of all configurations satisfying the contact constraints forms a lower dimensional submanifold of $\mathcal{C}$.

A *direct parameterization* allows explicit sampling of the submanifold where the contact constraints are satisfied [12]. The closed chain is divided into "floating" and "trailing" joints. First, the floating joint angles are sampled at random. Then, the trailing joint angles are determined using analytic non-redundant inverse kinematics (IK). So, the submanifold is directly parameterized by the floating joint angles.

A sample may fail if there is no inverse kinematics solution for the trailing joints. For this reason, the random loop generator (RLG) technique proposed in [12] incrementally samples each floating joint angle along the chain in such a way that the rest of the chain meets a simple necessary condition to close the loop. This condition is that the endpoint of the sub-chain whose configuration has been sampled lies within a spherical approximation of the workspace of the rest of the chain. But this approximation works poorly when joints have tight limits (especially for the trailing chain). Our experiments with the HRP-2 robot indicate that it often results in low success rate.

B. Iterative Constraint Enforcement

RLG generates few samples that satisfy equilibrium and collision constraints. Furthermore, analytical IK solutions for position and orientation constraints are often not available, (e.g. any link on the HRP-2 except the feet). These shortcomings led us to develop a sampling approach that handles general robot kinematics and also reduces the rejection rate.

Our approach is much like numerical IK, a second common method for sampling closed chain kinematics [13], which uses iterative numerical root-solving methods to move an initial configuration into the submanifold of $\mathcal{C}$ where the contact constraints are satisfied. Numerical IK can be viewed as iterative enforcement of contact constraints. Our ICE method uses the same framework to iteratively enforce equilibrium and collision constraints as well.

1) Contact Enforcement: We use the following approach to enforce contact constraints. Define an error function $C_\sigma(q)$ that is zero when all contacts are achieved. Starting from some initial configuration, we use a Newton-Raphson method to solve $C_\sigma(q) = 0$. At each cycle this method calculates the pseudo-inverse of the Jacobian matrix of the robot contact points and takes a step in configuration space to reduce the constraint residual. This repeats until the residual is below some tolerance or cannot be reduced further. Tests show that major speedups are obtained by choosing initial configurations that do not violate too much $C_\sigma(q) = 0$. To do this, we use the RLG technique to pick floating joint angles at random and the Cyclic Coordinate Descent [14] technique to pick values of the trailing joint angles that almost close the loop.

2) Equilibrium Enforcement: Since $\sigma' \subset \sigma$, the robot must break a contact when it switches from σ to σ'. So, the equilibrium constraint must be satisfied for the smaller stance σ'. On difficult terrain this constraint can be very limiting. Often, only a tiny fraction of configurations satisfying the contact constraints also satisfy the equilibrium test.

To enforce samples to directly satisfy the equilibrium constraint, we use the support polygon derived from (2). We sample a point $p = (p_x, p_y)$ in its interior and we enforce the CM of the robot at the sampled configuration to lie in the vertical line passing through p. To do this we simultaneously solve for $C_\sigma(q) = 0$ and the constraints $CM_x(q) = p_x$ and $CM_y(q) = p_y$ using the same Newton-Raphson technique. As above, this computes the pseudo-inverse of a Jacobian matrix, but here the Jacobian includes both the contact points and the projection of the CM on the horizontal plane.

TABLE I

SAMPLING RESULTS FOR DOUBLE-LEGGED STANCE ON FLAT GROUND

	Direct Parm.	ICE	
Method	RLG	Numerical IK	All
% Successfully Solved	5.4	89	26
% Pass Equilibrium	0.02	0.9	—
% Pass Collision	0.02	0.4	—
Time / sample (ms)	0.83	9.1	69
Time / feasible sample (s)	4.2	2.3	0.27

3) Collision Retraction: Collisions with the environment and self-collisions are detected using the PQP package [15]. To eliminate unnecessary self-collision checks, we pre-compute all possibly colliding pairs of links.

In highly constraining environments, a large fraction of samples are in collision. To avoid rejecting too many sampled configurations, we retract colliding configurations out of collision. This approach has been previously used in several PRM planners to increase the number of configurations sampled in difficult areas of the configuration space (e.g. narrow corridors) [16], [17]. We incorporate it into ICE as follows. After generating a sample, we use PQP to detect if there is a collision. If so, we estimate the deepest penetrating point and normal using a method similar to [18] and we construct a function $d(q)$ that approximates the penetration distance as a function of q by assuming the point and normal to be constant. On the next iteration, we solve for $C_\sigma(q) = 0$, $CM_x(q) = p_x$, $CM_y(q) = p_y$, and $d(q) = \epsilon$ (where ϵ is a small separation distance) by extending the Jacobian matrix to the deepest penetrating points.

C. Experimental Results

We compare some experimental results of the transition sampling strategies. Table I reports results for the HRP-2 standing on two feet on flat horizontal ground. The support polygon is 10cm x 20cm under the left foot. The direct parameterization of RLG is compared to two ICE methods. "Numerical IK" only enforces contact constraints, using the method described in Sec. VI-B.1. "All" performs full ICE as described above. We generated 10,000 samples with each technique on a 2.8 GHz Pentium IV. A sample is considered solved successfully if it has a valid analytical IK solution (in RLG) or achieves convergence (in the ICE methods). For successfully solved samples, we incrementally tested the equilibrium constraint and the collision constraints to reject infeasible samples, so that feasible samples remain after the collision test. All times include constraint testing. Note that ICE generates samples that are collision free and almost always satisfy equilibrium (no samples in this case violated the torque limits).

RLG generates each sample very quickly, but very few successfully solve IK. Numerical IK takes more time per sample, but has a much higher IK success rate. In both cases, nearly all samples failed the equilibrium test. ICE converges

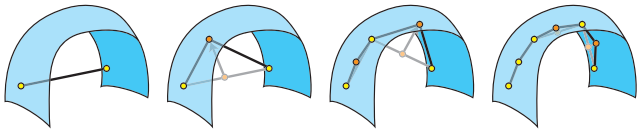

Fig. 2. Local planning on the constraint manifold. Orange configurations are moved into feasible space.

less often than Numerical IK, but almost always converges on a feasible sample, causing it to be 16 times faster than RLG.

There is a further decision of whether to use CM enforcement in the iterative process. The above result shows that enforcing the CM works especially well for small support polygons (about the size of the HRP-2 footprint). In some cases, larger support polygons are encountered (up to 1m in width or even unbounded), in which case enforcing the CM is counterproductive because most configurations will already satisfy equilibrium constraints. Therefore, our planner chooses to enforce a random CM position when the area of the support polygon is small (under a specified threshold), but not when the support polygon is large.

VII. Single-Step Roadmap Construction

To connect two transition configurations, our single-step planner uses a variant of the PRM planner (called SBL) presented in [19]. Like SBL, our planner constructs a roadmap made of two trees of sampled feasible configurations. Each tree is rooted at one of the two transition configurations, and edges connect feasible configurations. At each cycle, the planner expands either tree by first picking a node q at random, then sampling a new feasible configuration q' in a δ-neighborhood of q, and connecting q to q' with an edge. The planner regularly performs a connection step. First, an additional edge is placed bridging the two trees, creating a sequence of feasible configurations from one transition configuration to the other. Then, "local" feasible paths are planned between configurations in this sequence. When the connection succeeds, it returns a complete path between the two transition configurations. When the planner exceeds a specified time limit, it terminates with failure. We make some adjustments to basic SBL for planning with closed-chain constraints.

- Neighborhood sampling: We first sample the neighborhood of q in $\mathcal{C}$. Then, we solve contact constraints using numerical IK as described in Section VI-B.1. We repeat until we get a feasible configuration q'.
- Node connection: Straight-line paths in configuration space are not feasible, so we deform the straight path between two nodes q and q' into feasible space. To do so, we recursively bisect the path segment until its length is below some small threshold (see Fig. 2 for an illustration). We project the midpoint of each path segment into feasible space using the procedure described in Section VI. On failure, the edge from q to q' is removed from the roadmap.

VIII. Example Motions

We show several examples of motions generated using our planner for the HRP-2 robot in very different types of terrain. A constant coefficient of friction is assumed for all examples. Since PRM planners produce paths that randomly explore the robot's entire configuration space, we employed a postprocessing step to make the initially jerky motions look more smoother and more natural.

Fig. 3 depicts a simple 0.5m stair-step that cannot be climbed by the HRP-2 only using footsteps, yet permitting hand contact admits a feasible motion. Six foot contacts were placed in a row, and six right hand finger contacts were placed in various positions and orientations on the ledge. The planner found a 9-stance sequence in about 90s, with roughly 45s spent on stance graph search and one-step planning each.

Fig. 4 depicts frames from a motion starting in a steep-walled depression in the ground, which requires using the hands to climb out and start walking. Frames 2-4 show the robot using its hands for support as it climbs to higher ground. The terrain was generated from a heightfield of fractal noise in a 3m x 3m area, and is depicted as topographical map. The candidate contacts given to the planner were 800 randomly distributed footholds and fingertip contacts. The final motion uses 19 steps. Because of the multitude of available contacts, the planner explored thousands more stances than were required to plan the motion. Despite the fact that the one-step motions were planned in just less than 3 minutes, searching the stance graph took over 3 hours.

Fig. 5 depicts a ladder climbing motion. 45 candidate contacts were manually generated in obvious locations for the feet and hands. The stance graph search is efficient, taking only 7 minutes, primarily because the candidate contacts were well-placed to permit a large number of feasible stances. The limiting factor was one-step motion planning, which took about 3 hours, likely due to the presence of narrow passages in the feasible space caused by rungs of the ladder.

These examples demonstrate the versatility of our planner to handle a variety of terrain, but also expose the fact that the high-quality candidate contact placements are critical for efficient operation.

IX. Conclusion

We take a contact-before-motion approach to non-gaited motion planning for humanoid robots. Our planner first searches for a sequence of steps that make useful contacts with the terrain. This is accomplished by sampling feasible transition configurations between stances. Upon completion, each individual step is planned using a PRM planner. This approach allows us to automatically create motions for rough terrain, using any part of the body for contact.

Our main technical contribution is a strategy for increasing the success rate of sampling transition configurations. A numerical technique iteratively refines samples to satisfy the constraints of the feasible space.

A critical issue in the use of such a planner is starting with a set of useful candidate contacts. Future work should address

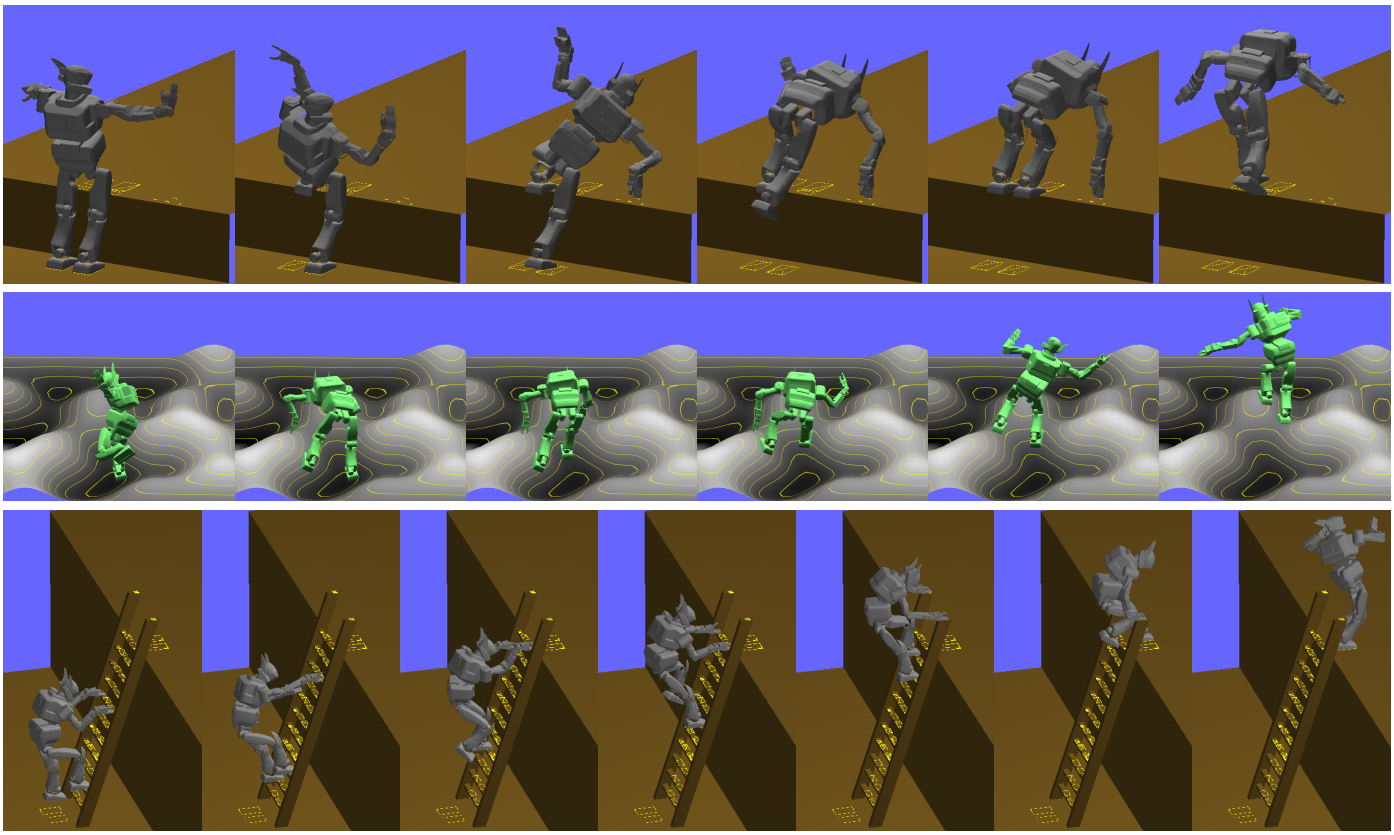

Fig. 3. 0.5m stair-step requiring the use of hands
Fig. 4. Using hands to climb out of a depression in fractal noise terrain
Fig. 5. Climbing a ladder

automatically characterizing and extracting useful contacts from an environment model. Similarly, unnecessary contacts could be avoided during planning with additional work on analyzing contact reachability.

Acknowledgements:

This research was funded by NSF grant 0412884. Kris Hauser is supported by a Stanford Graduate Fellowship. We thank K. Harada and others at AIST Japan for their support.

REFERENCES

[1] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi, "Humanoid robot hrp-2," in *IEEE Int. Conf. Rob. Aut.*, 2004.
[2] A. Okamura, N. Smaby, and M. Cutkosky, "An overview of dexterous manipulation," in *IEEE Int. Conf. Rob. Aut.*, 2000, pp. 255–262.
[3] T. Bretl, "Multi-step motion planning: Application to free-climbing robots," Ph.D. dissertation, Stanford University, Stanford, CA, 2005.
[4] T. Bretl, J. Latombe, and S. Rock, "Toward autonomous free-climbing robots," in *Int. Symp. Rob. Res.*, Siena, Italy, 2003.
[5] T. Bretl, S. Lall, J. Latombe, and S. Rock, "Multi-step motion planning for free-climbing robots," in *WAFR*, 2004.
[6] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in *IEEE Trans. Robot. Automat.*, vol. 12(4), 1996, pp. 566–580.
[7] J. Kuffner, "Autonomous agents for real-time animation," Ph.D. dissertation, Stanford University, Stanford, CA, 1999.
[8] J. Pettré, J.-P. Laumond, and T. Siméon, "A 2-stages locomotion planner for digital actors," in *Eurographics/SIGGRAPH Symposium on Computer Animation*, 2003.
[9] T.-Y. Li, P.-F. Chen, and P.-Z. Huang, "Motion planning for humanoid walking in a layered environment," in *IEEE Int. Conf. Rob. Aut.*, 2003.
[10] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion planning for humanoid robots," in *Int. Symp. Rob. Res.*, Siena, Italy, 2003.
[11] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, "Planning biped navigation strategies in complex environments," in *IEEE Int. Conf. Humanoid Robots*, 2003.
[12] J. Cortés, T. Siméon, and J.-P. Laumond, "A random loop generator for planning the motions of closed kinematic chains using prm methods," in *IEEE Int. Conf. Rob. Aut.*, 2002.
[13] S. M. LaValle, J. H. Yakey, and L. E. Kavraki, "A probabilistic roadmap approach for systems with closed kinematic chains," in *IEEE Int. Conf. Rob. Aut.*, 1999.
[14] L. Wang and C. Chen, "A combined optimization method for solving the inverse kinematics problem of mechanical manipulators," in *IEEE Trans. On Robotics and Applications*, vol. 7, no. 4, 1991, pp. 489–499.
[15] S. Gottschalk, M. C. Lin, and D. Manocha, "Obb-tree: A hierarchical structure for rapid interference detection," in *ACM Siggraph*, 1996.
[16] N. M. Amato, O. B. Bayazit, and L. K. Dale, "Obprm: An obstacle-based prm for 3d workspaces," in *WAFR*, 1998.
[17] M. Saha and J.-C. Latombe, "Finding narrow passages with probabilistic roadmaps: The small step retraction method," in *IEEE/RSJ Int. Conf. Int. Rob. Sys.*, 2005.
[18] B. Heidelberger, M. Teschner, R. Keiser, M. Muller, and M. Gross, "Consistent penetration depth estimation for deformable collision response," in *Vision, Modeling, and Visualization*, Stanford, USA, 2004.
[19] G. Sanchez-Ante and J. Latombe, "A single-query bi-directional probabilistic roadmap planner with lazy collision checking," in *Int. Symp. Rob. Res.*, Lorne, Victoria, Australia, 2001.