# Deformable Trees – Exploiting Local Obstacle Avoidance

**Matthias Behnisch, Robert Haschke, Helge Ritter, Michael Gienger**

**2011**

# Deformable Trees – Exploiting Local Obstacle Avoidance

Matthias Behnisch, Robert Haschke and Helge Ritter
Research Institute for Cognition and Robotics, Bielefeld University, Germany

Michael Gienger
Honda Research Institute Europe, Offenbach, Germany

*Abstract*— **This work describes a hybrid trajectory planning approach for redundant robots, combining a local obstacle avoidance control framework with a global sampling-based planning component. The complexity induced by the high dimensionality of the configuration space is counteracted by shifting the global search to a low dimensional task space representation. It is shown how an advanced autonomy of the controller to locally circumvent obstacles can be exploited to speed up global planning and to increase the robustness of planned trajectories against small obstacle disturbances.**

## I. INTRODUCTION

Sampling-based motion planning algorithms are extensively used to solve complicated problems, for an overview see e.g. [1]. Depending on the actual planning problem they can operate quite fast, but the effort increases with the volume of the space to be searched on an exponential scale. Humanoid robots typically have a complex redundant body structure with many joints and are thus posing a high-dimensional planning problem. Moreover they are meant to operate in cluttered and dynamic human environments, making efficient motion planning a challenge.

In order to deal with the problem of searching high dimensional spaces, some works try to foster the exploration of the space along a lower dimensional subspace that defines meaningful search directions with a higher importance to the task at hand. A few examples of these low dimensional spaces are given in [2] for high dimensional underactuated and dynamic systems. The algorithm presented in [3] does not predefine this space but tries to estimate the dimensions that capture the largest data variance with principal component analysis and focus the search along this directions.

Another body of literature directly deals with articulated robots like robot arms and humanoid robots [4]–[7]. Here the search is carried out in the configuration space of the robot and a task level inverse kinematics controller is used to bias the search towards the task goal position. One advantage is an effective speed up of the search. Another benefit is that it is no longer necessary to predefine a set of valid robot postures at the goal, since a valid inverse kinematics solution is computed simultaneously during the search.

In this paper, we follow a different approach. Instead of searching the high dimensional configuration space, it is also possible to shift the whole search to a low dimensional task space representation. In [8] this is shown to greatly improve the performance for planning trajectories for high dimensional robot arms, if the search is reduced to the task space of a Jacobian pseudo-inverse feedback controller, although the search can no longer be guaranteed to be complete in the full space of possible configurations.

In [9] we propose a hybrid planning method that combines a control framework, able to exploit redundancy to locally circumvent obstacles, with a complementary sampling-based planner, able to find globally feasible solutions. By utilizing the redundancy to avoid obstacles, a large fraction of relevant problems can be solved.

In this work the avoidance ability of the control framework is taken one step further by exploiting not only the redundant space but also the task space. The framework now has the freedom to depart from a given task trajectory in order to stay away from obstacles. This has the advantage of allowing the global planner to operate on a coarser scale. On the other hand however, situations where local avoidance control fails need to be handled. We propose to adaptively control the avoidance behavior during planning to deal with such cases. Also a measure of local trajectory stability is introduced to bias the search towards paths with high stability, leading to an increased robustness against disturbances of obstacle positions.

## II. HYBRID PLANNING

The motion planning approach presented in this paper builds on the concept of shifting the planning problem to a task level representation, described in section II-A. During the planning process, explained in section II-C, the global sampling component just specifies sub-goals for the whole body motion control framework, generating full configuration space trajectories based on the task level control input. Details of the control scheme are given in section II-B. This way, our method is hybrid in the sense of dividing the search into a global component, based on a sampling-based planning algorithm, and a local component, able to avoid obstacles with distance based potential functions.

### A. Task Space Representation

The objective of motion planning in this paper is to find a smooth movement trajectory for a redundant robot, starting at some initial posture and reaching a specified task position without collisions. As a compact representation of

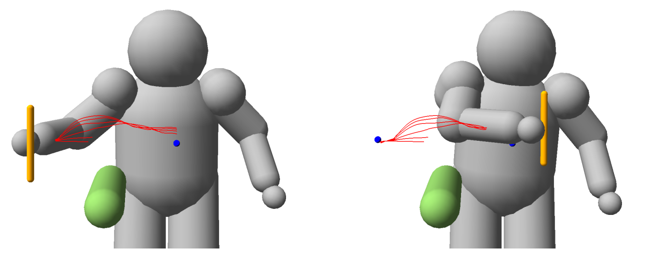

Fig. 1. Local obstacle avoidance with different weighting of target directed and obstacle avoidance motion.

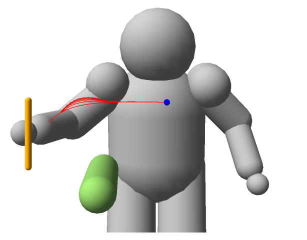

Fig. 2. For a different target position, all local trajectories with different weighting of target directed and obstacle avoidance motion converge to one point.

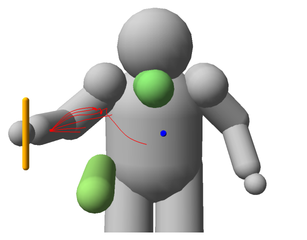

Fig. 3. Local obstacle avoidance does not always create a collision free trajectory.

the goal task position, an attractor point is placed in the task space and a second order dynamic is used to compute task space velocities [10]. A discrete time formulation of the dynamic equation is given in [11]. The dynamic system can be expressed with a continuous second order state transition function $\ddot{x} = h(x, \dot{x}, u(t))$, starting at $x$ with velocity $\dot{x}$. The control input $u(t)$ is a linear time dependent function, continuously interpolating the attractor point from its start position towards the target position.

Both the position and velocity are needed to fully determine the dynamics behavior, i.e. the domain of the dynamic process is the phase space $(x, \dot{x})$. It is not possible to describe the motion without specifying the velocity, which implies the presence of non-integrable differential constraints, also called non-holonomic constraints [1].

## B. Control Framework

The control framework is based upon the whole body motion control method presented in [10]. It solves the inverse kinematics problem following the well known principle of resolved motion rate control and redundancy resolution with the gradient projection method (e.g. see [12]). Given a desired task space velocity, the corresponding joint velocities are generated, while the kinematic redundancy is exploited in order to fulfill secondary motion objectives. Possible secondary motion objectives employed for whole body motion control framework are joint-limit avoidance [10] and collision avoidance [11], [13].

In this work we employ both criteria, joint-limit avoidance and collision avoidance. These criteria are locally optimized by gradient descent in two cost functions in the joint space. The joint-limit avoidance cost function $H_{limit}$ penalizes deviations of individual joints from a defined reference position with a quadratic function [12]. The collision-avoidance cost-function $H_{coll}$ penalizes both collisions of the robot with itself and collisions with external objects. For this all pairs of closest points between robot segments and between robot segments and external objects are computed. A quadratically increasing cost function associates higher costs for distances below a given safety threshold [11].

The contributions from these secondary cost functions can be projected into the redundant space, where they are not interfering with the primary target following motion. We propose to allow some deviations from straight task-execution, in order to gain more freedom that can be utilized to improve the optimization of the secondary motion objectives. For motion planning, preventing collisions is crucial and thus we choose to use the gained freedom for obstacle avoidance. However, the extend of influence of obstacle avoidance to the task space motion should not be fixed but can be smoothly varied.

The overall task space motion consists of a weighted superposition of the target directed task space velocity $\dot{x}_{tgt}$ and an obstacle avoidance velocity $\dot{x}_{av}$. The control equation

$$\dot{q} = J^{\#}(\alpha\dot{x}_{tgt} - \beta\dot{x}_{av}) - \gamma N(\nabla H_{limit} + \nabla H_{coll}) \quad (1)$$

relates task velocities $\dot{x} \in \mathbb{R}^m$ to joint velocities $\dot{q} \in \mathbb{R}^n$ with the weighted generalized pseudo-inverse $J^{\#} = W^{-1}J^T(JW^{-1}J^T)^{-1}$. The second term accounts for the redundant space movement and $N$ denotes the null space projection matrix $N = (1 - J^{\#}J)W^{-1}$.

While the target directed velocity is given by the attractor dynamic, the obstacle avoidance velocity is determined by projecting the avoidance cost function $H_{coll}$ to the task space as

$$\dot{x}_{av} = \nabla H_{coll}^T J^{\#} \quad (2)$$

Given the three components of the motion, target directed motion, task space avoidance and redundant space motion, the influence of each is weighted by the parameters $\alpha$, $\beta$ and $\gamma$, respectively.

The weighting of the redundant space motion with the parameter $\gamma$ is by design independent of the weighting of the task space motion, because the two spaces are orthogonal. Thus the actual value of this parameter is not critical. For the weighting of target tracking and obstacle avoidance however, this is not the case. Here the motions are both in the task space and thus directly influencing each other.

The challenge is to weight both parts to achieve the potentially contradicting goals of getting closer to the target and staying away from obstacles. We propose to control the behavior by setting the parameter $\beta$. If it is set to 1 there is a large emphasis on obstacle avoidance while if it is set to 0 pure target following without obstacle avoidance is done. The weighting of target directed motion $\alpha$ is coupled to the value of $\beta$ and the magnitude of the obstacle avoidance gradient, i.e. the desired motion away from obstacles:

Algorithm 1. Incremental Tree Growing

```
tree $T$ with nodes $n_i \in (x, \dot{x}, x_{tgt}, t, q)$, weights $w_i$
while not goal reached and not time over do
    $n_s$ = node selection($T$)
    $x_{tgt}, \beta_{tgt}$ = sampling($n_s$)
    for all $\beta_j \in \{\beta_{tgt} \pm [k\delta], k = 0, 1, 2, ...\}$ do
        $n_j$ = local planner($n_s, x_{tgt}, \beta_j$)
    $w_{st}$ = stability heuristic($\{n_j\}$)
    $T$ = add to tree($T, n_0, w_{ex} \cdot w_{st}$)
```

Algorithm 2. Stability Heuristic

```
child nodes $C = \{n_i | i = 1...N\}$
for all $n_i \in C$ do
    $d_i = \frac{1}{N-1} \sum_{j=1, j \neq i}^{N}$ distance($n_i, n_j$)
stability $= 1/\exp(\sum_{i=1}^{N} d_i)$
```

$$\alpha = \frac{\beta}{\exp\left(s\|\nabla H_{coll}\|\right)} + 1 - \beta \tag{3}$$

If no avoidance gradient is present $\alpha$ is 1 and the weight is completely on target directed motion. With an increasing gradient, the value of $\alpha$ is reduced to emphasis avoidance motions. $\beta$ is used to set he baseline of this decay to $\alpha = 1 - \beta$, to guarantee a certain remaining amount of task following even if the gradient is large. Thus the adaptive weighting of the local obstacle avoidance behavior is controlled by a single scalar parameter $\beta$ that smoothly varies between the extremes of dominant obstacle avoidance and dominant target following. Figures 1 and 2 show example trajectories in the presence of an obstacle, for different values of this parameter.

This scheme can dynamically adapt to the complexity of the scene, but by the nature of this approach, the target is not reached in conflicting situations. It is quite easy to arrange obstacles such that they can not be circumvented. If a certain situation can be handled with a specific parameter value, the same value does not generalize necessarily to different situations. See figure 3 for an example. Another problem is that the movement can get stuck in local minima because only local gradient information is used. Thus it is necessary to augment the control framework with a global search component to select the right weighting parameter and proper sub-targets to escape global minima.

### C. Sampling-based Planning

Global planning is done with a sampling-based single-query tree planner. The general operation scheme is to select an existing tree node first and then sample a new state in a second step. The selected tree node is extended towards the sample using a local planner and by iterating this process an incremental tree is grown. A number of algorithms operate with this scheme, differing in how nodes are selected and samples are created [14]–[16]. They can handle non-holonomic movement constraints, which arise

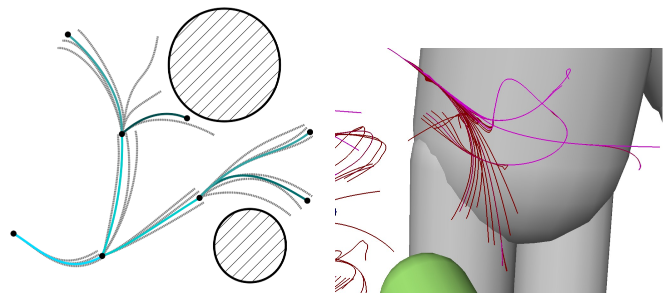

Fig. 4. View of the search tree, conceptual (left) and in simulation (right). From every expanded node a set of local trajectories is generated, which sample various $\beta$ values weighting target directed vs. collision avoidance motions. Trajectories converging to similar task-space regions are preferred for further exploration as they are more robust w.r.t. the weighting parameter and thus to a variation of obstacle positions. More robust trajectories are drawn with brighter color.

from the attractor dynamic task representation in section II-A. Also it is easy to incorporate custom search heuristics by simply changing the weighting function of the tree nodes.

In our hybrid planning approach the space actually searched with sampling is the task space introduced in section II-A, with the task trajectories represented in the task phase space. But as already mentioned, the whole body motion generated by the control framework is still represented in the full configuration space of the robot. Thus the individual nodes of the search tree contain not only the current task space target $x_{tgt}$, trajectory duration time $t$ and the initial phase space position $(x, \dot{x})$, but also the corresponding initial configuration space position $q$.

Four steps are done in each iteration, as shown in algorithm 1:

*1) Node Selection:* Like in the Expansive Space Tree (EST) [14], each node has an associated weight $w_{ex}$ estimating the extend of exploration already done in its neighborhood. It is set to be inversely proportional to the density of nodes in its neighborhood, thus favoring regions with fewer nodes to achieve an uniform coverage of the space.

A second objective of node selection, added in our approach, is to favor paths of higher stability. This property is estimated for each path during local planning and a corresponding weight $w_{st}$ is associated with the resulting end point node. The overall probability of selecting a node $i$ is proportional to the combination of both weights, $w_i = w_{ex} \cdot w_{st}$.

*2) Sampling:* In the EST algorithm a new sample point is created in the neighborhood of the selected node. When planning is done in the task space, this corresponds to sampling of a new task space attractor point $x_{tgt}$. The behavior of the local obstacle avoidance has to be controlled by setting the weighting parameter of the control framework $\beta_{tgt}$. This parameter is randomly sampled initially and systematically varied during local planning. In order to focus the search towards the goal, some goal biasing is done by not sampling a random point at certain times but instead directly trying to reach the goal.

*3) Local Planning:* The local planning step is done with the described control framework. Given a starting point from the node selection $n_s$ and a task space target $x_{tgt}$ from the sampling, a local trajectory is created. The local planner is called multiple times here to create a set of local trajectories with a different weighting of target directed and obstacle avoidance motions. The weighting is determined by the parameter $\beta_j$, systematically varied in a certain range of values. This set of local trajectories is only created to be used for computing the stability heuristic, only the trajectory with the sampled $\beta_{tgt}$ is added to the tree.

*4) Stability Heuristic:* In this step the set of trajectories with different weighting of target directed and obstacle avoidance motions is used to determine the local stability of the control system, as illustrated in figure 4. Stability is defined here as the independence of the systems convergence point to changes of the weighting parameter $\beta$. Motivation for this is that a trajectory that does not change much if the weighting is changed, also does not change much if the obstacle position is changed. Thus the robustness of the controller against disturbed obstacle positions due to imprecise sensor information or small movements is increased.

The computation of a corresponding stability heuristic is shown in algorithm 2. The distance between a given trajectory endpoint towards all other endpoints is averaged and based on this the stability is estimated to be low for large distances and high for small distances.

## III. EVALUATION

The proposed hybrid motion planning method is implemented within the OOPSMP library [17], [18]. Collision checking and distance computations for the simple geometric primitives of spheres, boxes and sphere swept lines are done by the Vortex engine [19]. To evaluate the method, a simulation study and an experiment with a humanoid robot are carried out.

### A. Simulation

Figure 5 and 6 show two simulation setups, together with an example search tree and solution path. In both cases the sampled task space represents the position and orientation of the hand. The orientation is described as the position of the grasp axis – the axis enclosed by the closed hand. The angle around this axis is not constrained with this description. Thus the task space that contains the positions and orientations has one dimension less than a full position and orientation representation and is given by the 5 dimensional manifold $T = \mathbb{R}^3 \times \mathbb{S}^2$.

While the target position in the first setup (figure 5) is given in absolute coordinates, the bi-manual task of the second setup (figure 6) is defined in relative coordinates. The position of the right hand is given relative to the position of the left hand and this way a target position of the flower inside the basket can be defined independently of the actual position of the basket.

To show the impact of the task space obstacle avoidance, different strategies for the selection of the avoidance parameter $\beta$ are compared. The first case is a constant weighting value of 0, disabling avoidance in the task space completely, only the redundant space is used. In the second condition, the weighting factor is randomly sampled in the interval from 0 to 1. In the last condition finally, the stability heuristic in the fourth step of the algorithm is enabled, biasing the solution towards paths with higher stability.

The stability bias heuristic was introduced under the assumption that trajectories with stable convergence points

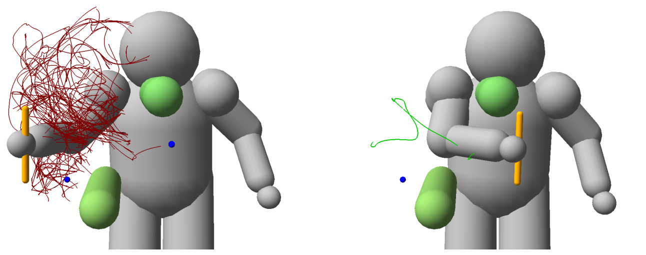
Fig. 5. First simulation setup: Holding a rod (right hand), the task is to reach from left to right with two cylindrical obstacles in the way.

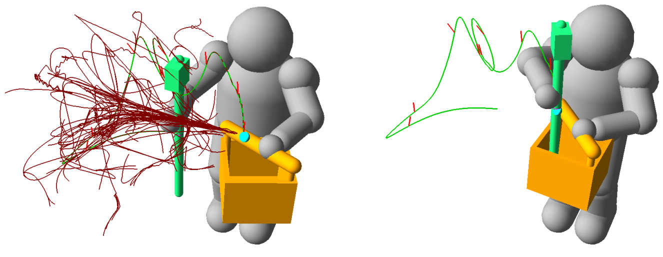
Fig. 6. Second simulation setup: The task is to move the flower (right hand) into the basket (left hand).

| avoidance weight $\beta$ | success rate | solution time sec. ($\sigma$) | tree size ($\sigma$) |
|---|---|---|---|
| 0 | 0.9 | 31.02 (25.53) | 8144 (6018) |
| random | 1.0 | 20.31 (22.91) | 4726 (4779) |
| stability heuristic | 0.64 | 28.60 (28.28) | 8928 (7737) |

TABLE I
RESULTS FOR THE FIRST SIMULATION SETUP.

| avoidance weight $\beta$ | success rate | solution time sec. ($\sigma$) | tree size ($\sigma$) |
|---|---|---|---|
| 0 | 0.86 | 29.56 (26.98) | 10271 (8728) |
| random | 0.95 | 27.37 (26.41) | 8718 (7848) |
| stability heuristic | 0.1 | 38.50 (28.99) | 28842 (10850) |

TABLE II
RESULTS FOR THE SECOND SIMULATION SETUP.

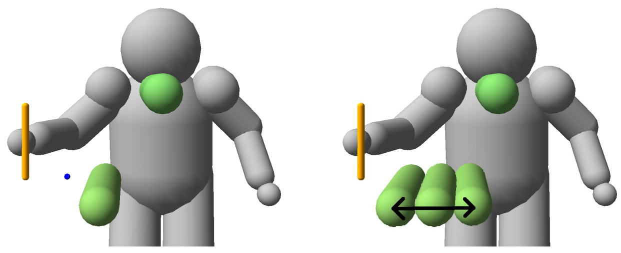

Fig. 7. Systematic displacement of the lower obstacle in horizontal direction.

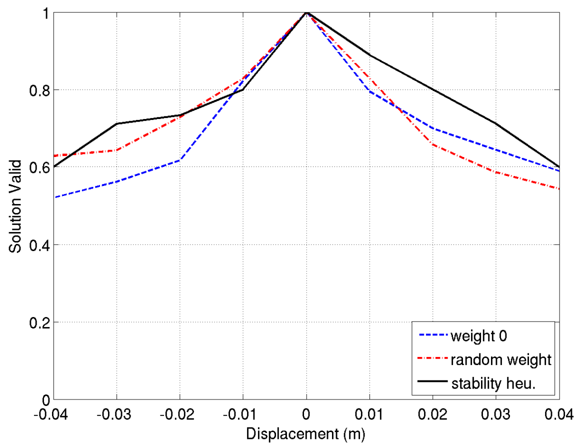

Fig. 8. Fraction of solution trajectories reaching the goal for different amounts of obstacle displacement.

are less influenced by disturbances of obstacle positions. To check to which extend this argument holds, one obstacle in the first setup is systematically displaced, as shown in figure 7. For each obstacle positions all solution trajectories are checked whether they are still reaching the goal and this fraction is plotted against the amount of displacement in figure 8.

### B. *Experiment*

In addition to the simulation, a robot experiment demonstrates the feasibility of our method for real world motion planning in human like environments. Basically the same software system as in the simulation is used. A Vicon motion capturing system [20] with 8 cameras tracks the position of the robot, the flower and the basket. The setup is subject to positioning errors of the objects relative to the robot, caused by tracking errors and inherent motion execution errors of the robot. Another source of changes of object positions during the motion is possible slipping of the objects inside the robots hand. For this reasons, we can benefit from the advanced obstacle avoidance capabilities of our method here. Figure 9 shows the setup and an example solution motion sequence.

## IV. DISCUSSION

Quantitative results for the simulation study are summarized in table I and II, for the three different avoidance weighting conditions. The columns show the fraction of planning runs that were successful in the given time limit (120 sec.), the average time needed for a solution and the tree size of expanded nodes. All results are averaged over 100 runs. Comparing the cases of $\beta = 0$ and random sampled $\beta$, both simulations show that for the latter the success rate is higher and the average computing time for a solution is lower. This clearly indicates that allowing avoidance motions in the task space indeed helps the planner to find a solution. The gain is more evident for the first setup where a significant part of any solution trajectory lies close to obstacles. In the second setup, only the last part of the motion is actually constrained by obstacles, thus the obstacle avoidance has less chance of being helpful and the speed difference is smaller.

Looking at he condition with the stability heuristic, the success rate and average runtime is larger. This is expected because each tree extension has a larger cost, since multiple trajectories with different $\beta$ are computed for a single tree extension. For this reason the tree size is also larger.

The positive impact of the stability heuristic can be seen for the obstacle disturbance test in figure 8. Without any task space avoidance motions (weight 0) the fraction of valid solutions decreases faster than for allowing avoidance motions (random weight). Also there is an advantage in favor of using the stability heuristic visible, especially in positive displacement direction (stability heu.).

No quantitative results are given for the actual robot experiment. To give a hint of the power of our method we can state that no collision occurred during the whole experiment consisting of around 10 trials.

## V. CONCLUSION

A hybrid trajectory planning approach for redundant robots is described in this work. It combines a global sampling-based planner with a local control framework, able to optimize motions regarding joint-limit avoidance and collision avoidance. The local collision avoidance capability is enhanced by allowing deviations from a strict target directed trajectory. This ability can be smoothly varied by a weighting parameter in order to deal with situations where the objectives of reaching the goal and circumventing obstacles are in conflict.

It is shown that the enhanced avoidance behavior can be exploited for planning. Depending on the setup, planning succeeds earlier and less computation time is needed. Also the local stability of trajectories can be estimated and used during planning to bias the search towards solutions that are more robust against disturbances of obstacles.

## ACKNOWLEDGMENTS

Matthias Behnisch gratefully acknowledges the financial support from Honda Research Institute Europe for the project ”Sampling-based planning for goal directed robot control”.

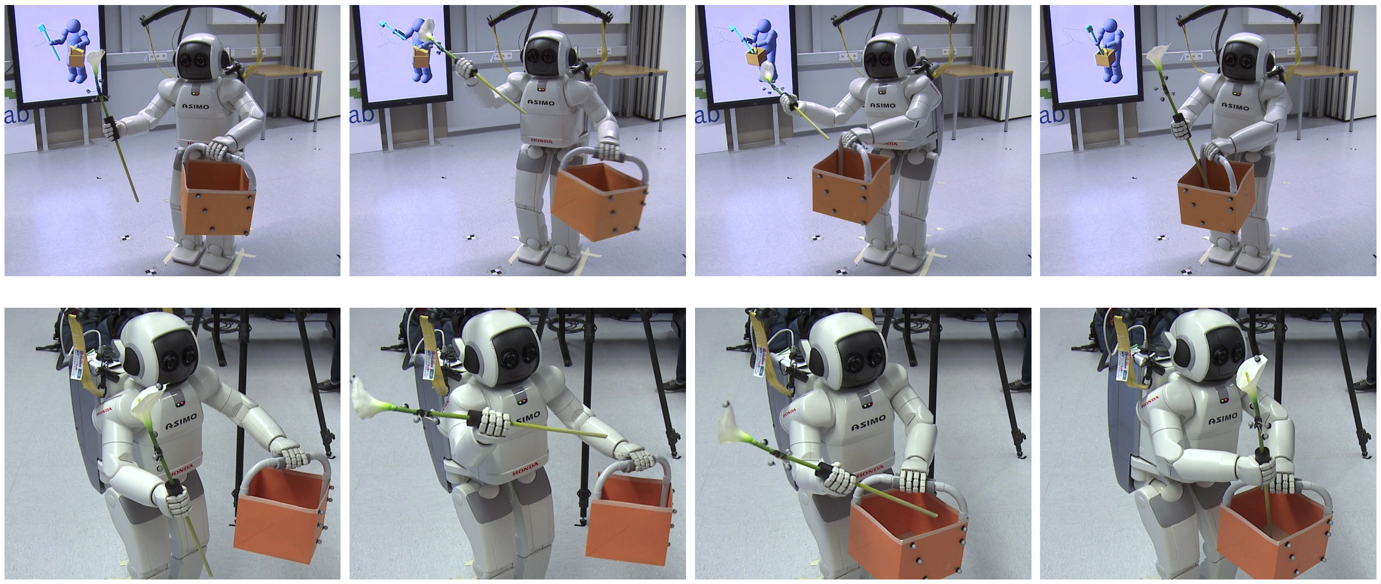

Fig. 9. Experimental setup with the humanoid robot. The task is to put the flower grasped with the right hand into the basket in the left hand. Prior to planning, the goal position of the flower is given by a human operator by holding the flower at its desired position into the basket and by saving its tracked position. Tracking is done with the Vicon motion capturing system, which markers can be seen on flower and basket.

## References

[1] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[2] I. Sucan and L. Kavraki, "Kinodynamic motion planning by interior-exterior cell exploration," in *Algorithmic Foundations of Robotics VIII: Selected Contributions of the Eighth International Workshop on the Algorithmic Foundations of Robotics*, vol. 57, 2009, p. 449.

[3] S. Dalibard and J. Laumond, "Control of probabilistic diffusion in motion planning," *Algorithmic Foundation of Robotics VIII*, pp. 467–481, 2009.

[4] Y. Yang and O. Brock, "Elastic roadmaps: Globally task-consistent motion for autonomous mobile manipulation in dynamic environments," in *Proceedings of Robotics: Science and Systems*, 2006.

[5] D. Bertram, J. Kuffner, R. Dillmann, and T. Asfour, "An integrated approach to inverse kinematics and path planning for redundant manipulators," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

[6] M. V. Weghe, D. Ferguson, and S. Srinivasa, "Randomized path planning for redundant manipulators without inverse kinematics," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2007.

[7] R. Diankov, N. Ratliff, D. Ferguson, S. Srinivasa, and J. Kuffner, "Bispace planning: Concurrent multi-space exploration," in *Proceedings of Robotics: Science and Systems IV*, 2008.

[8] A. Shkolnik and R. Tedrake, "Path planning in 1000+ dimensions using a task-space voronoi bias," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2009.

[9] M. Behnisch, R. Haschke, and M. Gienger, "Task space motion planning using reactive control," in *Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems*, 2010.

[10] M. Gienger, H. Janssen, and C. Goerick, "Task-oriented whole body motion for humanoid robots," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2005.

[11] M. Toussaint, M. Gienger, and C. Goerick, "Optimization of sequential attractor-based movement for compact movement representation," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2007.

[12] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley Publishing Co., Inc., 1991.

[13] H. Sugiura, M. Gienger, H. Janssen, and C. Goerick, "Real-time self collision avoidance with whole body motion control for humanoid robots," in *Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems*, 2007.

[14] D. Hsu, R. Kindel, J. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *The International Journal of Robotics Research*, vol. 21, no. 3, p. 233, 2002.

[15] J. Phillips, N. Bedrossian, and L. Kavraki, "Guided expansive spaces trees: a search strategy for motion-and cost-constrained state spaces," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004.

[16] B. Burns and O. Brock, "Single-query motion planning with utility-guided random trees," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 3307–3312.

[17] E. Plaku, K. Bekris, and L. Kavraki, "Oops for motion planning: An online, open-source, programming system," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007.

[18] OOPSMP. [Online]. Available: http://www.kavrakilab.rice.edu/

[19] CM Labs Vortex. [Online]. Available: http://www.vxsim.com/

[20] Vicon. [Online]. Available: http://www.vicon.com/