Stance Selection for Humanoid Grasping Tasks by Inverse Reachability Maps

Felix Burget Maren Bennewitz

***Abstract*— In grasping tasks carried out with humanoids, knowledge about the robot's reachable workspace is important. Without this knowledge, it might be necessary to repeatedly adapt the stance location and call an inverse kinematics solver before a valid robot configuration to reach a given grasping pose can be found. In this paper, we present an approach to select an optimal stance location in SE(2) for a humanoid robot's feet relative to a desired grasp pose. We use a precomputed representation of the robot's reachable workspace that stores quality information in addition to spatial data. By inverting this representation we obtain a so-called inverse reachability map (IRM) containing a collection of potential stance poses for the robot. The generated IRM can subsequently be used to select a statically stable, collision-free stance configuration to reach a given grasping target. We evaluated our approach with a Nao humanoid in simulation and in experiments with the real robot. As the experiments show, using our approach optimal stance poses can easily be obtained. Furthermore, the IRM leads to a substantially increased success rate of reaching grasping poses compared to other meaningful foot placements within the vicinity of the desired grasp.**

I. Introduction

Humanoid robots are designed for mobile manipulation tasks in human environments. A prerequisite for successful task completion in such settings is that the robot is equipped with sufficient knowledge about the relevant aspects of the scene and its own capabilities. This also includes the ability to decide where to place itself relative to an object to be manipulated in order to achieve an admissible grasp configurations. Besides by obstacles in the environment, this decision is mainly influenced by the robot's kinematic model including the number of joints and their value range as well as its mass distribution. Based on these parameters, the robot's manipulation capabilities can be uniformly represented by a spatial data structure called *reachability map* [1] (see Fig. 1).

Many existing techniques follow the approach of dealing with locomotion and manipulation tasks as two distinct problems (e.g., [2], [3]). Thus, first a motion for the lower body is planned, i.e., for the base of a mobile platform or the legs of a humanoid, in order to bring an object to be manipulated within the extend of the robots upper body workspace. Subsequently, the reaching task is performed using only the upper body joints. Whether or not the object can actually be reached from the current stance location significantly depends on the number of available joints and constraints involved. Most mobile manipulator and humanoid robot platforms are designed to have at least one degree of redundancy in the upper body end-effector chains. This choice allows the robot to arrive at almost all of its reachable end-effector poses with arbitrary orientation without the need to reposition or reconfigure the mobile base, or lower body respectively. That means the choice of the stance location for a grasping task is compensated by the redundant kinematic structure of the upper body. However, when considering robots with an upper body equipped with only six or less degrees of freedom (DOF), the number of achievable end-effector orientations for the targets within the reachable workspace of the upper body is limited. As opposed to mobile manipulators that are forced to reposition their base once the desired end-effector orientation has been determined to be unachievable from the current stance location, humanoid robots are capable of adjusting their lower body configuration to extend the set of achievable end-effector orientations for a given grasping target. The actuation of the legs of a humanoid, however, introduces balance constraints and thus cannot fully compensate for the limited kinematic properties of the upper body chains. Therefore, the stance selection relative to an object becomes a crucial factor since repeated adaptions of the stance location and calls to an inverse kinematics solver for the arm chain should be avoided.

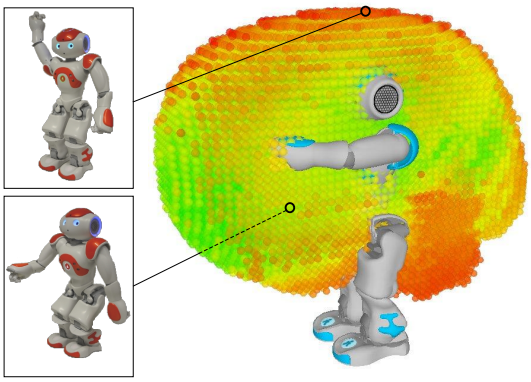

Fig. 1: Representation of reachable right hand locations from statically stable double support poses. Voxels are colored by the maximum value of the manipulability index among the configurations stored in them (green = high, red = low).

In this paper, we present an approach that is based on the concept of inverse reachability maps (IRM) proposed by Vahrenkamp *et al.* [4], [5] to select an optimal stance pose for the feet of a humanoid robot in SE(2), i.e., a 2D

F. Burget and M. Bennewitz are with the Humanoid Robots Lab and the *BrainLinks-BrainTools* Cluster of Excellence, Univ. of Freiburg, Germany. M. Bennewitz is also with the Humanoid Robots Lab, Univ. of Bonn. This work has been supported by the German Research Foundation (DFG) within the Cluster of Excellence *BrainLinks-BrainTools* (EXC1086) and by the European Commission under contract number FP7-610532-SQUIRREL.

position and 1D orientation, relative to the grasping pose. As we show in experiments with a Nao humanoid, we can easily select optimal stance poses from our IRM given a desired grasp. By comparing our approach to random stance pose selection within a meaningful extend around the target grasp, we achieve a substantially increased success rate. Note that such an IRM can also be used for humanoids with a higher number of DOFs to choose optimal stance locations for manipulation tasks.

II. Related Work

Several techniques for generating robot configurations to achieve grasping poses with the end-effector already exist. For example, Stulp *et al.* [6] introduced the concept of Action-Related Place (ARPlace). ARPlace represents a probability mapping that specifies the expected probability that the target object will be successfully grasped, given the positions of the target object and the robot. The authors present results for a wheeled mobile base equipped with two 6-DOF manipulators.

Jamone *et al.* [7] proposed a method to allow a humanoid robot to learn a representation of its own reachable space, referred to as the *reachable space map*, from motor experience. To do so, the authors first learn an arm-gaze model, mapping end-effector poses to a specific gaze configuration for the head and eyes of the robot. Using the inverse of that model an estimate of the reachable space map, i.e., a set of visually detected 3D points, for the robot's hand is obtained by solving a large number of IK queries. The authors differentiate between a basic and an enhanced map. While the former only specifies whether a fixated point in the gaze frame is reachable with the hand, the later additionally provides a quality measure about the degree of reachability.

Müller *et al.* precompute a reachability map for the arm chain where the origin of the reachability map is located in the shoulder [8]. While this approach is rather efficient, the manipulation range is restricted since whole-body motions are not considered but only arm motions, which are planned using A* search in workspace using configuration lookup tables and specialized heuristics considering the difference of the planned arm configuration and the target configuration. Zacharias *et al.* [1] proposed the so-called capability map which is a representation of kinematic reachability in the workspace for a robotic arm. The authors use a manipulability measure to evaluate poses in the workspace with regard to distance to singularities [9]. Using the capability map, one can determine relative object positions that allow for good manipulation by the robot. Vahrenkamp *et al.* [4] presented an approach to compute base positions for a wheeled mobile manipulation robot in order to reach object grasps. The authors precompute a so-called inverse reachability distribution (IRD) for the robot's base, which is centered at the hand pose. For a desired grasp, the IRD center is placed at the corresponding workspace pose and its volume is cut with the floor plane to find valid base positions and orientations. The possible base positions are evaluated based on factors such as the distance to singular configurations, joint limits under redundancy, and the distance between the robot's body parts and the approach has also been extended to bi-manual manipulation tasks [10]. Recently, Kaiser *et al.* [11] proposed to learn whole-body affordances associated with parts in the environment and use manipulability and stability maps to determine reachable locations.

Berenson *et al.* introduced the constrained bidirectional RRT (rapidly exploring random tree) planner that considers constraints as task space regions (TSRs). The authors used TSRs initially for goal specification [3] and showed later that more complex constraints can be described by chaining the TSRs together. However, in this approach it is assumed that the robot is already in a good stance position relative to the object to be manipulated.

In this paper, we extend the concept of inverse reachability maps from [4] to biped humanoids to automatically select an optimal stance location for a given grasping target and a specified manipulability criterion. We hereby have to take into account additional constraints such as kinematic loop closure and stability constraints.

III. Workspace Representation

Inspired by the work of Vahrenkamp *et al.* [4], we describe the robot's reaching and manipulation capabilities by a discretized representation of its workspace. The resulting spatial data structure, referred to as the *reachability map* (RM), is composed of voxels of constant size and is computed offline. Each voxel of the grid represents all possible robot configurations for which the end-effector pose lies within the extend of the voxel.

A. Constructing a Reachability Map (RM)

To build a reachability map for a kinematic chain, we sample configurations from the joint space. By computing the forward kinematics for each sample, the spatial pose of the end-effector and the voxel containing it can be determined. Following this method, we store for each voxel a number of configurations that can subsequently be evaluated based to certain criteria.

B. Manipulability Measure

In this work, we adopt the manipulability measure introduced in [9] in order to assign a quality index to every reachable voxel. This measure evaluates the configurations stored in each voxel in terms of maneuverability of the end-effector in workspace and is computed as

$$w = \sqrt{\det J(q) J^T(q)}, \tag{1}$$

where $J(q)$ is the Jacobian matrix of the respective kinematic chain in configuration q. Using the singular values decomposition, the expression can be rewritten as follows

$$w = \sigma_1 \sigma_2 \cdots \sigma_m, \tag{2}$$

where σ_i is the i-th singular value of the Jacobian matrix and m the dimension of the workspace.

Note that, in general, further aspects such as joint limits under redundancy and the distance between the robots

Algorithm 1: Construct Reachability Map (l_{root}, l_{tip}, Δq, p_{SWF}^{SUF})

```
1  chain ← GET_CHAIN(l_root, l_tip)
2  while q_c ← SAMPLE_CHAIN_CONFIG(chain, Δq) do
3      q_SUL ← GET_SUPPORT_LEG_CONF(q_c)
4      p_hip ← COMPUTE_HIP_POSE(q_SUL)
5      p_SWF ← DESIRED_SWING_FOOT_POSE(p_hip, p_SWF^SUF)
6      q_SWL ← SOLVE_SWING_LEG_IK(p_SWF)
7      if CHECK_CONF_VALIDITY(q_c, q_SWL) then
8          w ← COMPUTE_MANIPULABILITY(q_c)
9          p_tcp ← COMPUTE_FK(q_c)
10         idx ← FIND_EE_VOXEL(p_tcp)
11         RM ← ADD_CONF_TO_VOXEL(idx, q_c, q_SWL, w)
12     end
13 end
```

body parts could be included into the computation of the index [10].

IV. Reachability Maps for Whole-Body Humanoids

Constructing a reachability map for a humanoid robot equipped with multiple end-effectors is a challenging problem due to the high number of degrees of freedom and the number of constraints involved. As opposed to a fixed base and most mobile manipulators for which samples of the joint space are valid as long as they are self-collision free, additional stability issues arise when also lower body joints of a humanoid are considered. Although our framework allows to represent reachability information for arbitrary chains of a humanoid robot, e.g., reachable location of the torso when actuating the leg chains, we are in this work particularly interested in reaching and manipulations tasks and hence in the workspace volume covered by the gripper from statically stable double support configurations. The individual steps for the construction of a reachability map are shown as pseudo code in Alg. 1 and will be explained in detail in the following.

A. Building Whole-Body Reachability Maps

The algorithm takes as input a root and tip link l_{root}, l_{tip} for the chain for which sampling is performed. Here, Δq specifies the step width for sampling. Furthermore, a fixed desired pose p_{SWF}^{SUF} of the swing foot parallel and expressed w.r.t. the support foot, which corresponds also to the root of the sampled chain, is defined. After sampling a configuration of the chain (Line 2 of Alg. 1) the part of the configuration vector q_{SUL} storing the support leg configuration is extracted (Line 3 of Alg. 1) and the forward kinematics is solved to obtain the pose of the hip p_{hip} w.r.t. the support foot (Line 4 of Alg. 1). For a better understanding, Fig. 2 shows the joints and frames of the Nao humanoid robot. Given the hip and the desired swing foot pose expressed in the support foot frame we can easily determine the pose for the swing foot relative to the hip frame p_{SWF} required to achieve a double support configuration with the feet being placed parallel to each other (Line 5 of Alg. 1). Afterwards, the algorithm tries to solve the inverse kinematics problem for the swing leg chain (Line 6 of Alg. 1). If an IK solution is found it is stored in q_{SWL} and the whole-body configuration of the robot is checked for validity, i.e., we determine whether the whole-body pose is statically stable and collision-free (Line 7 of Alg. 1). If no IK solution exists or the pose is found to be invalid, the algorithm proceeds by sampling a new configuration for the kinematic chain and repeats the previous steps. Otherwise, if the configuration is valid the algorithm continues by computing the manipulability measure (see Sec. III-B) for the sampled configuration (Line 8 of Alg. 1). As described in Sec. III-A, the forward kinematics is subsequently computed to obtain the end-effector pose p_{tcp} for configuration q_c of the chain and the index idx of the voxel containing p_{tcp} is determined (Line 9, 10 of Alg. 1). In a final step the sampled configuration, its manipulability measure, as well as the IK solution for the swing leg chain is stored in the voxel with index idx (Line 11 of Alg. 1). Note, that the reachability map need to be built only once in an offline step.

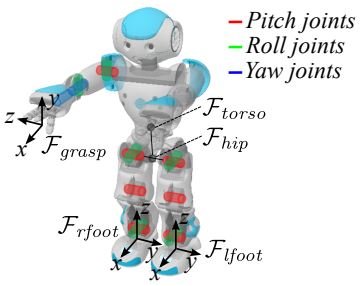

Fig. 2: Kinematic chains and frames of the NAO robot involved in building the RM.

B. $\mathcal{C}$-Space Sampling

To build a reachability map, we need to generate samples from the robot's configuration space $\mathcal{C}$. To do so, values can be either sampled uniformly from the respective joint range or they can be selected iteratively by stepping through the joint value range with a specific increment. The decision of how to sample the $\mathcal{C}$-space significantly depends on the number of DOFs and constraints involved. Considering fixed base and mobile manipulators most of the samples drawn will fall in $\mathcal{C}_{free} \subset \mathcal{C}$, i.e., the subspace of collision-free and hence valid configurations. However, in the case of humanoid robots additional loop-closure and stability constraints arise when leg joints are involved in the sampling process. Hence, the set of admissible configurations $\mathcal{C}_{stable} \subset \mathcal{C}_{free}$ for a humanoid covers only a small portion of the entire $\mathcal{C}$-space. In this work, we follow the approach of generating samples by stepping through the value ranges of the joints actuating the kinematic chains. Here, we define two different increments, a coarse increment for upper body joints and a finer one for lower body joints that directly affect the compliance with loop-closure and stability constraints.

C. Double Support Generation

In our approach, a voxel of the spatial data structure is reachable when the end-effector can be placed within

Algorithm 2: Reachability Map Inversion (RM)

```
1  while v ← GET_VOXEL(RM) do
2      n_c ← GET_NUM_CONFIGS(v)
3      for i = 1 to n_c do
4          (q_c, q_SWL, w) ← GET_CONFIG_DATA(v, i)
5          p_tcp ← COMPUTE_TCP_POSE(q_c)
6          p_base ← (p_tcp)^-1
7          idx ← FIND_EE_VOXEL(p_base)
8          IRM ← ADD_CONF_TO_VOXEL(idx, q_c, q_SWL, w)
9      end
10 end
```

its extend from a statically stable and collision-free double support configuration. Since sampling is performed only for a serial chain of the robot, e.g., for the joint between the foot and gripper link, the loop-closure and stability constraint must be additionally enforced. The former requires the adaption of the swing leg configuration such that the feet of the robot are placed parallel to each other on the floor. Here, we apply the *active-passive link decomposition* method introduced in [12] to achieve a closed-loop configuration for the leg, where the active chain is the support leg for which joint values are sampled and the passive chain is the swing leg. Let us assume w.l.o.g that the right leg is the support leg whose configuration is given by q_{SUL} (Line 3 of Alg. 1). By computing the forward kinematics we obtain the pose p_{hip} of the hip frame $\mathcal{F}_{hip}$ w.r.t. the support foot frame $\mathcal{F}_{rfoot}$ (see Fig. 2). Then, using the fixed transformation p_{SWF}^{SUF} expressing the desired pose of the swing foot frame $\mathcal{F}_{lfoot}$ w.r.t. the support foot we can infer the desired pose p_{SWF} of the swing foot w.r.t. frame $\mathcal{F}_{hip}$. We then apply an inverse kinematics solver to find a configuration for the swing leg q_{SWL}.

V. Reachability Map Inversion

The reachability map generated according to Sec. IV represents the robot's capability of reaching certain end-effector poses from statically stable, collision-free double support configurations. In manipulation and reaching tasks, however, we face exactly the inverse problem. Namely, the required end-effector pose is predefined by the pose of an object to be grasped and we aim at finding a base or feet configuration that maximizes the probability of successful task execution. For this purpose, we use an *inverse reachability map* (IRM) that represents potential base or feet poses relative to the end-effector. The IRM is generated by inverting the previously generated reachability map. The individual steps performed for inverting the reachability information are shown as pseudo code in Alg. 2 and will be explained in detail in the following.

Given the reachability map RM as input, we iterate through its voxels v and for each of them in turn through the n_c configurations stored in it (Line 1 of Alg. 2). Here, the i-th configuration of voxel v is represented by the data structure composed of the configuration of the sampled chain q_c, the configuration of the swing leg q_{SWL} and the manipulability measure w (Line 4 of Alg. 2). By computing the inverse of the end-effector transformation p_{tcp}, obtained by solving

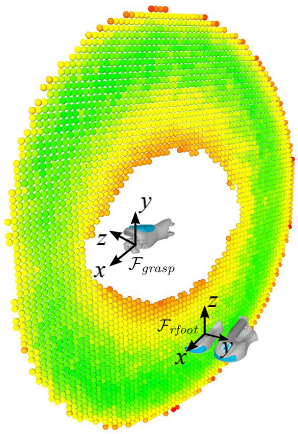

Fig. 3: Cross section through the IRM showing potential right foot locations (left foot is parallel) relative to the hand of the robot. Voxels are colored by their manipulability index (red = low, green = high).

the forward kinematics for the sampled chain, we obtain the pose p_{base} of the support foot w.r.t. the end-effector frame (Line 5, 6 of Alg. 2). Equivalent to the reachability map construction in Alg. 1, we afterwards determine the index idx of the IRM voxel containing the support foot in pose p_{base} and add the configuration to the inverse reachability map data structure (Line 7, 8 of Alg. 2). Note that the map inversion process does not invalidate any configurations of the RM. Thus, no additional check for constraint violation is required. The generated IRM is representing a set of valid stance poses relative to the end-effector independent from any specific grasp configuration. A cross section of an IRM, representing all right foot poses w.r.t. the right gripper is shown in Fig. 3. As with the RM, the IRM needs to be built only once in an offline step and can subsequently be used in all stance pose selection queries for required target grasp poses.

VI. Selecting Stance Poses Using the Inverse Reachability Map (IRM)

Once the IRM has been computed, it can be used to determine the optimal stance pose for a given grasping pose. Here, we assume a 6D target pose for the end-effector given as

$$p_{grasp}^{global} = (x, y, z, roll, pitch, yaw)^T, \tag{3}$$

where $(x, y, z)^T$ and $(roll, pitch, yaw)^T$ is the position and orientation of the desired grasp pose w.r.t. the global frame $\mathcal{F}_{global}$. To obtain the set of potential stance poses for a specific grasp pose from the IRM, we perform the following steps (see also Alg. 3). At first, the IRM is transformed in order to align its center with the grasp frame $\mathcal{F}_{grasp}$ (Line 1 of Alg. 3). Afterwards, we determine the intersection of the transformed map IRM_{grasp} with the ground plane on which the feet must be placed planar (Line 2 of Alg. 3). The resulting layer IRM_{floor} of ground floor voxels represents all support foot positions from which the grasp pose p_{grasp}^{global} is reachable. However, the orientation of the support foot poses

Algorithm 3: Stance Pose Selection (IRM, p_{grasp})

1 $IRM_{grasp} \leftarrow$ TRANSFORM_IRM(IRM, p_{grasp})
2 $IRM_{floor} \leftarrow$ INTERSECT_IRM_FLOOR(IRM_{grasp})
3 $IRM_{stance} \leftarrow$ CHECK_STANCE_FEASIBILITY(IRM_{floor})
4 $p_{stance}^{global} \leftarrow$ GET_MAX_MANIP_VOXEL(IRM_{stance})

stored in IRM_{floor} is not necessarily planar to the ground plane. Therefore, our algorithm iterates through the voxels of the IRM_{floor} and eliminates invalidated configurations. To do so, it determines for each configuration q stored in a voxel of IRM_{floor}, the pose of the support foot p_{SUP}^{global} with respect to the global frame as

$$p_{SUP}^{global} = p_{grasp}^{global} \cdot p_{SUP}^{grasp}, \tag{4}$$

where the transformation p_{SUP}^{grasp} is obtained by solving the forward kinematics for the sampled chain in configuration q. Given the z-axis of $\mathcal{F}_{global}$ being oriented perpendicular to the ground plane, we only need to check whether the roll and pitch angle of p_{SUP}^{global} is sufficiently close to zero in order to determine whether the stance pose is feasible (Line 3 of Alg. 3). Note that the roll and pitch angles are never exactly zero due to the discretization of the joint range for sampling in the generation of the RM. Thus, we consider a certain tolerance for the support foot orientation w.r.t. the ground plane to be acceptable. Furthermore, we exclude all support foot configurations, though kinematically admissible, whose vertical orientation with respect to the grasp frame exceeds $\pm 90^{\circ}$ (Line 3 of Alg. 3). This operation eliminates unnatural configuration in which the robot is oriented with the back to the object to be grasped. Finally, IRM_{stance} represents a set of statically stable, collision-free double support configurations from which the given grasp pose is reachable (see Fig. 4) and the stance pose p_{stance}^{global} of the voxel providing the whole-body configuration with the highest manipulability measure among all voxel configurations in IRM_{stance} can be retrieved (Line 4 of Alg. 3) (see also Fig. 4) and used as input for a footstep planner [13]. Assuming perfect navigation capabilities of the robot, the whole-body configuration stored in the voxel could be even used as a goal configuration for a bidirectional whole-body motion planner [14], once the stance destination has been reached. Thus, subsequent solving of the whole-body inverse kinematics would no longer be required.

VII. Implementation Details

We implemented our approach for pose selection for humanoid robots based on inverse reachability maps in ROS (Robot Operating System) and use FCL (flexible collision library) [15] for self-collision checks. We evaluate the stability of whole-body configurations by checking whether the robot's center of mass (CoM) projected to the ground plane is within the support polygon. For solving the forward and inverse kinematics, we use KDL [16] and find swing leg configurations using the Newton-Raphson numerical IK solver. We developed additional functions to be able to build arbitrary serial chains from the kinematic tree structure representing the robot. Previously, it was only possible to generate kinematic chains going forwards along branches of the kinematic tree.

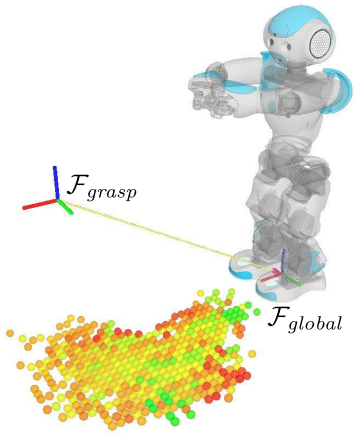

Fig. 4: Potential stance poses for the right foot for reaching a desired grasping target with the right gripper. The left foot is always set parallel to the right. Voxels are colored by the maximum achievable manipulability among the configurations stored in the voxel (red = low, green = high).

VIII. Experiments

For the experimental evaluation of our approach, we use a V4 Nao humanoid by Aldebaran Robotics. The robot is 58 cm tall and has 25 DOF: 2 in the neck, 6 in each arm (including one to open and close the hand), and 5 in each leg. In addition, the legs share a common (linked) hip joint that cannot be controlled independently. Inertia, mass, and CoM of each link are known from CAD models. For efficient collision checks, we created a low-vertex collision mesh model for each of the robot's links from the CAD models. Generally, our approach is capable of performing reachability analysis for arbitrary subchains of the robot, e.g., for the chain leading from the torso down to one of the feet. However, here we consider kinematic chains rooted at one of the feet and ending at one of the grippers/hands leading to 10 DOF. For building the reachability map for the Nao humanoid we used a sampling resolution of 0.3 rad for the upper body joints and 0.2 rad for the lower body joints what we found out to be a good compromise between performance and computational demands. The resulting IRM has a memory consumption of 5GB.

A. Selecting a Stance Pose for Grasping

Fig. 4 shows the robot initially located at the global frame and all potential stance poses for the support foot (here right foot, left foot is always parallel) for a given grasping pose $p_{grasp}^{global} = (0.5, 0.0, 0.3, 0.0, 0.0, 0.0)^T$ specified for the robot's right gripper. Note that each of the voxels (shown as spheres) can represent multiple stance poses of different orientations. The color of the voxels corresponds to the maximum manipulability measure encountered among the configurations stored in it. Finally, the stance pose p_{stance}^{global} with the highest manipulability measure among all voxel

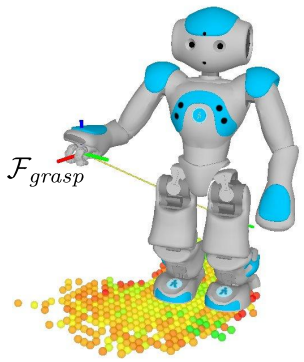

Fig. 5: Example of a stance pose and whole-body configuration for reaching a grasping target.

configurations in IRM_{stance} can easily be determined. Fig. 5 shows a possible stance location and whole-body configuration for reaching the desired grasping pose.

B. Comparison of Stance Selection With the IRM vs. Random Stance Pose Sampling With the RM

In order to quantitatively evaluate the improvement achieved by using inverse reachability information concerning the ability of reaching a given grasping target, we compared it to the success of reaching the same target pose using the forward reachability map. To do so, we randomly selected a stance pose from which the grasping target pose is reachable by at least one support foot orientation according to the reachability map generated in Sec. IV-A. Without further knowledge, this is a reasonable approach. We sampled 10000 double support stance poses and checked whether the grasp pose lying within the extend of the RM is actually reachable from the individual poses with the correct end-effector orientation. The result was that only 17% of the sampled stance locations allowed to reach the correct grasp. When using the IRM, a success rate close to 100% can be reached, depending on the joint sampling resolution chosen within the RM construction process (see Δq in Alg. 1). Thus, using the IRM, the number of succesfully reached target grasps without repeated stance pose adaption is substantially increased. This shows that the humanoid can highly benefit from the knowledge about its own manipulation capabilities represented by the IRM. It takes 820 ms on average on an Intel Core i7 3.4GHz to compute IRM_{floor} and IRM_{stance}, and to find the optimal stance pose.

IX. Conclusions

In this paper, we presented an approach to optimal pose selection for humanoids to carry out object manipulation tasks. We build an inverse reachability map (IRM) that reflects the robot's capability of reaching a grasping pose from possible stance locations, thereby also considering static stability of the robot. The IRM is computed only once in an offline step and can subsequently be used to generate stances poses optimizing a manipulability criterion for arbitrary grasping targets. Optimal stance pose selection is especially important if the number of available DOFs in the upper body is limited and redundancy is not given. By using the IRM for the pose selection, we substantially increase the probability of successful reaching task execution, without the need of repeatedly adapting the stance location and calling an inverse kinematics solver.

As we have shown in our experiments with a Nao humanoid, equipped only with a 5 DOF arm, the set of potential stance poses with high manipulability represented within the IRM is relatively small. This emphasizes the demand for intelligent stance pose selection for such platforms as realized by our technique. Note that our approach of building an IRM for legged humanoids and its use for stance pose selection is general and can also be applied to humanoids with a higher number of DOFs freedom to efficiently select an optimal configuration for manipulation.

References

[1] F. Zacharias, C. Borst, and G.Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[2] T. Rühr, J. Sturm, D. Pangercic, M. Beetz, and D. Cremers, "A generalized framework for opening doors and drawers in kitchen environments," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2012.

[3] D. Berenson, J. Chestnutt, S. Srinivasa, J. Kuffner, and S. Kagami, "Pose-constrained whole-body planning using task space region chains," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2009.

[4] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2013.

[5] N. Vahrenkamp and T. Asfour, "Representing the robot's workspace through constrained manipulability analysis," *Autonomous Robots*, vol. 38, no. 1, 2015.

[6] F. Stulp, A. Fedrizzi, L. Mösenlechner, and M. Beetz, "Learning and reasoning with action-related places for robust mobile manipulation," *Journal of Artificial Intelligence Research (JAIR)*, vol. 43, pp. 1–42, 2012.

[7] L. Jamone, L. Natale, K. Hashimoto, G. Sandini, and A. Takanishi, "Learning the reachable space of a humanoid robot: A bio-inspired approach," *IEEE RAS and EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, 2012.

[8] J. Müller, U. Frese, and T. Röfer, "Grab a mug - Object detection and grasp motion planning with the Nao robot," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2012.

[9] T. Yoshikawa, "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[10] N. Vahrenkamp and T. Asfour, "Representing the robot's workspace through constrained manipulability analysis," *Autonomous Robots*, 2014.

[11] P. Kaiser, D. Gonzalez-Aguirre, F. Schültje, J. B. Sol, N. Vahrenkamp, and T. Asfour, "Extracting whole-body affordances from multimodal exploration," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2014, to appear.

[12] L. Han and N. Amato, *Algorithmic and Computational Robotics: New Directions in Algorithmic and Computational Robotics*, 2000, ch. A kinematics-based probabilistic roadmap method for closed chain systems, pp. 233–245.

[13] J. Garimort, A. Hornung, and M. Bennewitz, "Humanoid navigation with dynamic footstep plans," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.

[14] F. Burget, A. Hornung, and M. Bennewitz, "Whole-body motion planning for manipulation of articulated objects," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

[15] J. Pan, S. Chitta, and D. Manocha, "FCL: A general purpose library for collision and proximity queries," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2012.

[16] R. Smits, "KDL: Kinematics and Dynamics Library," http://www.orocos.org/kdl.