Efficient Humanoid Motion Planning on Uneven Terrain Using Paired Forward-Inverse Dynamic Reachability Maps

Yiming Yang, Wolfgang Merkt, Henrique Ferrolho, Vladimir Ivan, and Sethu Vijayakumar

***Abstract*—A key prerequisite for planning manipulation together with locomotion of humanoids in complex environments is to find a valid end-pose with a feasible stance location and a full-body configuration that is balanced and collision-free. Prior work based on the inverse dynamic reachability map assumed that the feet are placed next to each other around the stance location on a horizontal plane, and the success rate was correlated with the coverage density of the sampled space, which in turn is limited by the memory required for storing the map. In this letter, we present a framework that uses a paired forward-inverse dynamic reachability map to exploit a greater modularity of the robot's inherent kinematic structure. The combinatorics of this novel decomposition allows greater coverage in the high-dimensional configuration space while reducing the number of stored samples. This permits drawing samples from a much richer dataset to effectively plan end-poses for both single-handed and bimanual tasks on uneven terrains. This novel method was demonstrated on the 38-DoF NASA Valkyrie humanoid by utilizing and exploiting whole body redundancy for accomplishing manipulation tasks on uneven terrains while avoiding obstacles.**

***Index Terms*—Dynamic reachability map, end-pose planning, humanoid robots, motion planning.**

I. Introduction

HUMANOID robots are designed with human-like morphology for better adaptations in environments designed for people without the need to change the infrastructures. Their high-dimensional kinematic structure offers excellent dexterity but, in turn, its complexity makes the motion synthesis extremely challenging particularly for safe and reactive tasks in close proximity to people. To date, one has had to hand-craft certain prior knowledge, e.g., a stance location and full-body configuration, in order to make planning and operation practical. For instance, to grasp a distant object, the robot needs to walk closer to a pre-grasp stance location first, then plan and execute a grasping motion. In this case, the pre-grasp stance location and the grasping configuration (so-called *end-pose*) are often provided by the operator, limiting significant robot autonomy. Therefore, an efficient algorithm for finding an appropriate and sufficient end-pose is a fundamental problem whenever robots need to explore kinematic redundancies and high number of degrees-of-freedom to work in cluttered and complex environments.

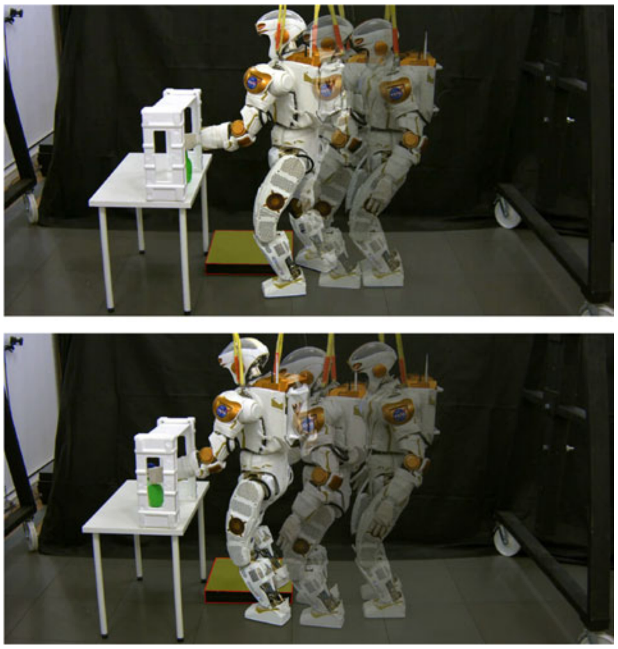
Fig. 1. Motion planning of grasping on uneven terrains. The robot automatically chooses collision-free stance locations and grasping configurations.

Since the DARPA Robotics Challenge (DRC), many studies attempted to relieve human operators from manually providing the end-pose for mobile manipulators and humanoid robots [1]. Zacharias *et al.* [2], [3] proposed a robot capability/reachability map (RM) to analyze, record, and access the information about how a fixed-base manipulator can reach different workspace poses. The concept of reachability has also been applied in other domains, such as human-robot interaction [4] where the RM is used to guide the robot movement; and multi-contact locomotion [5], [6] where the reachability is used to automatically find possible contacts for legged systems.

Manuscript received February 15, 2017; accepted June 12, 2017. Date of publication July 17, 2017; date of current version July 24, 2017. This letter was recommended for publication by Associate Editor E. Yoshida and Editor N. Amato upon evaluation of the reviewers' comments. This work was supported by the Engineering and Physical Sciences Research Council under Grant EP/L016834/1. The work has been performed at the University of Edinburgh under the Centre for Doctoral Training in Robotics and Autonomous Systems program. *(Corresponding author: Yiming Yang.)*

Y. Yang, W. Merkt, V. Ivan, and S. Vijayakumar are with the Institute of Perception, Action, and Behaviour, School of Informatics, University of Edinburgh, Edinburgh EH8 9AB U.K. (e-mail: yiming.yang@ed.ac.uk; wolfgang.merkt@ed.ac.uk; v.ivan@ed.ac.uk; sethu.vijayakumar@ed.ac.uk).

H. Ferrolho is with LIACC, DEI, Faculdade de Engenharia, Universidade do Porto, Porto 4099-002, Portugal (e-mail: henrique.ferrolho@fe.up.pt).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org

Digital Object Identifier 10.1109/LRA.2017.2727538

Though originally designed for fixed-base robots, the reachability map (RM) was directly extended to mobile systems by randomly or systematically selecting different base positions [7], [8]. Vahrenkamp *et al.* [9] introduced the inverse reachability map (IRM) by encoding the reachability information in the end-effector frame rather than in the fixed base frame, which allows floating-base robots to automatically find appropriate stance locations and reaching configurations given a desired end-effector pose. The IRM method only considers kinematic feasibility of the reaching problem without taking into account collisions between the robot and its environment. Collisions have to be checked online. This was applied to a humanoid robot to find $SE(2)$ (flat terrain) stance locations which vastly improves the success rate for humanoid manipulation [10]. Yang *et al.* [11] proposed the inverse dynamic reachability map (iDRM) in which the IRM was extended by utilizing a configuration-to-workspace occupation mapping [12] to enable efficient collision updates. Thus, iDRM is able to remove a large number of colliding samples and find collision-free end-poses in real-time as the collision computation and encoding is part of the pre-processing, while online, the map is only filtered and the highest scoring sample is selected.

Similar to [10], iDRM [11] only considers single stance locations on a flat plane in $SE(2)$ (2D position, 1D orientation), i.e., the relative positions of two feet are fixed with the same orientations on the horizontal surface. Moreover, the method only resolves end-pose problem for reaching using a single arm as a proof of concept. These simplifications provide the ability to solve a majority of manipulation scenarios interactively in real-time [11], and leave the adaptation of uneven surface for the feedback control stage [13], [14].

However, to make full use of the dual-arm and bipedal nature of humanoid robots, it is essential to find appropriate end-poses for bimanual manipulation tasks in environments with uneven terrains (i.e., $SE(3)$ for each foot, instead of $SE(2)$ for the mid-point of two feet only). However, it is non-trivial to directly extend the iDRM method to include both dual-arm and bipedal features due to the curse of dimensionality, as the memory required to ensure a sufficient configuration space coverage increases exponentially making it infeasible to run on current commodity hardware.

To resolve this issue, we propose a hybrid approach which combines the advantages of both the *Forward Dynamic Reachability Map* (DRM) and the *Inverse Dynamic Reachability Map* (iDRM) to plan end-poses for humanoid robots in complex and rugged environments. We use an upper-body iDRM to first find valid upper-body configurations and pelvis poses. We then use a lower-body DRM to find valid leg configurations on uneven floors. A valid full-body end-pose is then obtained by combining valid upper-body and lower-body configurations. After finding the end-pose, we then employ state-of-the-art walking planners such as [15] to plan footsteps for the robot to walk to that desired end-pose. Finally after arriving at the pre-action stance location, we can use full-body motion planners such as [16] to generate full-body reaching motions to complete the task. We have validated our work on the 38-DoF NASA Valkyrie humanoid robot and demonstrated that the proposed method is able to find valid, i.e., balanced and collision-free, end-poses for humanoid robots online for grasping tasks on uneven terrains, as shown in Fig. 1.

II. Humanoid Motion Synthesis

It is important to take full advantage of the mobility of humanoids for grasping and manipulating distant objects. A grasping task can be decomposed into three main actions similar to [11]:

1) *End-pose planning*: find an appropriate pre-grasp stance location and grasping configuration,

$$\mathbf{p}^*, \mathbf{q}^* = EndPosePlan(\mathbf{p}_s, \mathbf{q}_s, \mathbf{y}^*) \tag{1}$$

2) *Footstep planning and execution*: plan and execute a sequence of footsteps to walk to the pre-grasp stance location,

$$\mathbf{p}_{[0:T]} = FootstepPlan(\mathbf{p}_s, \mathbf{p}^*) \tag{2}$$

3) *Motion planning and execution*: plan and execute a full-body collision-free motion to complete the task,

$$\mathbf{q}_{[0:T]} = MotionPlan(\mathbf{q}^*) \tag{3}$$

where $\mathbf{p}_s$ and $\mathbf{q}_s$ are the current stance location and robot configuration, $\mathbf{y}^* = \{\mathbf{y}^*_{lhand}, \mathbf{y}^*_{rhand}\} \in 2 \times SE(3)$ are the desired poses for the left and right hands. An end-pose contains the desired stance location $\mathbf{p}^* = \{\mathbf{p}^*_{lfoot}, \mathbf{p}^*_{rfoot}\} \in 2 \times SE(3)$ and reaching configuration $\mathbf{q}^* \in \mathbb{R}^N$, which will later be used as the goal configuration in the motion planning module.

In most practical applications, the end-pose is provided manually by the operator because an automated solution is non-trivial, especially in complex environments with uneven terrains. To improve robot autonomy, we focus on solving the key issue of end-pose planning in this work, and use existing methods which can already efficiently plan footsteps and full-body motion.

We first explain the DRM and iDRM methods in Section III, and then discuss how to utilize the strengths of both to plan valid end-poses for dual-arm humanoid robots in complex environments with uneven terrains in Section IV.

III. Dynamic Reachability Maps

The forward and inverse dynamic reachability maps, i.e., DRM and iDRM, are the mappings from robot configuration space to workspace with an efficient indexing technique that updates the collision status of millions of configurations in real-time. DRM and iDRM are defined with respect to the base frame and the end-effector frame respectively. In other words, DRM encodes information of "*when fixing the base, what is the reachable space of the end-effector*", whereas iDRM encodes "*to reach a desired pose, where to best place the base*". However, from an algorithmic perspective, DRM and iDRM are very similar, and both of them have two stages: offline preprocessing and online planning.

A. *Offline Preprocessing*

The offline preprocessing contains four major steps for both DRM and iDRM, as highlighted in Fig. 2. First, the workspace

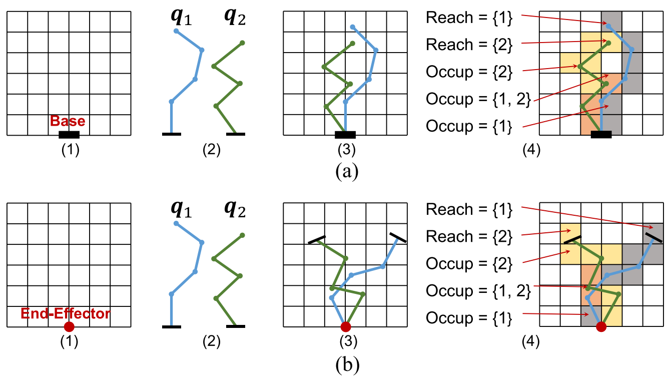

Fig. 2. Examples of DRM and iDRM offline map construction: (1) discretized space; (2) generate valid samples; (3) transform samples to map origin; (4) generate reach and occupation lists. (a) Forward dynamic reachability map (DRM). (b) Inverse dynamic reachability map (iDRM).

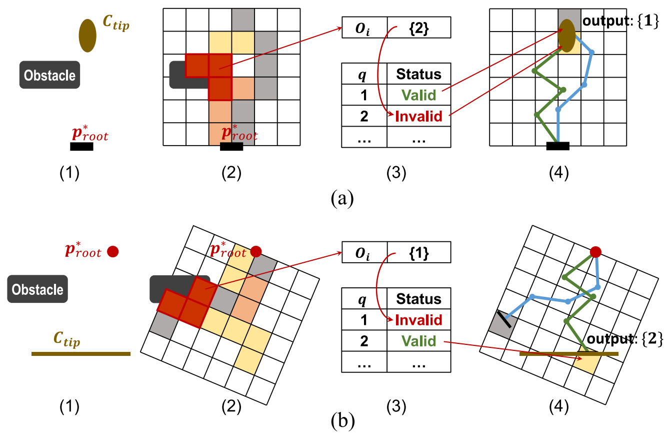

Fig. 3. Examples of DRM and iDRM online update: (1) problem setup; (2) transform map to root pose; (3) validate collision status; (4) check tip pose constraints and find valid samples. (a) Forward dynamic reachability map (DRM). (b) Inverse dynamic reachability map (iDRM).

is discretized into a bounded 3D voxel grid $\mathbb{V}$. The grid of DRM is defined with respect to the base frame while the grid of iDRM has its origin in the end-effector frame. In this letter, *root link* refers to the reference link, i.e., the base link for DRM and the end-effector link for iDRM. Also, *tip link* refers to the end-effector link for DRM and the base link for iDRM. Both DRM and iDRM can only have one root link but multiple tip links. Next, we generate M number of valid samples,[1] which are then transformed to the origin of the corresponding map. The last step generates the reach list R_v and occupation list O_v for each grid voxel $v \in \mathbb{V}$. The reach list R_v stores the indices of samples whose tip link falls into this voxel v. For a robot with K tip links, the reach list stores a list of paired values specifying both sample and tip indices, i.e., $R_v = \{(m, k) \ldots\}$, where $m \in M$ is the sample index and $k \in K$ is the tip index. In Fig. 2 , a robot model with only one tip link is used for clarity. Finally, the occupation list O_v is generated to store the list of samples that intersect with voxel v.

B. Online Update

During the online update phase, our goal is to find samples that are collision-free and satisfy tip link constraints C_{tip} given the root pose $\mathbf{p}^*_{root}$, as highlighted in Fig. 3. C_{tip} defines valid position and orientation regions for different tip links. Firstly, the DRM/iDRM map is transformed to $\mathbf{p}^*_{root}$. Conventional collision checking is then deployed to identify the colliding voxels and iteratively invalidate samples in the occupation list O_v of all colliding voxels. Finally, we check the reach lists of candidate voxels to find valid samples that satisfy C_{tip}, so the output samples are guaranteed to reach the target while being collision-free. For example, in Fig. 3, two samples from the DRM satisfy the tip pose constraint, but only sample 1 was selected since the other sample was invalidated during the collision update step. In the iDRM case, sample 1 was excluded from the result as it was in collision and violated the tip pose constraint.

[1]A valid sample has to satisfy a combination of kinematic joint limits of the robot, self-collision-free, static balance, etc.

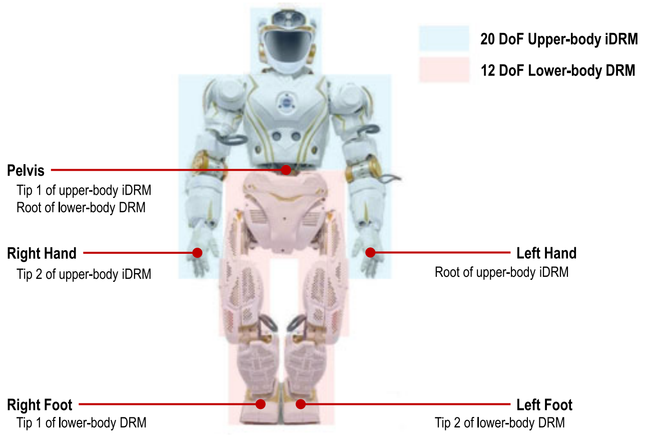

Fig. 4. Upper-body iDRM and lower-body DRM for the 38-DoF Valkyrie Robot. Each leg has 6 DoF and each arm has 7 DoF, the torso has 3 DoF, and the neck has 3 DoF. The pelvis represents an extra 6 DoF virtual joint that connects the robot to the world.

IV. End-Pose Planning for Bi-Manual Tasks on Uneven Terrain

The iDRM can be used directly for humanoid end-pose planning with the constrained positions of two feet [11], which is limited to flat ground only. As the iDRM can have multiple tip links, a direct and naïve approach is to create an iDRM with one root link and three tip links, where one hand is selected as the root and the rest three limbs are treated as tip links. However, this significantly increases the dimensionality of the problem, i.e., the number of samples increases exponentially for each tip link to cover the high dimensional space (see Section V). Consequently, the required memory size is so large that it becomes infeasible to run on any commodity hardware.

To plan end-poses on uneven terrain while keeping a manageable number of samples and memory size, we take advantage of the robot's inherent structure to treat upper-body and lower-body separately. We separate the robot at the torso pelvis joint, as illustrated in Fig. 4. We create an iDRM for the upper-body and a DRM for the lower-body. We choose one hand as the root

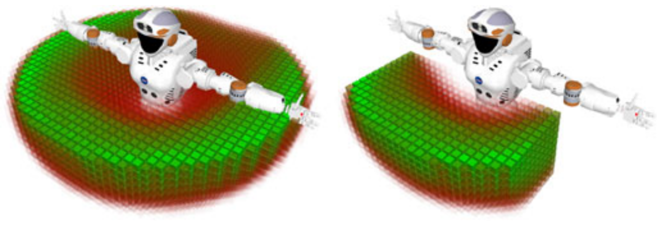
Fig. 5. Reachability map of upper-body: (left) unconstrained; (right) constrained to the front of the robot. All colored voxels are reachable and the green ones are of higher reachability scores. Only part of the map is plotted for clarity (the whole map has a spherical shape).

of the upper-body iDRM, and the other will become a tip link. We could further split the kinematic structure to obtain more but smaller components, i.e., further split the upper-body into left and right arms. However, as we will show later in V-C2, the proposed splitting approach is more efficient considering the trade-off between success rate and planning time. In the rest of this section, we will discuss how these two maps are created and combined to plan end-poses on uneven terrains.

A. Constructions of DRM/iDRM for Humanoids

1) Upper-Body iDRM: In this case study, the left hand is selected as the root link of the upper-body iDRM, and the right hand and pelvis are treated as two tip links. Several iDRM datasets with different number of samples (all with 10cm workspace voxel resolution) are generated for the 20-DoF upper-body of Valkyrie. Traditionally, samples of an inverse reachability should cover the whole configuration space, i.e., for the case of a humanoid, samples of the map should reach behind the robot. However, since the robot's sensor are predominantly facing forward, we want to express a preference for stable stance locations that give us reasonable manipulability. We adopt a heuristic in our method, where we only store samples with both hands reaching comfortable manipulation poses in front of the robot, as shown in Fig. 5. Note that the robot can still manipulate objects that are currently far away or behind the robot by walking to an appropriate pre-action stance location, which is the key point of end-pose planning.

2) Lower-Body DRM: The lower-body of Valkyrie has 12-DoF (6-DoF per leg). Though the legs have a large range of motion, the manifold of balanced configurations is much smaller even on uneven terrain. Therefore, we have reduced the "*reachability*" map for the lower-body so that the legs have the range to adapt to the uneven terrain but they won't reach most unnatural poses.[2] To this end, we generate lower-body configurations with two feet placed in a region below the pelvis ($0.8-1.1$ meters for Valkyrie), as shown in Fig. 6.
This ensures that the lower-body DRM has sufficient samples to adapt to uneven terrain without demanding extra memory for storing poses that can't provide support for the robot, e.g., poses where the feet reach above the pelvis.

[2]Here we define the terms *natural* and *comfortable* as the distance in the configuration space from a nominal one derived from the posture shown in Fig. 4, though a metric of being "unnatural" might appear to be subjective.

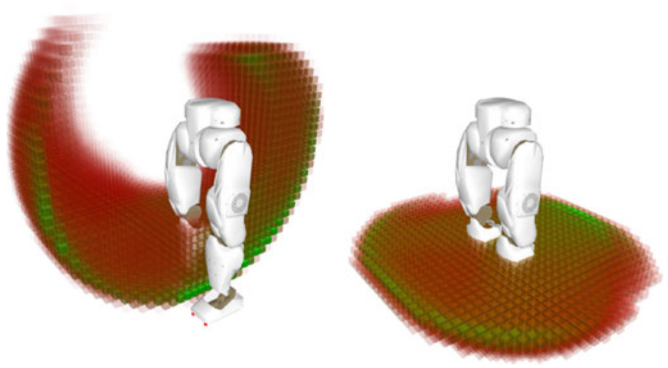
Fig. 6. Reachability map of the lower-body: (left) a unconstrained case, only part of the map is plotted for clarity; (right) a constrained case where feet are placed below the pelvis.

Algorithm 1: Humanoid End-Pose Planning.

```
Require: $\mathbf{y}^*_{lhand}, C$
Ensure: $\mathbf{p}^*_{lfoot}, \mathbf{p}^*_{rfoot}, \mathbf{q}^*$
 1: $\mathbf{y}^*_{root} = \mathbf{y}^*_{lhand}$
 2: Transform $M_{upper}$ to $\mathbf{y}^*_{root}$    // Fig. 3(b)(2)
 3: CollisionUpdate($M_{upper}$)    // Fig. 3(b)(2-3)
 4: $\mathbf{Q} = \emptyset$
 5: for $\forall \mathbf{q}_n \in$ collision-free subset of $M_{upper}$ do
 6:     $T_{pelvis}, T_{rhand}$ = TipGlobalPoses($\mathbf{q}_n, \mathbf{y}^*_{root}$)
 7:     if SatisfyConstraint($T_{pelvis}, T_{rhand}, C$) then
        // Fig. 3(b)(4)
 8:         Transform $M_{lower}$ to $T_{pelvis}$    // Fig. 3(a)(2)
 9:         CollisionUpdate($M_{lower}$)    // Fig. 3(a)(2-3)
10:         for $\forall \mathbf{q}_m \in$ collision-free subset of $M_{lower}$ do
11:             $\mathbf{p}_{lfoot}, \mathbf{p}_{rfoot}$ =
                TipPoses($T_{pelvis}(\mathbf{q}_n), \mathbf{q}_m$)
12:             if ValidTerrainContact($\mathbf{p}_{lfoot}, \mathbf{p}_{rfoot}$) then
        // Fig. 3(a)(4)
13:                 $\mathbf{q} = \{\mathbf{q}_n, \mathbf{q}_m\}$
14:                 if $\mathbf{q}$ is balanced then
15:                     $\mathbf{Q} = \mathbf{Q} \cup (\mathbf{p}_{lfoot}, \mathbf{p}_{rfoot}, \mathbf{q}, f(\mathbf{q}))$
16: $\mathbf{p}^*_{lfoot}, \mathbf{p}^*_{rfoot}, \mathbf{q}^*$ = LowestCost($\mathbf{Q}$)
    return $\mathbf{p}^*_{lfoot}, \mathbf{p}^*_{rfoot}, \mathbf{q}^*$
```

B. End-Pose Planning

Let M_{upper} be the upper-body iDRM and M_{lower} be the lower-body DRM. Given a task $\mathbf{y}^* = (\mathbf{y}^*_{lhand}, \mathbf{y}^*_{rhand})$, start states $\mathbf{p}_s, \mathbf{q}_s$ and the environment Env, the end-pose planner needs to find an end-pose that contains $\mathbf{p}^* = (\mathbf{p}^*_{lfoot}, \mathbf{p}^*_{rfoot})$ and $\mathbf{q}^*$. Firstly, we create two tip pose constraints $C = \{C_{pelvis}, C_{rhand}\}$ for the upper-body iDRM, where C_{pelvis} constrains the pelvis link to be inside a feasible height region and approximately perpendicular to the ground, and C_{rhand} constrains the right hand to be near $\mathbf{y}^*_{rhand}$. Algorithm 1 explains the end-pose planning method for bimanual tasks on uneven terrain, where lines 1-7 M_{upper} are used to find collision-free upper-body configurations that satisfy the constraints C, so that two hands can reach the goal $\mathbf{y}^*$ with the pelvis pose T_{pelvis}.

It is worth emphasizing that, given a upper-body configuration $\mathbf{q}_n$, these poses can be directly retrieved from iDRM, though the global pose of a link can be calculated by forward kinematics. For each tip link, i.e., pelvis and right hand, the iDRM reach pose is referenced in the root (left hand) frame. Given the desired root pose $\mathbf{y}^*_{lhand}$, the global pose of a tip link is

$$T_n^{tip,world} = \mathbf{y}^* \times T_n^{tip,root} \tag{4}$$

where $T_n^{tip,world}$ and $T_n^{tip,root}$ represent the tip pose of sample n in global and root frames accordingly. Here $T_n^{tip,root}$ is pre-computed for each sample during offline processing and $\mathbf{y}^*$ is given for each task. Hence, computing the global poses of the pelvis and the right hand is very efficient in our approach.

After retrieving the global poses, we can then check if the configurations satisfy the pelvis and right hand constraints. For a candidate upper-body configuration $\mathbf{q}_n$, we transform M_{lower} to T_{pelvis} and find valid lower-body configurations, i.e., collision-free and valid contacts with the terrain, as shown in lines 8–12 of Algorithm 1. To check foot contacts, we first extract the step regions from the environment. Similar to 4 with T_{pelvis} as the $\mathbf{y}^*$, we can obtain the tip (foot) poses in the global frame and check if the foot is within the step regions. If the lower-body configuration has valid contacts, we then combine the candidate upper and lower body configurations to acquire the full-body configuration. Since multiple valid end-poses may exist, we iterate though M_{upper} and M_{lower} to find the best candidate based on the cost function $f(\mathbf{q})$. Different cost functions can be defined for different tasks and environments. The following cost function is used

$$f(\mathbf{q}) = \|T_{pelvis}(\mathbf{q}) - T_{pelvis}(\mathbf{q}_s)\|_{W_1} + \|\mathbf{q} - \mathbf{q}_s\|_{W_2}, \tag{5}$$

where W_1, W_2 are weights, for obtaining an end-pose with minimum travelling distance to the start/nominal configuration.

After end-pose planning, the last step is to refine the output to satisfy all necessary constraints, e.g., the hands need to precisely reach the target, the feet need to be perfectly in contact with the terrain, and the pose shall be statically balanced. A non-linear optimization-based solver [17] (referred to as the non-linear IK in the rest of the letter) is used to optimize the candidate end-pose by sequential quadratic programming (SQP) considering these constraints

$$\begin{aligned} \mathbf{q}^* &= \arg \min_{\mathbf{q} \in \mathbb{R}^{N+6}} \|\mathbf{q} - \mathbf{q}_s\|^2_{Q_q} \\ \text{s.t. } & \mathbf{b}_l \leq \mathbf{q} \leq \mathbf{b}_u \\ & c_i(\mathbf{q}) \leq 0, c_i \in \mathbf{C} \end{aligned} \tag{6}$$

where $Q_q \succeq 0$ is the weighting matrix, $\mathbf{b}_l$ and $\mathbf{b}_u$ are the lower and upper joint bounds, and $\mathbf{C}$ is the set of constraints. If the solver fails or the solution is in collision, the optimization is repeated with the next best candidate end-pose.

C. Footstep and Motion Planning

After finding the end-pose, a footstep planner is invoked to plan a set of footsteps to enable walking from current stance location $\mathbf{p}_s$ to pre-grasp stance location $\mathbf{p}^*$, followed by a motion planner to generate a valid full-body trajectory to realize the end-pose $\mathbf{q}^*$. Footstep and motion planning are not the main focus of this work, and any suitable algorithms could be used. The footstep planner from [15] and the full-body motion planner from [16] are used in our experiments.

TABLE I
MAP CONSTRUCTION ANALYSIS

Map		No. samples	Construction time (min)	File size (MB)
Upper-body two arms	Φ_{1a}	10^5	28.8	108
	Φ_{1b}	10^6	289.7	1,082
	Φ_{1c}	4×10^6	1090.8	4,352
Upper-body left arm	Φ_{2a}	10^4	0.25	9
	Φ_{2b}	10^5	2.61	91
	Φ_{2c}	10^6	25.0	879
Right arm	Φ_{3a}	10^4	0.05	2
	Φ_{3b}	10^5	0.58	22
	Φ_{3c}	10^6	6.19	217
Lower-body two legs	Φ_{4a}	1,680	0.24	1
	Φ_{4b}	44,400	6.15	33
	Φ_{4c}	227,400	30.0	160
	Φ_{4d}	742,560	103.5	535

V. EVALUATION

A. Construction of Dynamic Reachability Maps

We have generated maps with different root/tip links and number of samples to analyze how different splitting of the map affects the performance:

1) Φ_1: An upper-body iDRM with the left hand as the root, pelvis and right hand as the tips. Three datasets are generated with different number of samples: $100{,}000\ (\Phi_{\mathrm{1a}})$, $1,000,000\ (\Phi_{\mathrm{1b}})$ and $4{,}000{,}000\ (\Phi_{\mathrm{1c}})$.
2) Φ_2: An upper-body iDRM with the left hand as the root, pelvis and right shoulder as the tips. Three datasets are generated with different number of samples: $10{,}000\ (\Phi_{\mathrm{2a}})$, $100{,}000\ (\Phi_{\mathrm{2b}})$ and $1{,}000{,}000\ (\Phi_{\mathrm{2c}})$.
3) Φ_3: A right arm DRM with right shoulder as the root and right hand as the tip. Three data sets are generated with different number of samples: $10{,}000\ (\Phi_{\mathrm{3a}})$, $100{,}000\ (\Phi_{\mathrm{3b}})$ and $1{,}000{,}000\ (\Phi_{\mathrm{3c}})$.
4) Φ_4: A lower-body DRM with the pelvis as the root, left and right feet as the tips. Four datasets are generated with different number of samples : $1{,}680\ (\Phi_{\mathrm{4a}})$, $44{,}400\ (\Phi_{\mathrm{4b}})$, $227{,}400\ (\Phi_{\mathrm{4c}})$ and $742{,}560\ (\Phi_{\mathrm{4d}})$.

All datasets are created with 10 cm workspace grid resolution. The construction time and file size are highlighted in Table I. The construction time of Φ_1 maps are relatively longer because many of the samples are rejected and only those where both hands fall into the region of interest are kept. The Φ_1 maps are also expensive to store since the kinematic structure includes the entire upper-body with two arms. It is worth emphasizing that the file size of Φ_1 is similar to Φ_2 and Φ_3 combined with same number of samples, e.g., $\Phi_{1b} \approx \Phi_{2c} + \Phi_{3c}$.

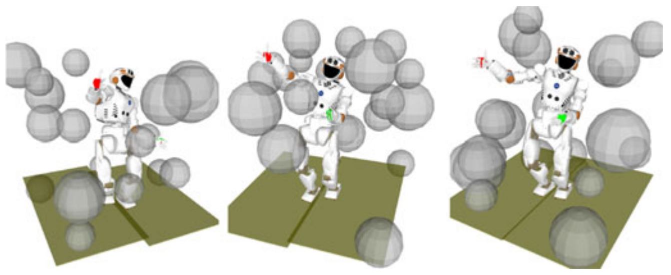

Fig. 7. Examples tested in the benchmark. The robot needs to reach desired left (green) and right (red) hand targets while avoiding spherical obstacles (grey) and keeping balance on the uneven terrain.

The proposed end-pose planning method can be obtained by combining Φ_1 and Φ_4, for example, combining Φ_{1a} and Φ_{4a} gives a dataset with a theoretical $10^5 \times 1680 = 168$ million full-body configurations; combining Φ_{1c} and Φ_{4c} gives a dataset with a theoretical 909.6 trillion full-body configurations. A further split method can be obtained by combining Φ_2, Φ_3 and Φ_4, for example, combining Φ_{2c}, Φ_{3c} and Φ_{4c} gives a dataset with a theoretical 2.274×10^{17} full-body configurations. It is clear that the total number of full-body configurations increases exponentially with the number of components. However, combining these maps significantly slows down the on-line planning (see Section V-C2).

B. End-Pose Planning Benchmarking Setup

We have created a set of benchmark problems by passing random hands and feet pose constraints, as well as quasi-static balance constraint, into the full-body IK solver to obtain a random but balanced configuration. The configurations are filtered for self-collisions. We then populate spherical obstacles into the free environment randomly but not colliding with the robot until a required number of obstacles is reached. Finally, we can extract the height and position of each foot from the generated configuration and create terrain areas accordingly. A valid end-pose planning problem is thereby generated. We also store the desired poses for both hands, collision environments and terrain areas. Note that the robot configurations are generated to ensure that at least one solution exists. The configuration is not known to the candidate algorithm, and the algorithm is allowed to find a different but valid solutions if multiple solutions exist. In our benchmarking, we created 1000 random problems, each of which contains 20 spherical obstacles with 5–10 cm radius, as highlighted in Fig. 7.

C. Simulation Benchmarking

1) Different Lower-Body Datasets: As we have mentioned, the lower-body is used for maintaining balance rather than for maximum reachability. Thus, we should use a dataset that contains enough samples which is sufficient for finding balanced configurations rather than having a dataset with millions of samples that consumes huge amount of memory and slows down on-line computation. We combine Φ_{1b} with different Φ_4 maps to analysis the effects different lower-body maps might introduce and therefore select the suitable one for other experiments.

TABLE II
ANALYSIS OF END-POSE PLANNING PERFORMANCE

Method	Map success rate	IK success rate	Final success rate	Avg. time (s)
$\Phi_{1b} + \Phi_{4a}$	72.7%	71.8%	71.4%	0.08 ± 0.02
$\Phi_{1b} + \Phi_{4b}$	73.7%	72.8%	72.5%	0.09 ± 0.03
$\Phi_{1b} + \Phi_{4c}$	80.7%	79.0%	78.7%	0.13 ± 0.10
$\Phi_{1b} + \Phi_{4d}$	86.3%	84.8%	84.2%	0.23 ± 0.33
Non-Linear IK	–	99.8%	59.3%	0.03 ± 0.01
$\Phi_{1a} + \Phi_{4c}$	57.9%	57.1%	56.8%	0.04 ± 0.01
$\Phi_{1c} + \Phi_{4c}$	88.6%	85.7%	**85.1%**	0.40 ± 0.37
$\Phi_{2a} + \Phi_{3a} + \Phi_{4c}$	70.0%	65.1%	63.7%	0.10 ± 0.05
$\Phi_{2b} + \Phi_{3b} + \Phi_{4c}$	91.3%	83.5%	80.4%	0.56 ± 0.39
$\Phi_{2c} + \Phi_{3c} + \Phi_{4c}$	96.9%	85.0%	81.2%	8.08 ± 4.68

Considering the trade-off between success rate and planning time, the method $\Phi_{1c} + \Phi_{4c}$ is used for hardware experiments.

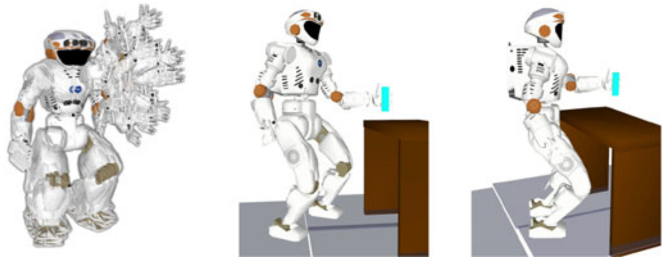

Fig. 8. The first figure highlights the upper-body iDRM and lower-body DRM samples, followed by two scenarios of selected end-poses.

We also evaluated the performance by directly applying the non-linear IK without using DRM/iDRM. The first 5 rows of Table II show the success rate and average planning time using different methods. The map success rate is the rate of DRM/iDRM reports finding valid candidate end-poses, which is then passed to the IK adjustment function. The IK success rate refers to the cases where the non-linear IK successfully adjusted the candidate poses and satisfied all constraints. The pose is then passed to a collision checking function, a final success is reported if the pose is collision-free.

We notice that these methods can not achieve 100% success rate, which is caused by several factors: firstly, although we have created each map with millions of configurations, it is still inefficient to cover the high dimensional full-body configuration space (38 dimension for Valkyrie); secondly, in the interest of time, we only allow the method to try the first 10 different poses from $\mathbf{Q}$, where a valid pose with relatively high cost might be discarded; lastly, some valid poses which are not in collision may get invalidated due to aliasing of the occupancy grid. Such artefacts can be reduced by using a finer workspace grid, but they can't be completely eliminated. This is a common issue with all grid-based methods.

It is interesting that the final success rate is very close to the initial map success rate, which means that once the DRM/iDRM maps find candidate end-poses, those poses are very likely to be valid. On the other hand, the direct non-linear IK method reports a 99.8% success rate, but only 59.3% is finally valid, e.g., collision-free. The result suggests that

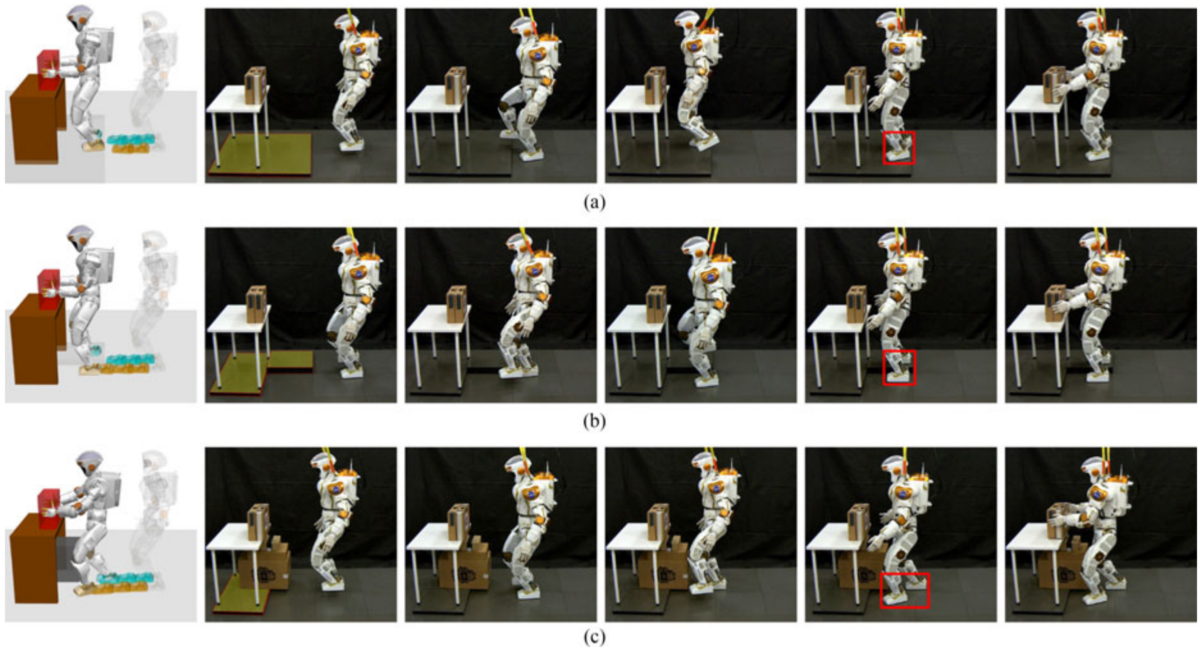

Fig. 9. Bimanual box-picking tasks on the terrains of different heights. The robot is able to automatically find appropriate standing locations and full-body configurations. (a) *B*1: Pick up a box from a higher terrain. (b) *B*2: Pick up a box by placing the right on a higher terrain. (c) *B*3: Pick up a box while the right leg position is restricted by a large obstacle.

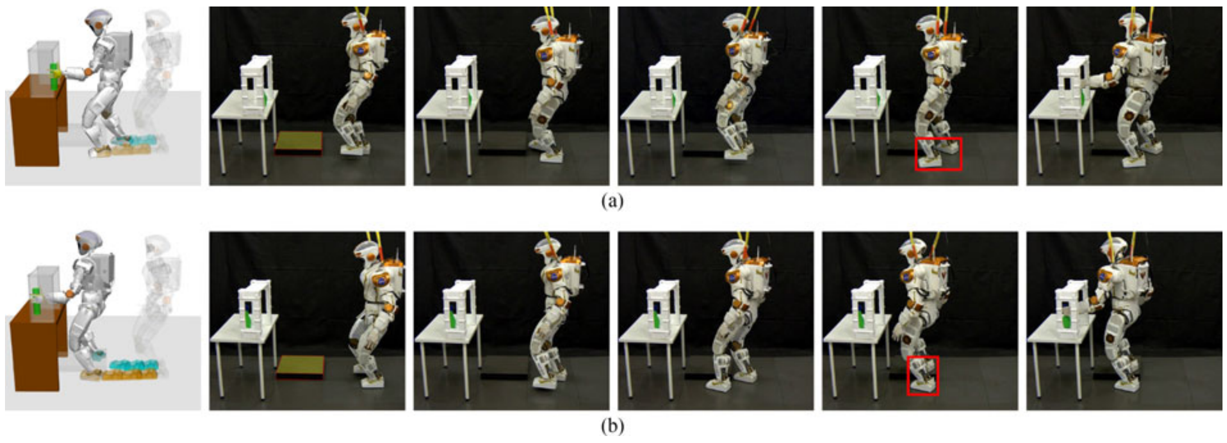

Fig. 10. Single-handed grasping tasks on the terrains of different heights. Case I: the target is easily reachable, so the robot does not need to be too close to the table; Case II, the robot needs to be closer to the table by placing the right foot on the uneven terrain. (a) *S1*: Grasp a target placed at the edge of the table. (b) *S2*: Grasp a target placed deeper on the table.

using only the non-linear IK is inefficient in cluttered environments, and the proposed method is indeed improving the success rate.

The benchmarking was done in randomized and complex environments to fully evaluate different approaches. Although these methods did not achieve 100% success rate, as shown later in Section V-D, they are sufficient for solving practical problems. Based on the result, we conclude that the success rate as well as planning time increase proportionally to the number of lower-body samples. We used the lower-body dataset Φ_{4c} for the rest of the experiments. However, other datasets with more samples might be used depending on the different demands between success rate and planning time.

2) Different Map Combinatorics: We choose to split the humanoid robot into two parts at pelvis. However, one can further split the upper-body into smaller parts, e.g., left body part (Φ_2) and right arm (Φ_3). The last 5 rows of Table II show the end-pose planning result of using different upper-body maps, where the success rate and planning time increases with the number of samples as expected. However, the further splitting ($\Phi_2 + \Phi_3 + \Phi_4$) leads to a much longer planning time while the success rate is not significantly improved compare to the proposed splitting ($\Phi_1 + \Phi_4$). Furthermore, in the case of using further split method with maps $\Phi_{2c} + \Phi_{3c} + \Phi_{4c}$, the final success rate is lower than using proposed split method with maps $\Phi_{1c} + \Phi_{4c}$. Note that the map reports a 96.9% success rate, but dropped to 85.0% after IK adjustment, most of which were caused due to failing to satisfy balance constraint. This means further splitting the body leads to higher chance of violating the balance constraint of the full-body. Splitting the upper- and lower-body at the pelvis link thereby is a reasonable trade-off between coverage, planning success rate, and algorithm runtime. We use the proposed split method with datasets Φ_{1c} for upper-body and Φ_{4c} for lower-body for the following experiments on robot hardware,

D. Hardware Experiments

To demonstrate the capability of end-pose planning on uneven terrain, we created three bimanual box-picking tasks with different terrain types. In the first scenario $B1$ (Fig. 9(a)), the robot had to walk onto a higher floor, which can be theoretically found by classic iDRM as well; in the second case $B2$ (Fig. 9(b)), the robot had to stand on surfaces at two heights; in the last scenario $B3$ (Fig. 9(c)), the robot needed to avoid a collision between its right leg and a large obstacle during the picking task. Our method is capable of finding different collision-free end-poses in these environments. We found that the viable pelvis poses are quite limited in practice for bimanual tasks, i.e., the robot has to stand directly in front of the box for picking it up by two hands. Nevertheless, our DRM/iDRM hybrid method provides a valid solution for the bimanual picking tasks in the presence of uneven terrain.

We further validated two single-arm grasping tasks where the target was placed at different locations, as shown in Fig. 10. A upper-body iDRM was created with the left hand as root link and pelvis as tip link. The right arm joints were set to a pre-defined nominal configuration for all samples, as shown in Fig. 8. The constrain set C then contains pose constraints only for the pelvis but not for the right hand. In the first scenario $S1$ (Fig. 10(a)), the target was placed at the edge of the table, where the robot could easily grasp without being too close. So, the robot stayed away from the high surface, and kept the target within a reachable distance. Whereas in the second task $S2$ (Fig. 10(b)), the target was placed farther away and enclosed by the obstacle. The end-pose planner found a feasible configuration to place two feet on different surfaces so the robot was close enough for grasping the target.

It shall be highlighted that the modular and combined forward inverse dynamic reachability maps is capable of finding end-poses including lunging body or taking a sidestep (in scenarios *B3* and *S1*) which increases the reachable workspace by leveraging the advantage of a legged system. This method performs better than the prior work [10], [11] which limited the foot poses to a constant distance and planning for the mid-feet point. A supplementary video can be found at `https://youtu.be/o-05EHf-gg8`.

VI. Conclusion

We presented a novel end-pose planning algorithm that combines the Dynamic Reachability Map (DRM) and inverse Dynamic Reachability Map (iDRM), which allows humanoid robots to automatically find appropriate end-poses in presence of uneven terrain. Using NASA's Valkyrie humanoid as a testbed, we demonstrated the effectiveness of the proposed method in planning end-poses for both single-arm and bimanual tasks on uneven terrains.

A current limitation of our method is the amount of memory required for storing the maps, e.g., 4.5 GB for Valkyrie using the datasets Φ_{1c} and Φ_{4c}. Our future work involves investigating new methods of encoding the configuration-to-workspace mapping for better memory efficiency. This will allow us to increase the resolution of the voxel grid and improve the success rate of our method.

References

[1] J. James, Y. Weng, S. Hart, P. Beeson, and R. Burridge, "Prophetic goal-space planning for human-in-the-loop mobile manipulation," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots*, 2015, pp. 1185–1192.

[2] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: Representing robot capabilities," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 3229–3236.

[3] F. Zacharias, C. Borst, S. Wolf, and G. Hirzinger, "The capability map: A tool to analyze robot arm workspace," *Int. J. Humanoid Robot.*, vol. 10, no. 4, 2013, Art. no. 1350031.

[4] N. Vahrenkamp *et al.*, "Workspace analysis for planning human-robot interaction tasks," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots*, 2016, pp. 1298–1303.

[5] S. Tonneau, N. Mansard, C. Park, D. Manocha, F. Multon, and J. Pettré, "A reachability-based planner for sequences of acyclic contacts in cluttered environments," in *Proc. Int. Symp. Robot. Res.*, 2015.

[6] P. Kaiser, N. Vahrenkamp, F. Schltje, J. Borrs, and T. Asfour, "Extraction of whole-body affordances for loco-manipulation tasks," *Int. J. Humanoid Robot.*, vol. 12, no. 03, 2015, Art. no. 1550031.

[7] J. Dong and J. C. Trinkle, "Orientation-based reachability map for robot base placement," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 1488–1493.

[8] D. Leidner, A. Dietrich, F. Schmidt, C. Borst, and A. Albu-Schffer , "Object-centered hybrid reasoning for whole-body mobile manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 1828–1835.

[9] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 1970–1975.

[10] F. Burget and M. Bennewitz, "Stance selection for humanoid grasping tasks by inverse reachability maps," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 5669–5674.

[11] Y. Yang, V. Ivan, Z. Li, M. Fallon, and S. Vijayakumar, "idrm: Humanoid motion planning with realtime end-pose selection in complex environments," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots*, 2016, pp. 271–278.

[12] P. Leven and S. Hutchinson, "A framework for real-time path planning in changing environments," *Int. J. Robot. Res.*, vol. 21, no. 12, pp. 999–1030, 2002.

[13] Z. Li, N. Tsagarakis, and D. Caldwell, "Stabilizing humanoids on slopes using terrain inclination estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 4124–4129.

[14] Z. Li, C. Zhou, N. Tsagarakis, and D. Caldwell, "Compliance control for stabilizing the humanoid on the changing slope based on terrain inclination estimation," *Auton. Robots*, vol. 40, no. 6, pp. 955–971, 2016.

[15] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2014, pp. 279–286.

[16] Y. Yang, V. Ivan, W. Merkt, and S. Vijayakumar, "Scaling sampling-based motion planning to humanoid robots," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2016, pp. 1448–1454.

[17] R. Tedrake and the Drake Development Team, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," 2016. [Online]. Available: http://drake.mit.edu