Hybrid Planning: Task-Space Control and Sampling-Based Planning

Robert Haschke

Abstract— We propose a hybrid approach to motion planning for redundant robots, which combines a powerful control framework with a sampling-based planner. We argue, that a suitably chosen task controller already manages a huge amount of trajectory planning work. However, due to its local approach to obstacle avoidance, it may get stuck in local minima. Hence, it is augmented with a globally acting planner, which now can operate in a lower-dimensional search space, thus circumventing the curse of dimensionality pertaining for modern, many-DoF robots.

I. INTRODUCTION

Modern two-handed service robots pose enormous challenges to planning and control, especially because they operate in highly cluttered and dynamic environments next to humans, thus demanding efficient, online and real-time capable motion planning and control algorithms. While there exist powerful sampling-based planning methods, which can solve complicated problems (for an overview see [1]), they typically suffer from the curse of dimensionality: the planning effort increases exponentially with the number of degrees of freedom. Modern two-handed, multi-fingered robots easily have more than 50 DoFs, rendering these approaches infeasible for real-world applications.

In order to deal with this complexity, most approaches decompose planning into independent subproblems. For example – considering grasping – a pre-determined database of grasps for a given object is used to relax the need to plan for the hand motion [2], [3]. Given a set of feasible grasps from this database, the planning can be restricted to the motion of the end effector to reach appropriate hand poses. This planning step is further subdivided into placement of the robot base and subsequent arm motion. However, due to their complexity and their need for pre-computed task knowledge, these algorithms are not yet deployable in unstructured and dynamic environments.

On the other hand, there exist powerful control-based methods, especially the control basis framework of Grupen et al. [4], [5], which provide online-capable approaches to planning and motion generation utilizing gradient-based optimization of suitable cost functions (to reach the object, establish contact, maximize grasp stability, and avoiding obstacles and joint limits). Obviously, those methods – due to their local approach – can be trapped in local minima.

The central idea of our work is the integration of local control and global planning into a hybrid approach, which tries to exploit the advantages while avoiding the drawbacks of

both worlds: Sampling-based planning, which acts globally, is restricted to a low-dimensional, well-suited task-space. This dramatically reduces the search space [6], but also restricts the amount of feasible solutions. To counteract this negative effect, an intelligent local control method exploits the redundancy in the task's null space to increase the success rate of motions between randomly sampled via points.

This share of work between local control and global planning shall allow the global planner to operate on a coarser scale, thus speeding up the overall planning process.

II. HYBRID PLANNING

In the following, we first outline the capabilities of task-space control methods, secondly summarize the expansive space tree approach, which we employ for sampling-based planning, and finally introduce the hybrid approach itself.

A. Task Space Control

Task-space control methods are founded on the fact, that there exists a (locally) linear correlation of joint movements $\dot{\mathbf{q}}$ and corresponding velocities $\dot{\mathbf{x}}$ of task-space coordinates, which can be easily inverted using the pseudo-inverse of the describing Jacobian $J(\mathbf{q})$ [7]:

$$\dot{\mathbf{q}} = J^+(\mathbf{q}) \cdot \dot{\mathbf{x}}. \tag{1}$$

In order to deal with numerical instabilities in the vicinity of singularities, several approximative methods were proposed, including singular value decomposition and damped least squares [7]. Redundancy induced by a smaller number of task-space dimensions compared to the number of joints can be exploited to maximize an arbitrary function H. To this end, the gradient $\nabla_{\mathbf{q}} H(\mathbf{q})$ is projected to the null space of J to limit the motion to the redundant space [8]:

$$\dot{\mathbf{q}} = J^+ \cdot \dot{\mathbf{x}} + N \cdot \nabla_{\mathbf{q}}^t H(\mathbf{q}), \tag{2}$$

where $N(\mathbf{q}) = \mathbf{1} - J^+ J$ is the null space projector of J. The main idea of the control basis framework (CBF) [4] is to assume, that the null space motion is generated by a subordinate task controller, J_2, which recursively applies the gradient projection method (2), thus composing complex controllers from simpler ones in an hierarchical fashion:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_1 + N_1(\dot{\mathbf{q}}_2 + N_2(\mathbf{q}_3 + \cdots))$$
$$= J_1^+ \dot{\mathbf{x}}_1 - N_1(J_2^+ \dot{\mathbf{x}}_2 + N_2(J_3^+ \dot{\mathbf{x}}_3 + \cdots)). \tag{3}$$

By choosing suitable task representations, one can generate naturally looking, smooth movements in a simple fashion. A major drawback of nowadays motion planning approaches is their attempt to fully specify the end-effector pose in 6D. However, many tasks – due to their inherent symmetry – do
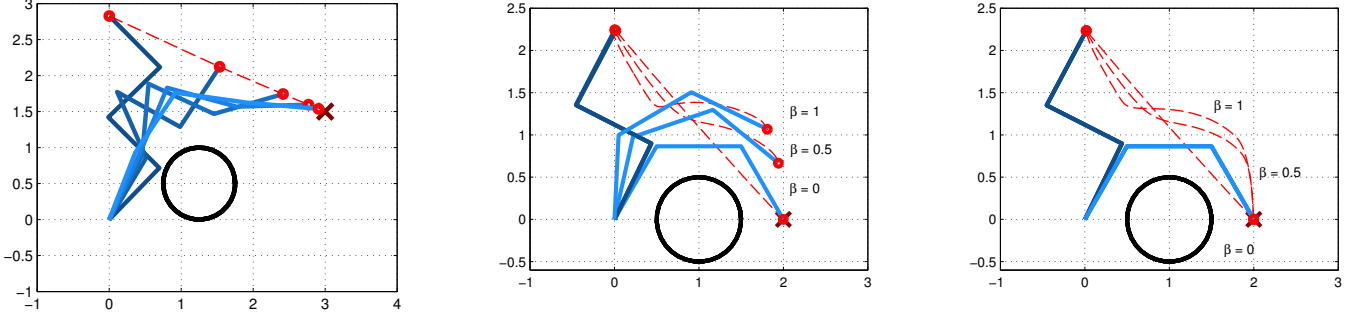
Fig. 1. Goal-directed task-space motion with collision avoidance. Left: Restricting avoidance motions to redundant space yields a straight line motion of the end effector. Middle: Using relaxed motion control (8), the trajectory more strongly avoids the obstacle for larger weights β, but does not converge to the target anymore. Right: Dynamic adaption of β achieves both goals, target reaching and obstacle avoidance (in this example).

not require this. For example, grasping a cylindrical object, like a bottle, only requires to align the hand axis with the object axis, the orientation angle around this axis can freely be chosen [9]. To allow even for flexibility, one may specify a task-space interval instead of a unique target [10]. Platt et al. propose even more abstract controllers, e.g. to maintain force closure, to optimize grasp quality, manipulability, or visibility [5].

Our own implementation of CBF [11], further allows to compose more complex tasks from simpler ones, by (i) stacking Jacobians (solving multiple tasks simultaneously with equal priority), (ii) subtraction of Jacobians (solving relative motion tasks, e.g. left relative to right hand), and (iii) adapting the Jacobian, for example to control the mere distance to a target, i.e.

$$J' = (\mathbf{x} - \mathbf{x}_{\text{goal}})^t \cdot J \qquad (4)$$

In the latter case, the task-space motion $\dot{\mathbf{x}}$ is a straight-line towards the goal, much like in classical Cartesian control. However, the redundant space at a given goal distance is the complete sphere around the target and any null space motion is automatically projected onto this sphere. In this manner, we can easily approach spherical objects for grasping from any direction, without the need to precompute a multitude of feasible grasps in advance.

B. Local Collision Avoidance

In the context of motion planning, an important subordinate optimization criterion to be applied in the redundant space is of course collision and joint limit avoidance. Joint limits can be easily avoided minimizing a quadratic or higher-order polynomial function [8], [9]:

$$H_{\mathbf{q}} = \sum w_i \, (q_i - q_i^{\text{ref}})^p \qquad w_i = (q_i^{\text{max}} - q_i^{\text{min}})^{-1}, \quad (5)$$

where $\mathbf{q}^{\text{ref}}$ defines a reference pose, e.g. in the middle of the joint range, and the w_i's weight the contribution of individual joints according to their overall motion range.

Local collision avoidance is achieved by a repelling force field originating from each object. To this end, Sugiura [12] proposes to minimize a quadratic cost function defined on the distance $d_{\mathbf{p}} = \|\mathbf{p}_1 - \mathbf{p}_2\|$ between the two closest points $\mathbf{p}_1$ and $\mathbf{p}_2$ on the robot and the obstacle:

$$H_{ca}(\mathbf{p}_1, \mathbf{p}_2) = \begin{cases} \eta \, (d_{\mathbf{p}} - d_B)^2 & d_{\mathbf{p}} < d_B \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

Here, d_B acts as a distance threshold below which the force field becomes active and η is a gain parameter. If active, the cost gradient can be computed in terms of the body point Jacobians by applying the chain rule:

$$\nabla_{\mathbf{q}}^t H_{ca} = 2\eta \, (1 - d_B/d_{\mathbf{p}})(J_{\mathbf{p}_1} - J_{\mathbf{p}_2})^t (\mathbf{p}_1 - \mathbf{p}_2), \quad (7)$$

which is easily formulated in the control basis framework. If we employ this cost function for Eq. (2), we yield straight-line task-space movements (e.g. of the end-effector in Cartesian space), while the redundancy is exploited to circumvent obstacles as schematically shown in Fig. 1, left.

To allow more flexible obstacle avoidance, in [13] we proposed a relaxed motion control scheme, which allows deviations from straight-line motions, if the robot gets too close to obstacles:

$$\dot{\mathbf{q}} = J^+ (\dot{\mathbf{x}} - \beta \, \dot{\mathbf{x}}_{ca}) - N(\nabla H_{ca} + \nabla H_{\mathbf{q}}) . \qquad (8)$$

Here, additionally to the null-space motion, which minimizes a superposition of both cost functions $H_{\mathbf{q}}$ and H_{ca}, an obstacle avoidance motion $\dot{\mathbf{x}}_{ca}$ directly occurs in task-space as well. This velocity is determined by projecting the cost gradient (7) to the task space:

$$\dot{\mathbf{x}}_{ca} = J \nabla_{\mathbf{q}}^t H_{ca} . \qquad (9)$$

Choosing different values of the weight β, we can smoothly adjust the importance of collision avoidance and target reaching as shown in Fig. 1, middle. However, because both contributions might be contradicting, the target is not always reached with $\beta > 0$. To prevent this, we can ensure, that the goal-directed motion always dominates the collision avoidance motion with a margin ε, if we dynamically adapt β, such that the following condition is fulfilled:

$$\|\dot{\mathbf{x}}\| - \varepsilon \geq \beta \|\dot{\mathbf{x}}_{ca}\| . \qquad (10)$$

---

while goal not yet reached **do**
  if goal bias **then**
    $\mathbf{p}$ = tree node closest to target $\mathbf{x}_{\text{goal}}$
    $\mathbf{x}_{\text{tgt}} = \mathbf{x}_{\text{goal}}$
  else
    $\mathbf{p}$ = tree node selected according to weights w_i
    $\mathbf{x}_{\text{tgt}}$ = task-space via point sampled in vicinity of $\mathbf{p}$
  end if
  $(\bar{\mathbf{x}}, \bar{\mathbf{q}}, t, Q)$ = final motion state of local controller
  add tree node **if** $t_{\min} < t < t_{\max}$
end while

The resulting motion is shown in Fig. 1, right. However, as collision avoidance is the more important objective, it is acceptable to miss the intermediate target. The sampling-based planning component, described in the next subsection, will accommodate for this by globally guiding the search process, providing new via points.

C. Sampling-based Planning

Sampling-based methods randomly grow a tree to explore the whole search space, starting from the initial pose and eventually reaching the targeted pose. The most prominent method, RRT [14], biases its search towards unexplored regions, thus rapidly exploring the whole space. However, we prefer the family of expansive space tree algorithms (EST) [15], because they allow to bias the search in a more fine-grained fashion employing various heuristics. In contrast to the RRT algorithm, EST switches the order of state sampling and tree node selection, performing the following sequence of operations to incrementally grow the tree:

1) randomly select a tree node $\mathbf{p}$
2) sample a new state / via point $\mathbf{x}_{\text{tgt}}$ in vicinity of $\mathbf{p}$
3) extend $\mathbf{p}$ towards $\mathbf{x}$ using a local planner

While sampling-based methods often directly operate on the joint space (to maximally cover the search space), the proposed hybrid planning approach shifts planning to a low-dimensional task-space representation and exploits the powerful task-space controller described in section II-B for local tree extensions. Hence, extensions are more often successful, reducing the need for extensive local refinement of the search tree. Consequently, our sampling-based approach focuses on rapid and coarse-scale exploration of global connectivity. The individual steps of the algorithm and the proposed sampling heuristics are outlined in the following.

1) Node Selection. The major advantage of EST compared to RRT is the possibility to determine, which tree node should be extended next. Plaku et al. bias tree growth towards less covered regions by more frequently choosing tree nodes for extension which have fewer outgoing edges [16]. Assuming a uniform distribution of edge directions and lengths, this yields a reasonable local coverage estimate. However, employing the nontrivial local planner, node extensions more frequently follow similar paths or fail in heavily

cluttered environments, because obstacles are avoided in a similar fashion. In this case, node selection should avoid nodes, whose extensions were less successful.

Fortunately, local planning provides various, nontrivial success measures for a node extension, which can be exploited for this additional biasing of the selection process. An important indicator for the presence of obstacles close to the path, is the accumulated magnitude of collision costs: $\mathcal{C} = \int H_{ca}$. However, this measures doesn't account for the direction of the repelling force field. A path should be only considered "difficult", if the costs increase towards the target, i.e. when the goal-direction motion $\dot{\mathbf{x}}$ and the collision avoidance motion $\dot{\mathbf{x}}_{ca}$ are counteracting. In this case the dot product of both vectors becomes negative, leading to the following quality criterion: $Q = \int \dot{\mathbf{x}} \cdot \dot{\mathbf{x}}_{ca}$.

2) State Sampling. In order to optimally cover the local free space in the neighborhood of a tree node $\mathbf{p}$, we propose to apply a sampling strategy which reduces local dispersion [17]. In order to focus the search towards the goal, we apply goal biasing occasionally. To this end, the tree node closest to the target is extended towards the goal. If the local motion controller succeeds to reach the target, we are done. If a node was unsuccessfully used for goal biasing before, the next closest node is used, thus preventing the goal biasing to use the same node over and over.

3) Local Planning. The local motion controller, used to connect a tree node to a newly sampled via point, is limited in duration ($t_{\max}$) to avoid convergence problems in cluttered environments. The local planner returns the reached task-space and joint-space positions $\bar{\mathbf{x}}$ and $\bar{\mathbf{q}}$ of the initial portion of the trajectory, which obeys both joint limit and collision constraints. Additionally, the elapsed control time t and the integrated path quality measure Q is returned.

Finally, the reached state is added as a new node to the tree, if the elapsed control time is between $t_{\min}$ and $t_{\max}$, i.e. if a sufficient path-length could be reached and the controller converged in time. The overall algorithm is summarized in Alg. 1.

D. Task Motion Generation and State Representation

So far, we didn't considered the important aspect of task-space motion generation, i.e. computation of task-space velocities $\dot{\mathbf{x}}$ towards the target. In the past we have employed an algorithm to compute smooth, time-optimal trajectories
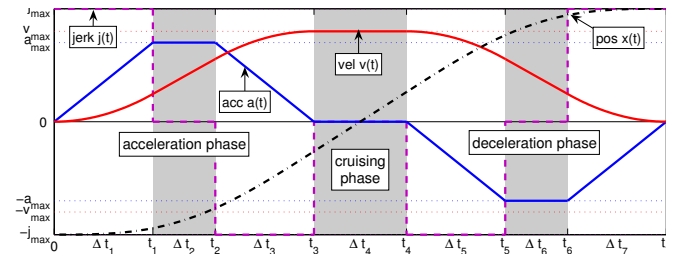

Fig. 2.   Time-optimal third-order trajectory profile consisting of seven phases corresponding to maximal jerk (pink), acceleration (blue), and velocity (red) application.
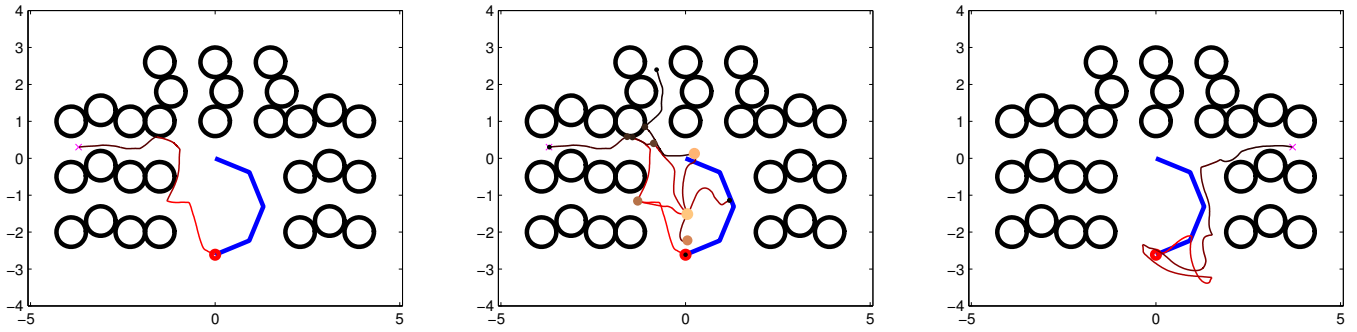
Fig. 3. Motion planning results for a planar 4-joint manipulator moving the end effector (red dot) towards the pink-colored cross. Left and right images show two example trajectories. Middle image illustrates the search tree, with larger and brighter nodes indicating higher exploration weights.

obeying limits on velocity, acceleration and jerk [18]. To this end, a motion trajectory is composed from cubic splines, partitioning the trajectory into phases of applying maximum velocity, acceleration, or jerk as illustrated in Fig. 2. But, the analytic approach involves the computation of zeros of a fourth-order polynomial, which may be ill-posed in certain conditions. Kröger solved this issue with a carefully designed Newton-Raphson algorithm [19].

However, in motion generation, especially for humanoids, it is not important to obtain *the* optimal solution, but it suffices to gain a *very good* one. Hence, we adopt the dynamical-systems approach proposed in [20], [21] utilizing a second-order attractor dynamics (spring-damper system) driving the task-space motion towards the target attractor in a smooth fashion obeying coarse motion limits on velocity, acceleration and jerk:

$$\ddot{\mathbf{x}}(t) = k(\mathbf{x}_{\text{tgt}}(t) - \mathbf{x}(t)) - \gamma\dot{\mathbf{x}}(t), \tag{11}$$

where k and γ denote the spring and damping constants respectively. Due to its similarity to dynamic motion primitives (DMP) [22], which adds an additional external force $f(t)$ to modulate the shape of the trajectory, it can be easily adapted to imitation learning tasks as well.

According to Eq. 11, the overall state information, which needs to be stored in each tree node, comprises the task-space coordinates $\mathbf{x}$, their velocities $\dot{\mathbf{x}}$, as well as the corresponding joint-space pose $\mathbf{q}$. The latter is required to resolve the redundancy when continuing the search from a specific tree node. That is, although sampling and thus growing of the search tree is performed in task-space primarily, a corresponding tree also exists in joint-space.

III. RESULTS AND DISCUSSION

In our publications [13], [23] we demonstrated, that the hybrid planning approach finds solutions more often and with fewer tree extensions. However, the more complex local controller takes much more time, such that the overall speedup is limited. Fig. 3 shows exemplary results of a planar 4-joint manipulator moving its end effector (red dot) towards the pink-colored, cross-marked target. The relaxed motion control scheme results in deformed trajectories to better avoid obstacles. Compared to a control scheme, whose

avoidance capabilities are limited to the redundant space, relaxed motion control has a higher success rate in complicated situations and takes fewer iterations.

The critical issue of the hybrid planning approach is how the share of workload between the local and global planning is accomplished. Our preliminary results are encouraging, but there is still room for further improvements devising good heuristics for tree node selection and via-point sampling. Summarizing, the shift from perfect towards near-optimal approaches is very promising to realize real-time motion control and planning algorithms for real-world application in many-DoFs, redundant robots.

REFERENCES

[1] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
[2] R. Diankov, N. Ratliff, D. Ferguson, S. Srinivasa, and J. Kuffner, "Bispace planning: Concurrent multi-space exploration," in *Proc. Robotics: Science and Systems IV*, 2008.
[3] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, "Humanoid motion planning for dual-arm manipulation and regrasping tasks," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2009, pp. 2464–2470.
[4] S. Hart, S. Sen, S. Ou, and R. Grupen, "The control basis API – a layered software architecture for autonomous robot learning," in *2009 Workshop on Software Development and Integration in Robotics at ICRA*, 2009.
[5] R. Platt, A. Fagg, and R. Grupen, "Null-space grasp control: theory and experiments," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 282–295, 2010.
[6] A. Shkolnik and R. Tedrake, "Path planning in 1000+ dimensions using a task-space voronoi bias," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2009.
[7] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley Publishing Co., Inc., 1991.
[8] A. Liegeois, "Automatic supervisory control of configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, no. 12, pp. 861–871, 1977.
[9] M. Gienger, M. Toussaint, and C. Goerick, "Whole-body motion planning -- building blocks for intelligent systems," in *Motion Planning for Humanoid Robots*, K. Harada, E. Yoshida, and K. Yokoi, Eds. Springer London, 2010, pp. 67–98.
[10] M. Gienger, H. Janßen, and C. Goerick, "Exploiting task intervals for whole body robot control," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2006, pp. 2484–2490.
[11] F. Schmidt, "Control basis framework," 2011. [Online]. Available: https://github.com/fps/CBF

[12] H. Sugiura, M. Gienger, H. Jannsen, and C. Goerick, "Reactive self collision avoidance with dynamic task prioritization for humanoid robots," *International Journal of Humanoid Robotics*, vol. 7, no. 01, pp. 31–54, 2010.

[13] M. Behnisch, R. Haschke, H. Ritter, and M. Gienger, "Deformable trees - exploiting local obstacle avoidance," in *Int. Conf. Humanoid Robots*, 2011, pp. 658–663.

[14] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Computer Science Dept, Iowa State University, Tech. Rep. TR*, pp. 98–11, 1998.

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

[16] E. Plaku, K. Bekris, and L. Kavraki, "Oops for motion planning: An online, open-source, programming system," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2007.

[17] S. R. Lindemann and S. M. LaValle, "Incrementally reducing dispersion by increasing voronoi bias in rrts," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004.

[18] R. Haschke, E. Weitnauer, and H. Ritter, "On-line planning of time-optimal, jerk-limited trajectories," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2008, pp. 3248–3253.

[19] T. Kröger and F. Wahl, "Online trajectory generation: basic concepts for instantaneous reactions to unforeseen events," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 94–111, 2010.

[20] M. Gienger, H. Janssen, and C. Goerick, "Task-oriented whole body motion for humanoid robots," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2005.

[21] M. Toussaint, M. Gienger, and C. Goerick, "Optimization of sequential attractor-based movement for compact movement representation," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2007.

[22] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," *Advances in neural information processing systems*, vol. 15, pp. 1523–1530, 2002.

[23] M. Behnisch, R. Haschke, and M. Gienger, "Task space motion planning using reactive control," in *IEEE-RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2010.