Planning Multi-Robot Grasping Motions

Nikolaus Vahrenkamp, Enrico Kuhn, Tamim Asfour and Rüdiger Dillmann
Institute for Anthropomatics
Karlsruhe Institute of Technology
Adenauerring 2, 76131 Karlsruhe, Germany
Email: {vahrenkamp,kuhn,asfour,dillmann}@kit.edu

***Abstract*— In this work, a motion planning approach for multi-robot grasping problems is presented. The proposed *Multi-Robot-RRT* planner controls multiple *IK-RRT* instances and evaluates grasp qualities for resulting multi-handed grasping configurations until a global solution is found. Furthermore the *Multi-Robot-RRT* concept avoids deadlock situations which may appear due to the decomposed planning approach. RRT-like search in phase space is applied to the results to generate collision-free phases for each motion that allow a simultaneous execution of planned trajectories.**

Two experiments are presented to demonstrate the usability of the proposed algorithms for planning problems covering up to 60 DoF of three humanoid robots. The planners enable humanoid robots to cooperatively grasp large objects by combining the search for solutions to the three main problems of multi-robot grasping to one integrated planning concept: building feasible bimanual grasping configurations for each robot, searching a stable multi-robot grasping setup and planning collision-free motions.

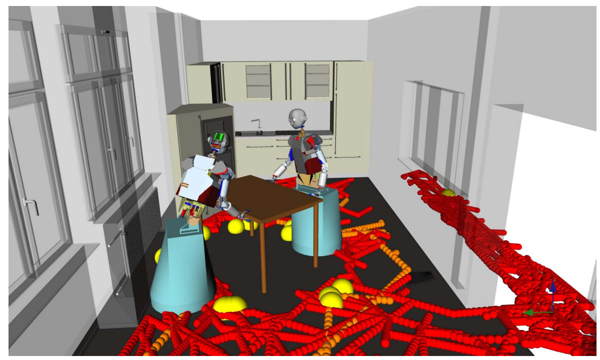

Fig. 1. Two humanoid robots are grasping a table cooperatively.

I. INTRODUCTION

Humanoid robots operating in human-centered environments should be able to plan collision-free motions for grasping and manipulation tasks. In case multiple robots operate in a shared workspace, the algorithms for motion planning have to deal with high dimensional configuration spaces (C-Space) and thus efficient planning algorithms are needed. Rapidly-exploring Random Trees (RRT) are suitable for searching high dimensional C-Spaces and for solving planning problems for a large number of degrees of freedom (DoF). Since the basic motion planning algorithms take into account a start and a goal configuration in C-Space, a solution of the inverse kinematics (IK) problem has to be pre-computed if a motion for a grasping task is searched. Since the redundancy of the robot defines a goal space which covers all configurations that solve the IK-problem, the selection of one IK solution out of the goal space limits the search for a collision-free motion unnecessarily. In case the goal is to grasp an object and multiple grasping positions are available for that object, the goal space covers the IK solutions for all grasps and again, a selection of one grasp with a specific IK-solution just selects one configuration out of the goal space. To avoid such heuristics for selecting grasps and IK-solutions, we employ the *IK-RRT* concept which samples the goal space during planning in order to generate potential target configurations. As shown in Fig. 2, the *Multi-Robot-RRT* manages n *IK-RRT* instances, one for each robot. The sub-planners serve potential grasping trajectories until a valid multi-robot solution is found that results in a stable multi-handed grasping configuration. Due to the decoupled planning approach deadlock situations have to be recognized and the planner rejects such invalid solutions. Finally a search in the n-dimensional phase space is applied in order to find a valid setup for simultaneous execution of the planned trajectories.

The problem of finding a suitable IK-solution for grasping covering the base position and redundant kinematic chains of a humanoid robot is addressed in section III. The proposed IK-solver is used in section IV for planning bimanual grasping motions. For collision-free execution of planned grasping motions a RRT-like search in phase space is applied (see sec. V). Finally, two evaluation setups are presented in section VI.

II. RELATED WORK

The search for collision-free motions can be efficiently done by sampling-based motion planning algorithms, e.g. the Rapidly-exploring Random Trees [1], [2]. In [3], [4], [5] the target for planning is defined explicitly or implicitly as a grasping target in workspace, which allows the use of a larger goal space for planning. By using heuristics or sampling techniques, potential goal configurations are generated during planning in order to cover a large set of targets. In [6] a planning approach for mobile manipulators is presented, where motions for a mobile platform together with a manipulator are planned. Trajectory planning for multiple mobile platforms is done in [7] and algorithms for planning bimanual motions are discussed in [8]. Numerous works exist dealing with optimal control and trajectory optimization

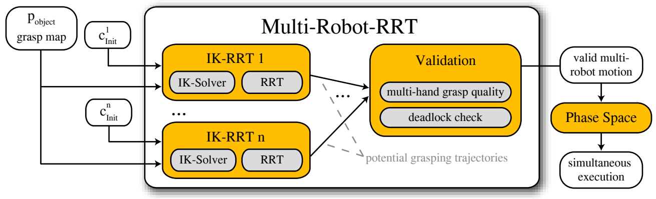

Fig. 2. Overview of the proposed planning scheme.

for grasping motions and for multi-robot setups. E.g. in [9] and [10] complex motions are generated by optimizing an initially given trajectory having the dependency of starting with a good guess. Furthermore these approaches rely on a fixed start and goal configuration which limits the feasible goal-space and may result in situations where no collision-free motions can be found.

The online RAMP-planner [11] is able to handle multiple robot motions. Here, a local sensing and re-planning concept is realized, where each robot acquires the motions of the other robots by sensing and adapts its motion according to this information. By contrast to this local approach, a centralized algorithm is presented in [12] to plan the motions of multiple robots by using time parameterizations of paths in two and three-dimensional C-Spaces. The resulting planning spaces are similar to the *phase space* concept, presented in this work, and called *unit square* and *unit cube* respectively. A decoupled method is described in [13] where tasks get prioritized and planned one after another. Simultaneous, collision-free planning using mixed integer linear programming formulations is presented in [14]. In [15] a decoupled approach for solving a a multi robot planning problem is presented, where several potential solution paths are generated for each robot. In a second step a valid scheduling that avoids inter-robot collisions is determined by applying a stop-and-go strategy. Deadlock configurations are discussed and local re-planning is proposed for handling such situations.

Reachability distributions are used in [16], [17] to quickly decide whether a workspace pose is reachable or not. These reachability distributions can be generated approximatively, e.g. by defining a discretized 6D voxel structure, or analytically as done in [18]. The problem of finding feasible robot positions for grasping is addressed in [19], where the distribution of robot-object relations is learned and uncertainties are covered. In [20] the position of the robot during execution of 3D trajectories is derived by correlating a 3D reachability map with the workspace trajectory of the end effector.

III. Whole Body IK-Solving

An often requested task in the context of manipulation planning is to find a collision-free grasping trajectory for one or two robot arms. In case the grasping configuration is known (e.g. computed by solving the IK problem) and the position of the robot is fixed, a standard BiRRT algorithm can be used in order to find a collision-free trajectory (see [1]). In case the goal is given as a 6D pose in workspace,

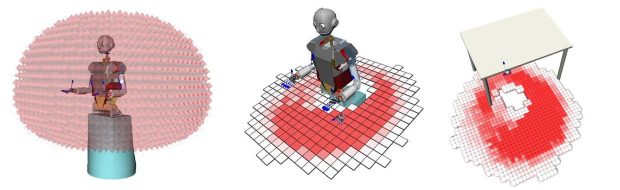

Fig. 3. (a) A projection of the discretized 6D reachability distribution for the left hand using 11 DoF of ARMAR-III. The sampled joints cover the platform rotation, three hip and seven arm joints. (b) The height and the orientation of the grasping pose define a 2D plane of the reachability distribution. The intensity of the color is proportional to the probability that a pose with the height and the orientation of the depicted grasp is reachable when using 11 joints of platform, hip and left arm. (c) The inverse reachability distribution for a grasping position.

defining the grasping pose of the end-effector (EEF), the goal space of the planning problem can be sampled during planning (see [3], [5]) in order to generate potential target configurations. As described in [5], the use of a discretized reachability distribution can speed up the randomized search for IK-solutions in the context of redundant mobile manipulation. The general approach splits the kinematic chain into two parts: the joints covered by an IK-solver (e.g. 6 or 7 DoF of an arm) and the free parameters which cover the remaining joints. By sampling the free parameters and using the reachability distribution to quickly decide if a call to the IK-solver will be successful, an efficient randomized IK-solver can be constructed. In case the IK-algorithm is used to search a feasible base position of the robot together with a valid joint configuration, we propose the use of inverted reachability distributions to efficiently sample potential robot positions related to a grasping configuration.

In [17], the reachability distribution of an arm is used to build equivalent classes, representing rotated grasping poses in the plane. These equivalent classes are used to build an inverse reachability map, which serves an approximated probability that a position of the robot is feasible for finding an IK solution to a given grasping pose. The approach we are presenting here, builds upon the idea of building a 2D reachability map that offers the possibility for finding potential robot positions for grasping. Instead of building equivalent classes for representing orientations, we propose a direct way of building the inverse reachability map from a reachability distribution that covers the orientation of the robot. This means that the orientational relations between the grasping pose and the robot's base coordinate system are already covered by the reachability distribution and thus no equivalent classes have to be built. As an example the 6D reachability for 11 joints of ARMAR-III [21] can be seen in Fig. 3(a). Here, the kinematic chain used for building the reachability distribution covers the rotation of the robot base, three hip joints and seven arm joints of the left arm.

A. Inverse Reachability Map

When considering a grasping pose $g = (t_g, o_g)$, consisting of a translational part $t_g = (g_x, g_y, g_z)$ and an orientation $o_g = (g_\alpha, g_\beta, g_\gamma)$, a 2D distribution can be constructed by fixing g_z, g_α, g_β and g_γ. This distribution, visualized in

Fig. 3(b), describes the reachability of a grasping position at height g_z and with the orientation o_g in the robot's base coordinate system. The base coordinate system could be defined in the platform for a wheeled mobile robot or in the torso when considering a legged humanoid robot. When changing the point of view from the robot's base coordinate system to a specific grasping pose, the inverse reachability map can be constructed by applying the inverse transformations to the grasping pose. Thus instead of defining the reachability for an EEF pose in the robot's base coordinate system we are now defining the distribution of the 2D base positions describing the probability that a specific EEF pose is reachable.

Fig. 3(c) shows the inverse reachability map of ARMAR-III that has been calculated for the depicted grasping pose. Due to the discretized reachability structure, the entries in the reachability map can be efficiently calculated by transforming each grid point to the (x, y)-plane in the robot's base coordinate system by applying the inverse base-to-EEF transformation for the actual grasping pose.

If a set of k grasping configurations $g_1, \dots, g_k$ is defined for an end-effector and an object, the resulting reachability map can be derived by uniting the reachability distributions.

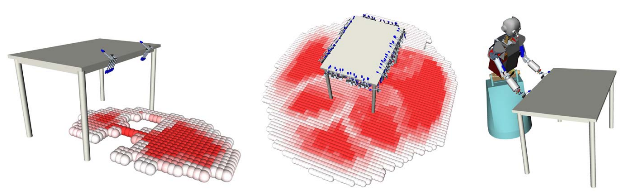

Fig. 4. (a) The inverse bimanual reachability map for two grasps. (b) The united inverse reachability map for 60 grasping positions. (c) Result of the dual arm IK-solver.

B. Inverse Reachability Map for Dual-Arm Grasps

If there are two grasp maps defined, G_l for the left and G_r for the right hand of a humanoid robot, it is possible to define dual-arm grasping combinations by testing all $|G_l| \cdot |G_r|$ possible combinations in advance. All solution to the dual-arm IK problem are then stored in a dual-arm grasp map. We propose a different way of finding dual-arm grasping combinations which can be done online. By computing the minimum of the two reachability maps for the left and the right hand, the unified bimanual reachability distribution R_{bi} gives a good hint where potential robot positions for applying dual-arm grasps are located. Here, the minimum of both reachability distributions is used, since the resulting value should represent the probability of finding a dual arm IK solution and when the search for one arm fails, the whole IK-algorithm fails. Additionally to the probability of reaching the grasping poses, a link to all reachable grasps is stored in each cell of R_{bi}, so that the set of potential reachable grasps can be retrieved quickly later on. In Fig. 4(a) and (b) R_{bi} is depicted for two and for 60 grasping positions.

$$R_{bi}(x, y) = \min(R_{left}(x, y), R_{right}(x, y)) \tag{1}$$

C. Dual-Arm IK-Solver

Solving the inverse kinematics (including the position of the robot) for a grasping task can be done in a two-step scheme, where in the first step, the position and orientation of the robot is determined and then the IK is solved to bring the hand to a target grasping pose. By contrast to such two-step approaches we propose a randomized algorithm for solving dual arm IK-problems covering the base position of the robot and a redundant kinematic chain for grasping. Since this IK-approach will be used within a sampling-based motion planning algorithm, the probabilistic approach fits well in the planning scheme.

Alg. 1 shows a general approach for solving dual-arm IK-problems. This algorithm can be used, in case multiple grasping poses (G_{left} and G_{right}) are available for applying a bimanual grasp, without the need to specify any bimanual combination in advance. The approach is used to find a solution covering the base position of the robot and additional *free parameters*, which represent all joints not belonging to the arms. It is assumed that there is an analytic IK-solver for an arm present in the system, which can be called for a specific grasping pose (see Alg. 1 line 7 and 8). The approach first queries the inverse reachability map for sampling a suitable base position for the robot and the free parameters are sampled randomly. Furthermore the call to the inverse reachability map serves a list of all grasps ($G'_{left} \subseteq G_{left}, G'_{right} \subseteq G_{right}$) that are reachable from the sampled base position (x, y) and thus only these bimanual grasping combinations have to be checked by the analytic IK-solver. If there exists a solution for at least one grasp from the left and at least one from the right set of potential grasping poses, the bimanual solution is reported. An exemplary result of the bimanual IK-solver can be seen in Fig. 4(c).

Algorithm 1: $IkDualArm(G_{left}, G_{right})$

1 **while** $(!TimeOut())$ **do**
2     $(x, y, G'_{left}, G'_{right}) \leftarrow SamplePosition(R_{bi})$;
3     $c_{free} \leftarrow SampleFreeParameters()$;
4     $c_{sampled} \leftarrow \{x, y, c_{free}\}$;
5     $Robot.SetConfig(c_{sampled})$;
6   **forall the** $(g_l \in G'_{left}$ *and* $g_r \in G'_{right})$ **do**
7         $c^{left}_{arm} \leftarrow SolveIKAnalytic(g_l)$;
8         $c^{right}_{arm} \leftarrow SolveIKAnalytic(g_r)$;
9   **if** $(c^{left}_{arm} \,\&\, c^{right}_{arm})$ **then**
10             $c_{result} \leftarrow \{c_{sampled}, c^{left}_{arm}, c^{right}_{arm}\}$;
11   **if** $(!Collision(c_{result}))$ **then**
12   **return** c_{result};
13   **end**
14   **end**
15   **end**
16 **end**
17 **return** $NULL$;

Algorithm 2: Bimanual IK-RRT(c_{start},G_{left},G_{right})

$RRT1.AddConfiguration(c_{start})$;
$RRT2.Clear()$;
while *(!TimeOut())* **do**
  if $(\#IKSolutions == 0 \mid\mid rand() < p_{IK})$ **then**
    $c_{IK} \leftarrow IkDualArm(G_{left}, G_{right})$;
    if (c_{IK}) **then**
      $RRT2.AddConfiguration(c_{IK})$;
    end
  else
    $c \leftarrow GetRandomConfiguration()$;
    if $(RRT1.Connect(c)\ \&\ RRT2.Connect.(c))$ **then**
      $Solution \leftarrow BuildSolutionPath(c)$;
      return $PrunePath(Solution)$;
    end
  end
end
return $NULL$;

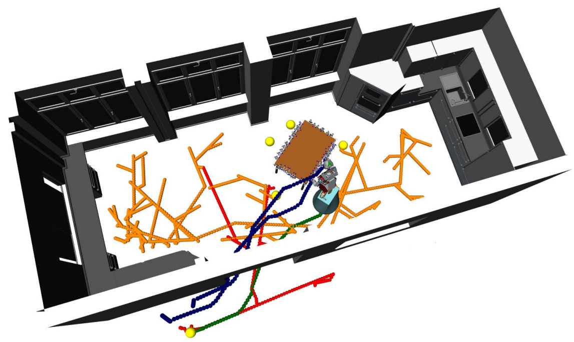

Fig. 5. A planning scene for 20 DoF of ARMAR-III covering platform, hip and both arms. The resulting trajectory is marked in blue (movements of both TCPs) and green (platform movement). The platform positions of the start and the sampled goal configurations are marked in yellow. The platform movements of the two search trees are depicted in red and orange.

IV. Planning Bimanual Grasping Motions

To plan single and dual arm grasping motions, the randomized IK-approaches described in section III-C are used to build a planning algorithm that samples potential target configurations while searching a collision-free motion. This concept of integrating the two tasks of searching IK-solutions and planning collision-free motions to one planning scheme, enlarges the goal space compared to a planner which uses one fixed IK-solution as target (e.g. RRT-Connect [1]). Similar to the *Workspace-Goal-Region planner* [3] or the *Re-Grasp RRT* [5], the proposed planners *Bimanual IK-RRT* and *Multi-Robot-RRT* take advantage of the large goal space that can be quickly sampled by the probabilistic IK-solvers.

A. Bimanual IK-RRT

The *Bimanual IK-RRT* algorithm is a bidirectional RRT-approach which is initialized with a forward search tree containing the start configuration and an empty backward search tree (see Alg. 2). During planning, the IK-solver is queried and IK-results are added to the backward tree as possible target configurations. The parameter p_{IK} adjusts how often a new IK-solution should be searched. When the RRT algorithm finds a connecting configuration between the two search trees, a solution can be generated and by going back the backward search tree, the resulting IK solution together with it's grasping position is determined and reported. Fig. 5 shows an exemplary result of the *Bimanual IK-RRT* approach.

B. Multi-Robot-RRT

To plan cooperative grasping motions for n robots, the *Multi-Robot-RRT* starts n parallel instances of the *Bimanual IK-RRT* algorithm which are configured to search solutions until a stop signal arrives. The solutions are collected and the resulting multi-robot grasping configurations are evaluated by an online grasp scoring module. Evaluating the quality of multi-handed grasping configurations is done by computing the resulting wrench space for the given setup (see e.g. [22], [23]). Two exemplary multi-robot grasping setups are depicted in Fig. 7, where the table is grasped by two robots and 20 contact points are marked by the resulting friction cones. The computed grasp quality is used to rate the grasping configuration and if a minimum score was achieved and the final robot poses do not collide, the sub-planners are stopped and the solution is reported.

C. Deadlocks

To avoid deadlock situations during execution later on, deadlocks are determined (and rejected) by the *Multi-Robot-RRT* planner (see Alg.3 line 9). In order to guarantee, that no deadlock can occur in the resulting trajectory setup, a check for the following condition is performed and in case it is violated the trajectories $(t_1, \dots, t_n)$ are rejected:

The trajectories t_i can be executed without mutual collisions, if for each robot $k \in (1, \dots, n)$ the start and goal configuration (c^k_{start} and c^k_{goal}) does not collide with any configuration on the trajectories t_j of a robot $j \neq k$. This can be summarized as follows:

$$\begin{aligned} \forall k :& \\ \forall j \neq k :& \\ \forall c_j \in t_j :& \quad !Collision(c^k_{start}, c_j) \wedge \\ & \quad !Collision(c^k_{goal}, c_j) \end{aligned} \tag{2}$$

When (2) is true, a collision-free execution of the given trajectories is definitely possible, since at least a sequently execution does not result in any conflicts. This is guaranteed because for this kind of execution only one robot k is moved at a time step and the other robots remain in their start or goal configuration, which are known to be collision-free w.r.t. t_k. This condition may be too strict in some special situations but it guarantees that the trajectories can be executed at least sequently.

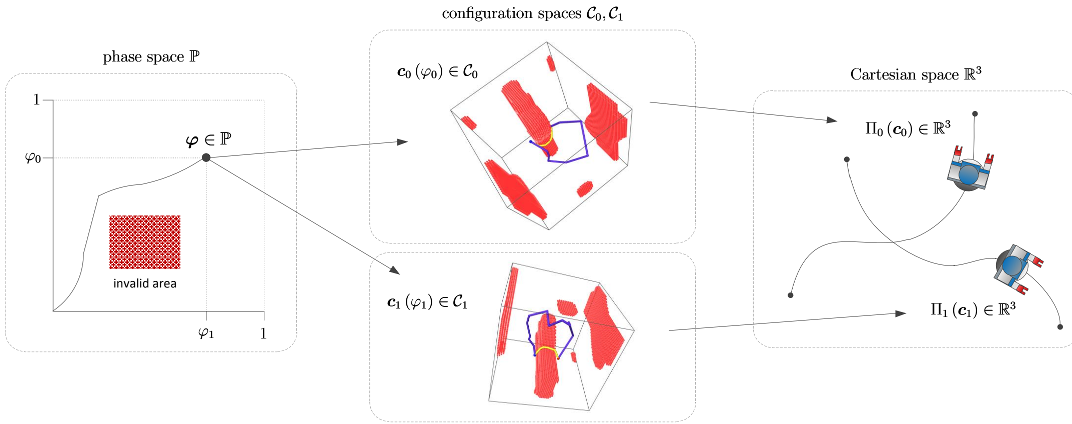

Fig. 6. Double indirection of the validation process for a phase vector $\boldsymbol{\varphi}$. First the components of the phase vector are mapped into the configuration spaces that can be mapped to Cartesian robot positions in the second place. Finally, a collision check is performed.

Algorithm 3: Multi-Robot-RRT($c_1, \ldots, c_n, G_l, G_r$)

for $l = 1$ **to** n **do**
  $RRT_l \leftarrow$
  $StartThread$(Bimanual IK-RRT(c_l, G_l, G_r));
end
while $(!TimeOut())$ **do**
  if $(NewSolution(RRT_1, \ldots, RRT_n))$ **then**
    forall the $(s_1 \in Solutions(RRT_1), \ldots,$
    $s_n \in Solutions(RRT_n))$ **do**
      if $(!Deadlock(s_1, \ldots, s_n)\,\&$
      $GraspScore(s_1, \ldots, s_n) > s_{min})$ **then**
        $StopAllPlanners()$;
        return $PrunePaths(s_1, \ldots, s_n)$;
      end
    end
  end
end
$StopAllPlanners()$;
return $NULL$;

V. Planning The Simultaneous Execution Of Multiple Trajectories

Since the planned trajectories are only collision-free in the static environment and potential movements of other robots are not considered during planning, the execution has to take care of inter-robot collisions. A simple strategy will be, to execute the trajectories sequently, which can be done in any case, as this is guaranteed by the deadlock checks of the *Multi-Robot-RRT* planner. A general approach for planning the simultaneous execution of multi-robot trajectories is given in this section.

A trajectory $\boldsymbol{t}$ is defined to be a list of m configuration vectors $\boldsymbol{c} \in \mathcal{C}$ with an additional value $\varphi \in [0,1]$ attached to each of these configurations: $(\boldsymbol{c}, \varphi) \in \boldsymbol{t}$. As φ represents the chronological order when the configurations $\boldsymbol{c}$ are reached, it is asserted that $\forall (\boldsymbol{c}_j, \varphi_j), (\boldsymbol{c}_k, \varphi_k) \in \boldsymbol{t} : j < k \leftrightarrow \varphi_j < \varphi_k$. Thus, φ is called the *phase of the trajectory* throughout this

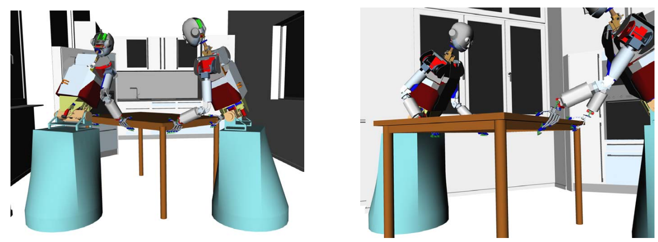

Fig. 7. The table is grasped by two robots and the resulting friction cones of 20 contact points are visualized. The online grasp quality measurement module reported a grasp quality of 0.56 for the left situation and the right configuration was evaluated with 0.79.

paper.
Observe, that $\boldsymbol{t}$ only has a finite number of configurations stored, whereas φ operates on a continuous interval. Due to this fact, it is assumed, that an interpolation function $\mathcal{I}_{\boldsymbol{t}}$ is provided, which maps from $[0,1]$ into the configuration space.

Giving n trajectories $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n$, each assigned to an exclusive robot, the task is to execute all of the movements simultaneously. Because every configuration $\boldsymbol{c}_i$ in every $\boldsymbol{t}_i$ can be addressed by a phase term φ_i, a vector $\boldsymbol{\varphi} \in \mathbb{P} := [0,1]^n$ describes a complete execution state of all robot trajectories. $\mathbb{P}$ is called the *phase space* here. Note, that these trajectories may neither have the same lengths nor the same configuration spaces. Therefore, each C-Space and trajectory is labeled with the index of the corresponding robot.

Also, $\forall \boldsymbol{t}_i$ it is assumed that $\boldsymbol{t}_i$ by itself is collision-free, i.e. $\forall \varphi \in [0,1] \;\; : \mathcal{I}_{\boldsymbol{t}_i}(\varphi) \in \mathcal{C}_{i,\text{free}}$, where $\mathcal{C}_{i,\text{free}}$ denotes the static, obstacle-free C-Space of the regarded robot i. As the robots pairwise act as obstacles to each other, it has to be asserted, that configurations $\boldsymbol{c}_i, \boldsymbol{c}_j$, which are elements of $\mathcal{C}_{i,\text{free}}$ and $\mathcal{C}_{j,\text{free}}$ but result in a collision of the robots i and j at a specific phase $\boldsymbol{\varphi}$, are not applied to the associated robots at the same time. The following predicate *valid* models this issue:

$$\text{valid}(\boldsymbol{\varphi}) = \text{true} \Leftrightarrow \varnothing = \bigcap_{i \in \{1,\ldots,n\}} \left(\Pi_i \circ \mathcal{I}_{\boldsymbol{t}_i}\right)(\varphi_i)$$

where Π_i is a map from the configuration space of robot i into the workspace covering all Cartesian points occupied by this robot. A visualization of that scheme is depicted in Fig. 6. To simultaneously execute the movements, the goal is to find a path $\Phi = \left(\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_p\right) \in \mathbb{P}^p$ that fulfills the following requirements:

$$\begin{aligned} \forall \boldsymbol{\varphi} \in \Phi : \mathrm{valid}\left(\boldsymbol{\varphi}\right) \\ \boldsymbol{\varphi}_1 = (0, \ldots, 0)^T \\ \boldsymbol{\varphi}_p = (1, \ldots, 1)^T \end{aligned}$$

Furthermore, a segment between two successive phases has to be valid. Similar to sampling-based motion planning approaches, where it is assumed that a path segment is collision-free, when a densely sampled sequence of intermediate configurations is collision-free, a sampling of intermediate phases can be applied for validating path segments in phase space.

A. RRT-based phase space sampling

Describing this issue in such a manner, enables to directly translate the approach into a RRT-solvable problem. In order to use the RRT-CONNECT planner as described in [1] for searching a valid path Φ in phase space, we present an adaption of the EXTEND method for building the tree. In Alg. 4 the modified version is shown, where $\mathcal{R}$ denotes an object of the search tree in phase space, which supports common methods like adding a vertex (i.e., a phase vector) and the corresponding edge to the nearest node of the RRT. Two parameters are used in the given algorithm: $\varepsilon_{\text{sampl}}$ describes the sampling step size for validating the path segment and $\varepsilon_{\text{extend}}$ defines the EXTEND step size for adding new nodes to the RRT.

Algorithm 4: Extend($\mathcal{R} \in \mathrm{PhaseSpaceTree}$, $\boldsymbol{\varphi} \in \mathbb{P}$)

$\boldsymbol{\varphi}_{\text{near}} \leftarrow \mathrm{NearestNeighbor}\left(\mathcal{R}, \boldsymbol{\varphi}\right)$;
$\Delta\boldsymbol{\varphi}_{\text{sampl}} \leftarrow \varepsilon_{\text{sampl}} \cdot \frac{\boldsymbol{\varphi}-\boldsymbol{\varphi}_{\text{near}}}{\|\boldsymbol{\varphi}-\boldsymbol{\varphi}_{\text{near}}\|}$;
$\boldsymbol{\varphi}_{\text{new}} \leftarrow \boldsymbol{\varphi}_{\text{near}}$;
for $l = \varepsilon_{sampl}$ **to** $\min\left\{\varepsilon_{extend}, \|\boldsymbol{\varphi} - \boldsymbol{\varphi}_{near}\|\right\}$ **step** ε_{sampl} **do**
  $\boldsymbol{\varphi}_{\text{new}} \leftarrow \boldsymbol{\varphi}_{\text{new}} + \Delta\boldsymbol{\varphi}_{\text{sampl}}$;
  if $\neg\,\mathrm{valid}\left(\boldsymbol{\varphi}_{new}\right)$ **then**
    return *Trapped*;
  end
end
$\mathcal{R}$.AddVertex($\boldsymbol{\varphi}_{\text{new}}$);
$\mathcal{R}$.AddEdge($\boldsymbol{\varphi}_{\text{near}}$,$\boldsymbol{\varphi}_{\text{new}}$);
if $\boldsymbol{\varphi}_{new} = \boldsymbol{\varphi}$ **then**
  return *Reached*;
else
  return *Advanced*;
end

The main observation is, that by sampling the phase space two indirections arise in order to check the validity of a phase. Every feasibility check performs two mappings, first from the phase space into the configuration spaces which in turn are consecutively mapped to spatial point sets. If any of these point sets intersect, the phase is invalid and therefore rejected by the validation predicate.

The loop in Alg. 4 ensures a dense sampling to avoid collisions between two adjacent phases of the RRT. However, this is only guaranteed when $\varepsilon_{\text{sampl}}$ is chosen appropriately. One estimation can be made by assuming that a phase difference of length $\varepsilon_{\text{sampl}}$ does not cause a workspace displacement greater than the smallest spatial latitude $\mathrm{d}_{\min}$ of all obstacles (including robot parts) on any robot:

$$\begin{aligned} &\forall \boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2 \in \mathbb{P}\ \forall \boldsymbol{t}_i : \left(\|\boldsymbol{\varphi}_2 - \boldsymbol{\varphi}_1\| \leq \varepsilon_{\text{sampl}}\right) \\ &\rightarrow \left(\mathrm{d_{WS}}\left(\mathcal{I}_{\boldsymbol{t}_i}\left((\boldsymbol{\varphi}_2)_i\right) - \mathcal{I}_{\boldsymbol{t}_i}\left((\boldsymbol{\varphi}_1)_i\right)\right) \leq \mathrm{d}_{\min}\right), \end{aligned}$$

where $\mathrm{d_{WS}}$ denotes the worst case workspace displacement a robot can experience with respect to a specific configuration change, see [24].

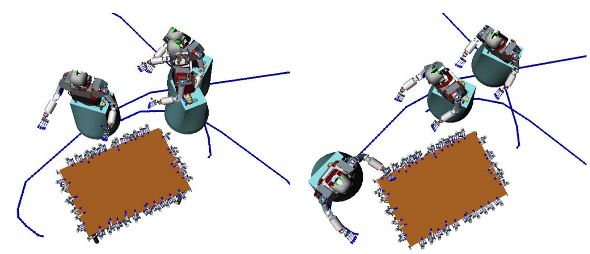

Fig. 8. Snapshots of the robots processing their trajectories. While in the left figure two of the robots collide, the situation was resolved due to the phase space planning in the right figure.

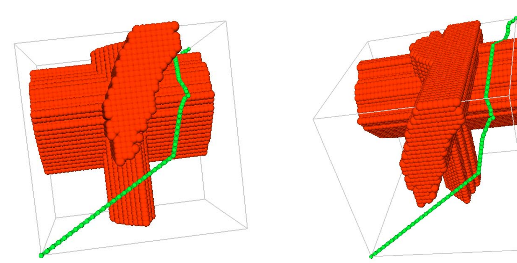

Fig. 9. Phase space visualization of the previous example. The invalid area is marked in red, the solution path is depicted in green.

Fig. 8 shows a snapshot of the simultaneous execution of planned trajectories for three robots. In the left image, the motions are executed with constant speed, resulting in collisions, whereas the right image shows how the phase space planning approach allows a collision-free execution of the same trajectories. The corresponding phase is visualized in Fig. 9. Here, the three-dimensional phase space was sampled and all $\boldsymbol{\varphi} \in \mathbb{P}$ resulting in a collision in workspace are marked in red. The path Φ, which was found by the RRT-based search in phase space, is shown in green.

VI. EVALUATION

The proposed algorithms are evaluated with two scenarios where collision-free grasping motions are searched for multiple robots. The evaluations, described in the next two sections, have been carried out on a quad-core 2.4 GHz system. The implementation was realized using the Simulation and Motion Planning Toolbox SIMOX [25]. In all tests, the probability of finding a new goal configuration p_{IK} was set to 0.03, i.e. in 3 of 100 planning cycles a new IK solution is searched.

A. *Two Robots: 40 DoF*

In this setup, two robots are located outside the kitchen and the goal is to grasp the table which is placed inside the room. The *Multi-Robot-RRT* planner is initialized with the two start configurations of the robots, the Cartesian coordinates of the table and 60 potential grasping positions, defined relatively to the target object (see Fig. 10). In Table I the average results of 30 test runs are shown. The planning time was measured with 9.5 seconds and on average 2.8 four-handed grasping combinations were checked until a collision-free and stable grasping configuration for both robots is found. The number of computed grasping trajectories is 7.8, which means that each sub-planner computes 3.9 solution trajectories on average. Each sub-planner computes IK solutions during planning, and since not all IK solutions result in a solution trajectory, the number of computed IK solutions is larger (16.8). The planning time for finding a valid solution in phase space for collision-free trajectory execution was measured with 1.0 seconds on average.

TABLE I

THE AVERAGE PLANNING RESULTS FOR THE KITCHEN SCENE WITH TWO ROBOTS (40 DOF)

Plan. Time	# IK Results	# Sol. Traj.	# Col. Checks	# Four-Handed Grasp. Conf.	Phase Space Plan. Time
9.5s	16.8	7.8	7028	2.8	1.0s

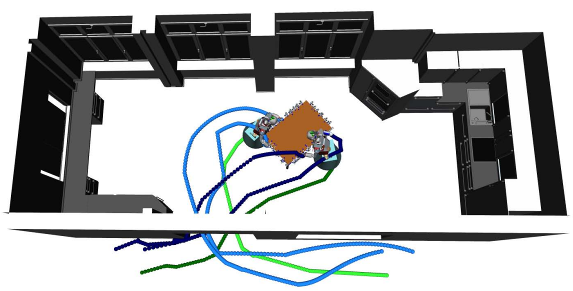

Fig. 10. The resulting trajectories for 40 DoF of two ARMAR-III robots. The Cartesian movements of platform and both hands are visualized in green and blue. The phase space based trajectory execution planning guarantees a valid execution without collisions between the two robots.

B. *Three Robots: 60 DoF*

In this test case, the *Multi-Robot-RRT* is used to generate grasping trajectories for three humanoid robots in an artificial scene (see Fig. 11). The C-Space for each robot covers 20 DoF and thus the combined C-Space is 60 dimensional. For each test run, a planning setup is generated by randomly placing 30 boxes in the scene without any initial collisions. Due to the random positioning of the boxes, a situation may occur where no trajectory can be found to a stable grasping configuration. Such a situation could not be observed directly, but when a planning run takes more than 5 minutes, it is stopped and a failure is reported (during the 30 test runs, this situation was observed once).

The evaluation is performed on a multi-core system, allowing the *Multi-Robot-RRT* planner to take full advantage of the parallelized concept, since the four cores host three *Bimanual IK-RRT* planners and one thread for collecting the results and computing the grasp qualities as described in section IV-B. The results of Table II emphasize the good performance of the parallelized planning approach. The high number of checked grasping configurations is caused by the grasp checking module, which rejects most grasping configurations due to mutual collisions of the robot's grasping positions or because the deadlock check reports a trajectory setup which cannot be executed simultaneously. Searching a solution in the three dimensional phase space was successful for all trajectories and the average planning time was measured with 7.0 seconds.

TABLE II

THE AVERAGE PLANNING RESULTS FOR THE RANDOM BOX SCENE WITH THREE ROBOTS (60 DOF)

Plan. Time	# IK Results	# Sol. Traj.	# Col. Checks	# Six-Handed Grasp. Conf.	Phase Space Plan. Time
8.9s	20.2	13.0	19434	37.1	7.0s

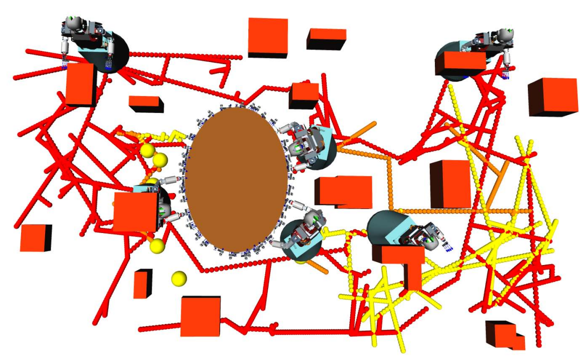

Fig. 11. An exemplary 60 DoF planning scene with three humanoid robots. The start and goal configurations of the robots are shown together with the workspace representation of the search trees (platform movements).

VII. CONCLUSIONS

In this work an approach for efficient planning of multi-robot grasping trajectories was presented. The *Bimanual IK-RRT* approach combines the search for valid IK-solutions with RRT-based planning of collision-free motions and thus a large goal space can be used.

The proposed bimanual IK-solvers utilize an inverted reachability distribution for quick sampling of suitable robot base positions for grasping. With this approach an efficient sampling of bimanual IK solutions for 20 DoF of a humanoid robot can be realized as showed by the experiments.

By parallelizing the execution of multiple *Bimanual IK-RRT* planners, the *Multi-Robot-RRT* approach realizes the simultaneous search for collision-free grasping trajectories for multiple robots. The online grasp quality measurement is used to find a stable multi-robot grasping configuration together with its solution trajectories. The *Multi-Robot-RRT* planner guarantees, that the execution of the resulting motion can be done at least sequently due to the included deadlock check. In order to find better execution setups avoiding inter-robot collisions, a RRT-based search in phase space is applied to find non-colliding phases allowing a simultaneous execution of planned motions. The experiments showed that, due to the parallelized approach, the performance is suitable for real-world systems and solutions in high dimensional configuration spaces can be computed efficiently.

Since the proposed approach relies on a perfect world-knowledge, future work will address how inaccurate world modeling or missing information can be handled and how efficient re-planning can be done for changing environments. Furthermore we plan to investigate how dynamics can be considered by the phase space concept.

VIII. ACKNOWLEDGMENTS

The work described in this paper was partially conducted within the German Humanoid Research project SFB588 funded by the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft) and the EU Cognitive Systems project GRASP (IST-FP7-IP-215821) funded by the European Commission.

REFERENCES

[1] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *IEEE Int'l Conf. on Robotics and Automation (ICRA'2000), San Francisco, CA*, 2000, pp. 995–1001.

[2] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning." *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.

[3] D. Berenson, S. Srinivasa, D. Ferguson, A. Collet, and J. Kuffner, "Manipulation planning with workspace goal regions," in *IEEE Int'l Conf. on Robotics and Automation (ICRA'2009),Kobe,Japan*, 2009.

[4] M. Gienger, M. Toussaint, and C. Goerick, "Task maps in humanoid robot manipulation," in *IEEE International Conference on Intelligent Robot and Systems (IROS 2008)*. IEEE, 2008.

[5] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, "Humanoid motion planning for dual-arm manipulation and re-grasping tasks," in *Intelligent Robots and Systems, IROS*, October 2009.

[6] O. Brock and O. Khatib, "Real-time replanning in high-dimensional configuration spaces using sets of homotopic paths," in *In IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 550–555.

[7] Y. Yamamoto and S. Fukuda, "Trajectory planning of multiple mobile manipulators with collision avoidance capability," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 4, 2002, pp. 3565 – 3570 vol.4.

[8] Y. Fei, D. Fuqiang, and Z. Xifang, "Collision-free motion planning of dual-arm reconfigurable robots," *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 4, pp. 351 – 357, 2004.

[9] M. Toussaint, "Robot trajectory optimization using approximate inference," in *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*. New York, NY, USA: ACM, 2009, pp. 1049–1056.

[10] N. Ratliff, M. Zucker, J. A. D. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2009.

[11] J. Vannoy and J. Xiao, "Real-time adaptive motion planning (ramp) of mobile manipulators in dynamic environments with unforeseen changes," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1199 –1212, oct. 2008.

[12] B. Aronov, M. de Berg, A. F. van der Stappen, P. Švestka, and J. Vleugels, "Motion planning for multiple robots," *Discrete and Computational Geometry*, vol. 22, no. 4, pp. 505–525, Dezember 1999.

[13] M. Erdmann and T. Lozano-Pérez, "On multiple moving objects," *Algorithmica*, vol. 2, no. 1-4, pp. 477–521, November 1987.

[14] J. Peng and S. Akella, "Coordinating multiple robots with kinodynamic constraints along specified paths," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 295–310, 2005.

[15] C. Ferrari, E. Pagello, M. Voltolina, J. Ota, and T. Arai, "Varying paths and motion profiles in multiple robot motion planning," jul. 1997, pp. 186 –193.

[16] L. Guilamo, J. Kuffner, K. Nishiwaki, and S. Kagami, "Efficient prioritized inverse kinematic solutions for redundant manipulators," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, Aug. 2005, pp. 3921–3926.

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

[18] D. Kee and W. Karwowski, "Analytically derived three-dimensional reach volumes based on multijoint movements," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 44, pp. 530–544(15), 2002.

[19] F. Stulp, A. Fedrizzi, and M. Beetz, "Action-related place-based mobile manipulation," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 10-15 2009, pp. 3115 –3120.

[20] F. Zacharias, W. Sepp, C. Borst, and G. Hirzinger, "Using a model of the reachable workspace to position mobile manipulators for 3-d trajectories," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, 7-10 2009, pp. 55 –61.

[21] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "Armar-III: An integrated humanoid platform for sensory-motor control." in *IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006)*, December 2006, pp. 169–175.

[22] N. Pollard, "Parallel methods for synthesizing whole-hand grasps from generalized prototypes," *Technical Report AI-TR 1464, MIT, Artificial Intelligence Laboratory*, 1994.

[23] A. T. Miller, "Graspit!: a versatile simulator for robotic grasping." Ph.D. dissertation, Department of Computer Science, Columbia University, 2001.

[24] N. Vahrenkamp, C. Scheurer, T. Asfour, R. Dillmann, and J. Kuffner, "Adaptive motion planning for humanoid robots," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008.

[25] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Simox: A Simulation and Motion Planning Toolbox for C++," Karlsruhe Institute of Technology (KIT), Tech. Rep., 2010.