Optimization of fluent approach and grasp motions

Michael Gienger, Marc Toussaint, Nikolay Jetchev, Achim Bendig and Christian Goerick

***Abstract*— Generating a fluent motion of approaching, grasping and lifting an object comprises a number of problems which are typically tackled separately. Some existing research specializes on the optimization of the final grasp posture based on force closure criteria neglecting the motion necessary to approach this grasp. Other research specializes on motion optimization including collision avoidance criteria, but typically not considering the subsequent grasp as part of the optimization problem. In this paper we aim to combine existing techniques for grasp optimization, trajectory optimization, and attractor-based movement representation, into a comprehensive framework that allows us to efficiently compute a fluent approach and grasping motion. The feasibility of the proposed approach is shown in simulation studies and experiments with a humanoid robot.**

I. Introduction

In robotics, grasping objects in a daily environment poses a lot of challenging questions. Many different criteria need to be fulfilled to successfully reach for and grasp an object. First, the initial robot posture may be far from a feasible grasp posture. Hence we need to plan a collision-free reaching motion towards a feasible grasp. A classical solution to this is a motion planning technique towards a predefined goal posture. Second, objects typically can be grasped at many different locations and with different grasp strategies. The best grasp choice clearly depends on the initial posture of the robot and the obstacles in the environment. Hence, predefining a grasp position and using a classical motion planner is a suboptimal solution. In essence, we are faced with the *coupled problem of grasp choice and reaching motion planning*. In this paper we solve this problem by proposing an object representation in terms of an object-specific task map which can be learnt from data and, during movement generation, efficiently coupled into a movement optimization process.

Most literature on grasp optimization focuses on the grasp itself, isolated from the reaching movement. For instance, [1] review the various literature on defining grasp quality measures, [2] learn which grasp positions are feasible for various objects, [3] efficiently compute good grasps depending on how the objects shall be manipulated, and [4] simplify the grasp computation based on abstracting objects into shape primitives. The coupling to the problem of reaching motion optimization is rarely addressed. In [5], reaching and grasping is realized by reactive control primitives. A recent approach [6] makes a step towards solving the coupled problem by including an "environment clearance score" in the grasp evaluation measure. In that way, grasps are preferred which are not prohibited by immediate obstacles directly opposing the grasp. Still, the full reaching motion is neglected in the grasp evaluation. A sampling-based movement planning approach based on an attractor description has been proposed in [7]. The authors focus on planning the reaching movement, while the grasp itself is neglected.

Our approach is based on directly coupling the grasp choice problem into the motion optimization procedure, and thereby solving both coupled problems in an integrated framework. The model-based motion optimization is based on recent work which optimizes trajectories based on a sequence of task space attractors [8]. A more probabilistic view on this is given in [9].

If we could predefine a desired pregrasp posture as the goal of the reaching movement, this optimization technique could readily compute optimal reaching motions. However, since there is many different choices of pregrasps we need to adapt the method.

A key pre-requisite for this is to learn object-specific *task maps* which represent the set of feasible grasps for an object. This is similar to the approach in [10] where a robot-centered representation of the feasible workspace is learnt, and to the object-specific "valid grasp sets" in [6]. Once such a map is learnt, it defines a *goal set* in a certain task space of the reaching motion rather than an explicit goal position. Hence, we generalize the motion optimization method to cope with general goal sets in arbitrary task spaces.

The previous approaches to learning maps of the workspace or of feasible grasps are based on exhaustive sampling over a grid. We build on previous work [11] and propose to use a more efficient technique to learn task maps. It assumes a rough shape estimate of the object, which for instance can be acquired by computer vision.

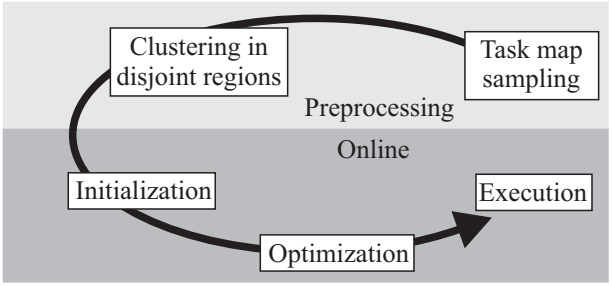


Figure 1:
Functional elements of the presented movement generation scheme

The paper is organized according to Figure 1, which shows

Michael Gienger, Achim Bendig and Christian Goerick are with the Honda Research Institute Europe, Carl-Legien-Strasse 30, 63073 Offenbach/Main, Germany. Marc Toussaint and Nikolay Jetchev are with the Machine Learning group at Technical University Berlin, Franklinstr. 28/29, 10587 Berlin, Germany.

the functional elements of our approach. Sampling and clustering task maps can be performed offline in simulations or experiments. The underlying methods will be explained in Section II. Section III revisits briefly the motion generation and optimization system, which is based on ASIMO's whole body motion control framework [12]. We will then introduce mechanisms to incorporate acquired task maps into the motion generation scheme. Section IV presents simulation studies and experimental results in a reach-grasp scenario. We find that the resulting choice of grasp nicely depends on the efficiency of the reaching motion towards the grasp from the initial robot posture. To summarize, the key novelties of our approach are:

- We propose an efficient learning scheme of task maps to rapidly explore the set of feasible grasps for an object.
- We learn object-specific task maps as a representation of a goal manifold in a task space, and cluster them into qualitatively different solution spaces.
- We solve the coupled problem of grasp choice and reaching motion optimization. This leads to a preference of grasps which are easy to reach (reachable with little cost) and thereby a disambiguation of grasp choice for manifolds of feasible grasps.
- We apply the proposed methods in reach and grasp experiments with a humanoid robot.

II. Task maps

With the term *task map*, we refer to a map that comprises a set of sampled control parameters, each associated with a scalar quality measure. In previous work [11], we proposed for instance to represent a set of hand-object pregrasp poses with respect to a failure/success criterion. These maps generally replace the concept of one explicit reaching target by the concept of a whole manifold of feasible targets in the task space. This relaxes the constraints imposed on the subsequent movement optimization process, which is particularly beneficial to improve other criteria governing the movement. If the chosen quality measure can be determined with the robot's sensors, it is further possible to build up or refine task maps in real experiments.

In the following, we will focus on simple "power grasps". However, the approach is not limited to a certain grasp. It is possible to represent different grasp types (e. g. precision grasps etc.) in several task maps. The concept even holds for bi-manual grasp strategies.

A. Task map acquisition with force closure quality measure

Learning a task map requires to explore many different grasps on a specific object. The first question is how different grasp trials are sampled. The second how each trial is evaluated. Previous approaches consider an exhaustive sampling over a grid [6]. If the grasp space is high dimensional (e.g., 6-dimensional if grasps are parameterized by the preshape position and orientation) a grid-based sampling approach is rather inefficient. In [11] it was proposed to use Rapidly exploring Random Trees for the sampling. While this technique is very fast, it is hard to generate a very dense set of samples. Further, when the set of feasible grasps separates into disjoint clusters, RRTs typically only explore one of the clusters.

In this paper we combine some of the heuristics proposed in [6] and [13] in a sampling based approach. We assume that the robot can (visually) acquire a rough estimate of the object volume. Using the approximate shape information we can sample a random point from the volume and compute a pregrasp posture. For this, we initialize the hand inside the object volume. The hand is then retracted so as to get in palm contact with the object. Subsequently the finger joints are closed until they contact the objects surface. For this grasp, we collect the following data: (1) the hand position and orientation in coordinates relative to the object frame – this 6D point $g \in \mathbb{R}^6$ will become the control parameters in the task space. (2) The contact points, normals and penetration between the finger segments and the object – this information is used to compute a quality measure based on force closure, which is a well-known measure determining the ability of a grasp to resist external forces [14].

While this learning phase can be executed on a real robot, we use realistic simulations to speed up the process. The force closure is computed from simulated tactile sensors positions and normals, excluding those that have zero pressure. In that way we collect a data set consisting of control parameters in $\mathbb{R}^6$ and the corresponding force closure scalars.

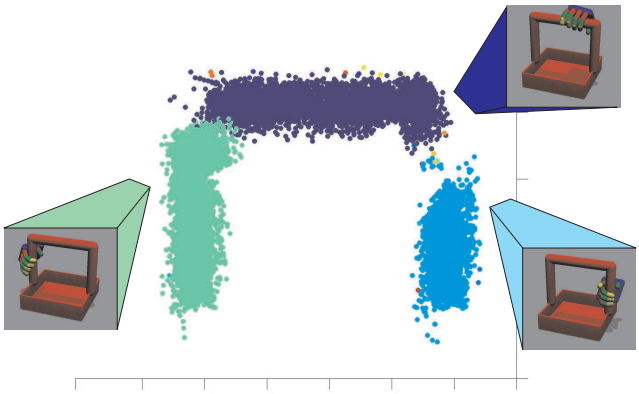


Figure 2:
Disjoint clusters in sampled task map. Solutions with the thumb and palm pointing upwards have been left out.

B. Clustering into contiguous regions

For many realistic objects, the task map will consist of disjoint clusters which form qualitatively different solutions. Consider for instance a simple basket with handle that can be grasped on the top bar or on each of the side bars. The orientations and positions of feasible grasps change discontinuously from bar to bar, such forming a set of clusters. We employ an Euclidean distance based hierarchical clustering approach to extract a set of qualitatively different solutions. The chosen

algorithm does not make an a priori assumption on the number of clusters.

Figure 2 displays the extracted clusters for the given example. Clusters are formed by grasps applied to the left, right and top handle of the basket. In the figure, only the position elements of the 6-dimensional task map parameters are visualized. We only consider clusters of samples that have a significant size, such eliminating outliers that comprise only a few samples.

III. Online movement generation

In this section we refer to the lower part of Figure 1. Other than the processing steps in Section II, the online movement generation has to be computationally efficient in order to find smooth and collision-free movements within interaction time.

The underlying robot control model is described in the form of a tree structure. The individual links are connected by degrees of freedom (joints) or fixed transformations. The tree may also comprise objects from the environment, so that task descriptors can also account for robot-object relations. We define a task as the relative movement of two tree nodes[1] and such can compute the task velocity as $\dot{x}_{task} = \dot{x}_{ef} - \dot{x}_{base}$. Indices *ef* and *base* denote the effector body and its reference, respectively.

A task can be described in different ways, for instance as linear position, inclination, spherical and Euler angles, etc. A task element may comprise just individual parts of such a description, such as the vertical element of a 3-D position. Based on this, we derive an augmented Jacobian holding all controlled task elements (see also [12], [15]). The underlying whole body motion control is based on the scheme by Liegeois [16], the equations are given in Table I, eq. (A.2). Redundancies are resolved by mapping the gradient of an optimization criterion (joint limit avoidance, etc.) into the null space of the motion.

The trajectories are generated using a dynamical systems approach. We apply the simple attractor system in eq. (A.4) (see [12] for details) to the elements of the augmented task vector. The same attractor dynamics are applied to other controllers that are not related to the inverse kinematics, such as "closing the fingers to a power grasp", etc.

A. Optimization-based motion generation

To generate a joint-limit and collision free reaching motion, we apply the attractor-based optimization scheme presented in [8]. It incorporates the employed controller (eq. (A.2)) into the optimization process. The key idea is to optimize a scalar cost function by finding an optimal sequence of task space attractor vectors which determine the robots motion.

We consider an integral cost function over the movement in the general form of eq. (A.1). Here, $q \in \mathbb{R}^n$ is the joint state vector of the n-DoF robot, $x \in \mathbb{R}^d$ is the task state vector in the augmented task space, $x^*_{1:K} \in \mathbb{R}^d$ is a series of attractor points, and $r \in \mathbb{R}^d$ an additional smoothing variable (see [8] for details). The function g subsumes cost criteria in the q-space which depend on single time steps. It is suited to account for costs that depend on the posture of the robot. We formulate criteria to account for the distance of the final end-effector state to a target, collisions and proximities between collidable objects throughout the trajectory, and joint limit proximities. The function h subsumes costs for transitions in q-space and depends on the current and the previous time steps. It is suited to formulate criteria like the global length of the trajectory in q-space and for the end effector velocity at the end of the trajectory.

The movement generation process can be summarized by equations (A.2) - (A.5). Since the dependencies between attractor points and the task space trajectories are determined by the attractor dynamics (Eqs. (A.13)-(A.17)) and the dependencies between task and joint space is determined by the whole body motion control, we can derive analytical gradients to relate the attractor point location to the chosen cost function:

$$\frac{dC}{dx^*} = \sum_{\text{children } y_i \text{ of } x^*} \frac{\partial y_i}{\partial x^*} \frac{dC}{dy_i} \,. \tag{1}$$

The partial derivatives are given in part d) of Table I. The gradient computation is carried out in a forward and a backward propagation step, for details see [8]. We use an efficient gradient-based optimization method that provides feasible solutions within the range of 0.5 to 2 seconds – which is below the critical "patience" threshold for the interaction with humans.

B. End-state comfort initialization

To find an appropriate initialization for the optimization problem, the target posture at the end of the movement is computed according to each cluster center, respectively. This requires to solve the inverse kinematics for the respective task vector, which can be done efficiently with regression techniques.

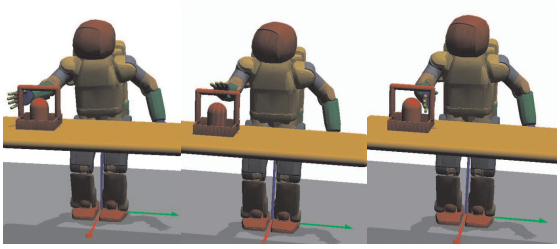
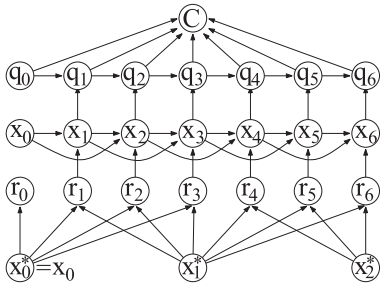


Figure 3: Goal postures according to task map clusters

The different target postures are then compared by an *end-state comfort* value, motivated from psychological studies. It is based on the joint limit cost (squared sum of the deviations of joint angles from their zero position). The best solution determines the target task vector to initialize the optimization problem. The attractor vector sequence is linearly interpolated between initial posture and the chosen goal task vector.

[1]There are other special cases for tasks, for instance the overall linear and angular momentum, etc.



a) cost function:

$$C = \sum_{t=0}^{T} g(q_t) + \sum_{t=0}^{T-1} h(q_t, q_{t+1}) \ , \tag{A.1}$$

b) movement generation:

$$q_{t+1} = q_t + J_t^{\#}(x_{t+1} - \phi(q_t)) - \alpha \ (I - J_t^{\#} J_t) \ W^{-1} \ (\partial_q H_t)^T \tag{A.2}$$

$$x_{t+1} = x_t + \pi(x_t, x_{t-1}, r_{t+1}) \tag{A.3}$$

$$\pi(x_t, x_{t-1}, r_{t+1}) = a(r_{t+1} - x_t) + b(x_t - x_{t-1}) \tag{A.4}$$

$$r_t = (1-\tau)x_k^* + \tau x_{k+1}^* \ , \quad k = \lfloor tK/T \rfloor \ , \quad \tau = \frac{t - kT/K}{T/K} \tag{A.5}$$

c) chain rules following (1):

$$\frac{dC}{dq_t} = \frac{\partial C}{\partial q_t} + \frac{\partial q_{t+1}}{\partial q_t} \ \frac{dC}{dq_{t+1}} \tag{A.6}$$

$$\frac{dC}{dx_t} = \frac{\partial q_t}{\partial x_t} \ \frac{\partial C}{\partial q_t} + \frac{\partial x_{t+1}}{\partial x_t} \ \frac{dC}{dx_{t+1}} + \frac{\partial x_{t+2}}{\partial x_t} \ \frac{dC}{dx_{t+2}} \tag{A.7}$$

$$\frac{dC}{dr_t} = \frac{\partial x_t}{\partial r_t} \ \frac{dC}{dx_t} \tag{A.8}$$

$$\frac{dC}{dx_l^*} = \sum_t \frac{\partial r_t}{\partial x_l^*} \ \frac{dC}{dr_t} \tag{A.9}$$

d) partial derivatives:

$$\frac{\partial C}{\partial q_t} = g'(q_t) + h'^1(q_t, q_{t+1}) + h'^2(q_{t-1}, q_t) \tag{A.10}$$

$$\frac{\partial q_{t+1}}{\partial q_t} = I - J_t^{\#} J_t + (\partial_q J_t^{\#})(x_{t+1} - \phi(q_t)) - \alpha \ (I - J_t^{\#} J_t) \ W^{-1} \ (\partial_q^2 H_t)^T + \alpha \ \partial_q (J_t^{\#} J_t) \ W^{-1} \ (\partial_q H_t)^T \tag{A.11}$$

$$\frac{\partial q_t}{\partial x_t} = J_{t-1}^{\#} \tag{A.12}$$

$$\frac{\partial x_{t+1}}{\partial x_t} = 1 + \pi'^1(x_t, x_{t-1}, r_{t+1}) \tag{A.13}$$

$$\frac{\partial x_{t+2}}{\partial x_t} = \pi'^2(x_{t+1}, x_t, r_{t+2}) \tag{A.14}$$

$$\pi'^1(x_t, x_{t-1}, r_{t+1}) = -a + b \ , \quad \pi'^2(x_t, x_{t-1}, r_{t+1}) = -b \tag{A.15}$$

$$\frac{\partial x_t}{\partial r_t} = \pi'^3(x_{t-1}, x_{t-2}, r_t) \tag{A.16}$$

$$\frac{\partial r_t}{\partial x_l^*} = (1-\tau)\delta_{l=k} + \tau \delta_{l=k+1} \ , \quad \tau \text{ and } k \text{ depend on } t \text{ as above} \tag{A.17}$$

TABLE I

FUNCTIONAL NETWORK OF THE CONTROL ARCHITECTURE.

It should be noted that this procedure ignores the course of the overall trajectory, but rather considers a "snapshot" of the final pose. It is therefore not guaranteed that the chosen initialization will lead to the global optimum. However, it considers the relative pose between robot and object. Further, it yields an estimate of the approximate target pose, and whether the target can be reached or not. Figure 3 illustrates the initializations for a given location of the basket on the table.

C. Integration into optimization procedure

In the following, we extend the optimization scheme to incorporate the learnt task maps seamlessly into the optimization process. The key idea is to formulate criteria to account for both the proximity to the nearest solution in the task map manifold and the quality of the resulting grasp.

We therefore formulate two optimization criteria: The *proximity criterion* enforces a smooth and collision free movement into the manifold of feasible grasps represented in the map. It so to say "pulls" the final grasp towards the manifold of valid preshape postures in the course of the optimization. The *quality criterion* evaluates the quality of the final grasp. It is based on the force-closure quality measure for each task map sample and guides the movement towards a preshape posture that leads to a high-quality grasp.

1) Proximity criterion: The proximity will replace the distance of the target end-effector state and contribute to the cost function g in eq. (A.1) during motion optimization. Given the final task state at the last time step T (e.g., the hand position and orientation relative to the object), we compute the nearest element x_{map} in the task map. Now we define a cost

$$g_p = (x_t^{\text{rel}} - x_{\text{map}})^T W_{\text{map}} (x_t^{\text{rel}} - x_{\text{map}}) \tag{18}$$

The metric W_{map} accounts for the difference in the linear and angular units. The nearest neighbor in the task map to the hand is computed with the approximate nearest neighbor algorithm described in [17]. For this, the hand position and orientation x_t^{rel} is represented in the reference frame of the object. The gradient is

$$\frac{\partial g_p}{\partial x_t^{\text{rel}}} = 2(x_t^{\text{rel}} - x_{\text{map}})^T W_{\text{map}} \tag{19}$$

2) Quality criterion: To account for the quality of the grasp that has been determined with the force closure measure, each sample of the task map is interpreted as a Gaussian $f_i = |2\pi\Sigma_i^{-1}|^{-\frac{1}{2}} \exp(-\frac{1}{2} d_{\Sigma,i})$ with a mean vector μ and some small neighborhood $d_{\Sigma,i} = (x_i - \mu)^T \Sigma_i (x_i - \mu)$, determined by covariance matrix Σ^{-1}. The overall cost is computed as a weighted mixture of Gaussians considering the quality measure (force closure) w_i associated with each sample:

$$g_q = \frac{1}{\sum |f_i|} \sum w_i f_i \tag{20}$$

We skip the gradient for brevity. This model has been chosen to account for the noise associated with the chosen force closure value. The noise is mainly due to the discontinuous number of contacts, the determination of the contact points

and some other aspects. The mixture model smoothens the criterion in a local region, such assuring a smooth gradient. This is illustrated in Figure 4, where we plot the grasp quality over a rotation angle of the hand with respect to a basket handle.

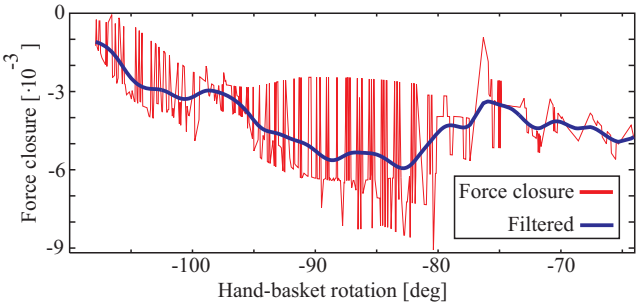


Figure 4:
Raw and smoothened force closure. The force closure is negative for stable grasps.

While the proximity criterion tries to pull the final hand posture towards the task map manifold, the quality criterion only becomes active when the final hand posture is close to the manifolds proximity. It tries to level the target pose toward the best force closure. Figure 5 shows the two cost terms over the time course of a trajectory. The proximity cost decreases towards converging to the task manifold, the quality cost only shows activity when getting close to its proximity.

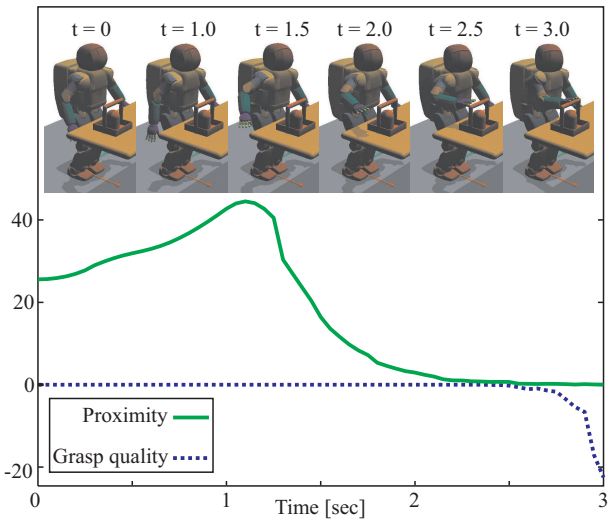


Figure 5:
Proximity and quality costs over a trajectory. During optimization, they are evaluated in the last time step of the movement.

To account for the formulated criteria during optimization, their gradient has to be derived with respect to the state vector according to term g' in eq. (A.10). For this, we can rewrite the differential kinematics as

$$\frac{\partial g_{\text{map}}}{\partial q_t} = \frac{\partial (g_p + g_q)}{\partial x_t^{\text{rel}}} \frac{\partial x_t^{\text{rel}}}{\partial q_t} = \frac{\partial (g_p + g_q)}{\partial x_t^{\text{rel}}} J_{\text{rel}} \qquad (21)$$

with J_{rel} being the Jacobian of the task vector relating to the relative hand-object coordinates of the task map. This means that the task map can be represented in different coordinates than we chose to control the system. We can for instance represent the task map in relative hand-object coordinates and control the movement in global coordinates.

Both quality and proximity terms are only evaluated in the last time step $t = T$ of the trajectory, since this corresponds to the hands final grasp pose. Their effect is back-propagated in time and on the attractor point locations with eqs. (A.11) ff. The nearest neighbor query only needs to be carried out once in each optimization iteration which is advantageous in terms of computational efficiency.

IV. Experiments

We have set up the control model according to the experiments shown in Figure 7. It comprises a model of the humanoid robot ASIMO, a table and a basket with a U-shaped handle. The robots upper body has the 6 rigid body degrees of freedom. Each leg is modeled with 6, and each arm with 5 dofs. The head has 2 dofs (pan and tilt). For the fingers, we introduced a one degree of freedom controller that drives the finger joints according to the coupled spring-tendon mechanism of ASIMO's hand.

In addition to the *proximity and quality criteria* described in Section III, we employ analytical gradients for joint limit avoidance, the overall length of the trajectory in joint space, the difference to a zero-velocity at the end of the movement, and a detailed collision model: The collisions are considered with a computationally efficient shape primitive model [18] of the respective segments. They are incorporated into the optimization with a distance-to-collision gradient presented in [8]. We consider collisions for the hand and finger segments, the lower arms, the body, the thighs and the environment objects table and basket. In our implementation, we can assign a distance threshold for each object pair. The self-collisions between the robot segments as well as the collisions between the robot and the table are penalized if they get closer than 0.1 m. The collisions between hand, fingers and basket are penalized if they get closer than 2 mm. The overall collision model comprises 30 body pairs.

The following task variables are subject to the optimization: Right hand position and orientation (2D, polar angles) between hand and basket. Further, we constrained the horizontal components of the center of gravity and the foot transformations. Another constraint has been added to the gaze direction vector: It will continuously travel towards the center of the top basket handle. Overall, the dimensionality of the task is 21. The employed task map has been sampled with 1000 valid grasps. The best initialization of the trajectory is determined according to Section III-B. Figure 3 illustrates the

three potential target postures that correspond to the respective cluster. The movement is initialized with a sequence of 12 21-dimensional attractor task vectors. The initial attractor sequence is interpolated linearly between the robots initial posture and the task state associated with the mean of the selected cluster. The movement shall be completed within 3 seconds.

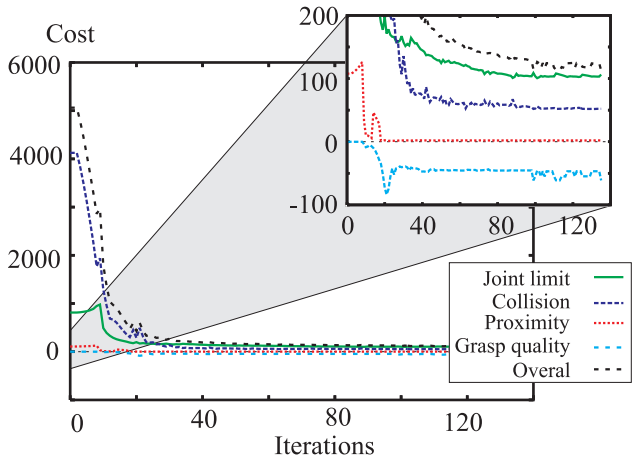


Figure 6: Cost terms during optimization run

The progression of the cost terms during the simulation run is depicted in Figure 6. The costs massively decrease in the beginning, since the initial trajectory leads to many collisions, violates some joint limits and doesn't exactly reach the task map manifold. After a few iterations, it rapidly converges to a minimum. The enlarged subimage image in the figure shows that 1) the target task manifold is exactly reached after approx. 15 iterations (red line) and 2) the grasp quality converges after approx. 25 iterations (light blue line).

Figure 5 shows the proximity and quality cost over the optimal trajectory. It can be seen that initially, the hand moves away from the table edge, and then moves upward in an arc around the table. The finally chosen grasp position is on the left side of the basket handle, which seems to lead to lower costs than to grasp the handle at its center.

Due to the local character of the chosen gradient-based optimization scheme and its parametrization, it turned out that the visually realistic modeling of objects is not always suitable for the collision avoidance mechanism. It should rather be avoided to shape attractor and repeller "sinks". In such cases, the solution sometimes gets stuck in local minima, or "tunnels" through objects, leading to unrealistic results. One example is the collision model of the basket, depicted in the top part of Figure 5. We had to insert a repelling object, since in some cases the hand got trapped under the handle.

We devised a set of experiments applying the scheme to reach for and grasp a basket at different locations on the table. In order to autonomously grasp the basket handle, the robot's fingers have been equipped with tactile sensors based on a force resistive measurement principle (FSR sensors). It turned out to be difficult to sense the initial contact between hand and object. This is due to the sensor response characteristics. It is necessary to exceed a minimum pressure threshold in order to get a signal. This is difficult when touching lightweight objects. We therefore integrated an additional proximity sensor based on a capacitive measurement principle on the palm. It already reacts shortly before the contact, which makes it possible to close the fingers without hitting the object. The proximity measurement depends on the material of the sensed object. We adjusted it to react to the basket at a distance of approx. 1.5 cm. When the basket is sensed, the fingers are automatically closed.

Figure 7 shows three representative movement sequences. In the upper row of the figure, the robot reaches and grasps for the inner handle of the basket. During reaching, it avoids to hit the other handles and finally shifts its hand to the lower part of the handle before closing the fingers. In the middle row, the basket has been moved away in frontal direction. The optimal pregrasp was found on the right end of the baskets top handle. In further experiments, it turned out that the top handle was preferably grasped when the basket was located farther away in front of the robot. In the lower row, the basket was moved to the left side. In this location, the movement optimization was initialized with the right handle. In this case, the hand did not have to move under the top handle, which resulted in a more "relaxed" movement.

V. Conclusion

We presented a framework for fluent reaching and grasping of objects with a humanoid robot. The coupled problem of reaching and grasping is solved by incorporating object-specific *task maps* directly into the movement optimization process. These maps represent a functional characterization of objects in terms of their grasp affordances, i. e. a manifold of feasible pregrasp poses.

We proposed a method to efficiently sample such maps in simulations, and to preprocess them to extract qualitatively different pregrasp postures. The proposed task maps contain both information about the proximity to a valid solution manifold and the quality of the grasp, represented by its force closure. Since the map parameters are computed from values that can be extracted by sensor data, this permits to generate or refine task maps also in real experiments.

The task map concept has been integrated in an online whole body motion control and optimization framework. It efficiently couples collision-free reaching with grasping, leading to a preference for grasps which are easy to reach and exhibit a high quality. A selection mechanism chooses the most appropriate initialization for the movement optimization, such adding a more global perspective.

Future work will focus on integrating computer vision and tactile feedback. Further, we plan to extend the methods to account for additional criteria (e. g. manipulability indices), more complex grasping strategies and bimanual grasping. Since the framework allows to generate movements within

Figure 7: Movements for different basket locations

interaction time, we will also focus on extending the scheme to dynamic scenes with moving objects, walking and human interaction.

ACKNOWLEDGMENT

The authors thank the members of Honda's ASIMO and robot research teams in Japan for their support.

REFERENCES

[1] R. Suárez, M. Roa, and J. Cornellà, "Grasp quality measures," Tech. Rep. IOC-DT-P 2006-10, Universitat Politècnica de Catalunya, Institut d'Organització i Control de Sistemes Industrials, 2006.

[2] D. Schwammkrug, J. Walter, and H. Ritter, "Rapid learning of robot grasping positions," in *Proceedings of the International Symposion on Intelligent Robotics Systems (SIRS)*, 1999.

[3] R. Haschke, J.J. Steil, I. Steuwer, and H. Ritter, "Task-oriented quality measures for dextrous grasping," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 6 2005, pp. 689 – 694.

[4] A.T. Miller, S. Knoop, H.I. Christensen, and P.K. Allen, "Automatic grasp planning using shape primitives," in *Proceedings of the IEEE International Conference of Robotics and Automation (ICRA)*, 2003, pp. 1824 – 1829.

[5] R. Platt, A.H. Fagg, and R. Grupen, "Improving grasp skills using schema structured learning," in *Proceedings of the International Conference on Development and Learning (ICDL)*, 5 2006.

[6] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner, "Grasp planning in complex scenes," in *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, 12 2007.

[7] X. Joiang and M. Kallmann, "Learning humanoid reaching tasks in dynamic environments," in *IEEE International Conference on Intelligent Robot and Systems (IROS)*, 11 2007, pp. 1148 – 1153.

[8] M. Toussaint, M. Gienger, and Ch. Goerick, "Optimization of sequential attractor-based movement for compact movement representation," in *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, 12 2007.

[9] M. Toussaint and Ch. Goerick, "Probabilistic inference for structured planning in robotics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[10] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 3229–3236.

[11] M. Gienger, M. Toussaint, and Ch. Goerick, "Task maps in humanoid robot manipulation," in *Accepted: Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS)*, 9 2008.

[12] M. Gienger, H. Janssen, and Ch. Goerick, "Task-oriented whole body motion for humanoid robots," in *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, 12 2005.

[13] J. Steil, F. Roethling, R. Haschke, and H. Ritter, "Situated robot learning for multi-modal instruction and imitation of grasping," *Robotics and Autonomous Systems*, vol. 47, no. Special Issue on Robot Learning by Demonstration, pp. 129–141, 2004.

[14] R. M. Murray, Z. Li, and S. S. Sastry, *Robotic manipulation*, CRC Press, Florida, USA, 1994.

[15] M. Gienger, H. Janssen, and Ch. Goerick, "Exploiting task intervals for whole body robot control," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS)*, 12 2006.

[16] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," in *IEEE Transactions on Systems, Man, and Cybernetics*, 12 1977, vol. SMC-7 no. 12.

[17] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *Journal of the ACM*, vol. 45, no. 6, pp. 891–923, 1998.

[18] H. Sugiura, M. Gienger, H. Janssen, and C. Goerick, "Real-time self collision avoidance for humanoids by means of nullspace criteria and task intervals," in *Proceedings of the 2006 5th IEEE-RAS International Conference on Humanoid Robots*, Genova, Italy, 2006, pp. 575–580.