Modeling and Planning High-Level In-Hand Manipulation Actions from Human Knowledge and Active Learning from Demonstration

Urbain Prieur†, Véronique Perdereau† and Alexandre Bernardino‡

***Abstract*— We propose a method to plan in-hand manipulation actions with a robotic anthropomorphic hand. We consider in-hand manipulation actions as sequences between canonical grasp types identified in the humans. Our work concerns the generation of this sequence, which should be autonomous and fast enough to be performed on-line. We use a Markov Decision Process (MDP) governing the transitions between grasp types, depending on the object and on the goal grasp. The policy is learnt directly from human behavior, after an initialization using an empirical estimation of the state action probabilities of the MDP. Then, the policy is finely learnt from samples of human in-hand manipulation records. These samples are chosen using active learning, in order to maximize the useful information of every record, and speed up the learning process. For planning, the policy gives the sequence with highest probability of success. We show a serie of realistic human-like grasp transition sequences derived from the proposed method.**

I. Introduction

The work presented in this paper concerns the planning of movements to make a multi-fingered robotic hand execute in-hand manipulation. Some situations require in-hand movements to execute fine actions, such as pressing a button, unlocking a key or opening a bottle. In-hand movements are also used to regrasp an object, because of a wrong initial pose, a problem of accessibility, or to apply specific manipulation actions like rolling the object at fingertips or performing finger gaiting to relocate or substitute fingers. A useful robotic hand should autonomously decide what to do with a given object, provided the high level objective (task) is known, and execute human-like movements, while adapting to the world context in real time.

As most of the work on this subject [1]-[12], we consider in-hand manipulation tasks decomposed in a two level hierarchy. The lower level focuses on the continuous control of the manipulation physical aspects (finger motions, contacts, forces) in order to achieve desired hand-object configurations. The higher level decomposes a manipulation task into discrete primitive actions, and focus on the rules for composing these actions to execute elaborate tasks. Due to the complexity of in-hand manipulation it is essential to choose a suitable set of primitive actions. A too abstract set will demand complex low-level controllers, whereas very elementary primitive actions will increase the complexity of the higher level. We adopt a representation based on canonical grasps identified in humans [2] and model in-hand manipulation actions as sequences of canonical grasps. As [3] suggests, *transitions between canonical grasps are the key to in-hand manipulation*.

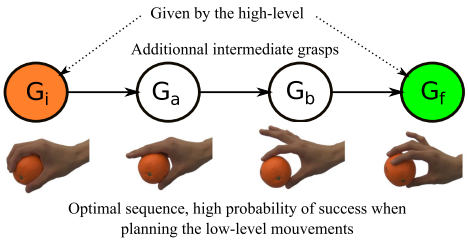

Fig. 1. Grasp sequence may require more or less intermediate grasps.

Dominant approaches to high-level in-hand manipulation use graph search techniques: a graph is first built and a path is searched in the graph to link initial and final configurations. The graph can represent the feasible transitions between either hand states (finger configurations) or object states [1] [4] [5]. Both require an additional lower level planner to check if a transition can be performed according to the constraints imposed by the object and by the fingers, respectively. The graph is constructed in a preliminary phase, and contains the nodes relevant to a specific task, which dramatically limits its versatility and adaptability.

Other solutions set up hybrid system control techniques: the hand-plus-object manipulation planning is represented as a hybrid system planning problem combining continuous as well as discrete aspects. The control law is then in charge of driving the system from the current state to a desired state through the evolution of its continuous state variables. One hybrid representation is in the form of a stratified system of the whole configuration space [6] [7] [8], each strata containing only configurations where specific fingers are in contact. The main difficulty is to determine links between strata that are necessary to resolve any manipulation problem. Another hybrid representation is an hybrid automaton [9] that divides a manipulation task into a sequence of sub-manipulations described by continuous variables and separated by discrete transitions (switches between nodes). In all these works the contact positions are considered as fixed which does not allow rolling (nor sliding) motions at fingertips.

In another approach, probabilistic techniques are used to plan in-hand manipulation. Probabilistic path planning methods are applied to the hand considered as a whole system, randomly sampling hand-plus-object configurations

The research leading to these results has been supported by the HANDLE project, which has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT 231640.

†UPMC Univ Paris 06, UMR 7222, ISIR, F-75005, Paris, France (e-mail: {urbain.prieur,veronique.perdereau}@upmc.fr).

‡ISR, Instituto Superior Técnico, Technical University of Lisbon, 1049-001 Lisbon, Portugal (e-mail: alex@isr.ist.utl.pt).

to build a tree-like graph in a high-dimensional configuration space. Proposed solutions are all variants of the classical PRM or RRT methods [10], with some modifications aimed at reducing the search space when determining the path [11], [12], [9]. These methods still remain time consuming. The probabilistic nature of the solution very often leads to unwanted movements: the generated hand motion does not look natural, and is far from being optimal.

The solution we consider in this work is also a probabilistic model but its constitutive elements are different. States are not high-dimensional hand-plus-object configuration vectors but discrete canonical grasps, said grasp classes in the grasp taxonomy described in [2], simply defined by numbered labels. Transitions are not exhaustively validated through numerous tests related to hand mechanics, object, task and environment, but initialized from humans empirical knowledge and refined through learning from human trials. Tasks are encoded as desired states in the probabilistic model whose transition probabilities represent the plausibility of human-like grasps. Object information is encoded in a discrete set analogous to [13] which allows a parsimonious representation. The simplicity of the proposed model allows the system to quickly sketch a realistic plan of actions in run-time, in the form of a grasp sequence (see Fig.1). Lower level planners/controllers will then just have to check for a limited number of grasp transitions. This method tackles the complexity of an in-hand manipulation movement, by dividing it into short steps from a stable grasp to another.

At first, the method used is explained (II), starting with description of the MDP modeling in-hand manipulation (II-A and II-B). Then, it is explained how a policy is modeled (II-C), and which is the optimal one (II-D). Then, in (III), the policy is first given a coarse estimation (III-A), and subsequently learnt from human movements(III-B). Results obtained so far are then described (IV), before concluding and presenting future work (V).

II. A MODEL OF HIGH-LEVEL IN-HAND MANIPULATION

The starting point for our model is the representation of manipulation actions by a sequence of stable grasp configurations, linked by local movements making the hand-plus-object reaching the next grasp configuration. A stable grasp is a configuration in which the hand holds the object. Motivated by studies in physiology, [2] has proposed a set of canonical grasp types that are the most frequently observed in human in-hand manipulation. The set includes power, precision and intermediate grasps involving a variable number of digits and contact surfaces (tips, pulp or lateral surfaces of the fingers). A few examples of the taxonomy can be seen in Fig.(7)-(9)-(10). Let us define the canonical grasp set as:

$$\mathscr{G} = \{g_1, \cdots, g_N\} \tag{1}$$

with $N = 33$ as in [2]

Transitions between grasp types are produced by human/robot actions. Let us consider the action set:

$$\mathscr{A} = \{a_{ij}\}, i, j \in \{1, \cdots, N\} \tag{2}$$

Each a_{ij} is the action primitive that drives the hand configuration from g_i to g_j. An action a_{ij} can only be applied if the hand is in configuration g_i. Depending on the grasp types and manipulated objects, these transitions may be hard or easy, comfortable or uncomfortable, efficient or not in terms of energy, prone to failures or not, both in humans and in the robot. An assumption of our work is that robot dexterous hands tend to match human capabilities, therefore human-like actions will mirror their characteristics to the robotic implementation, i.e. human-like actions executed in the robot will be more efficient, easy and less prone to failures.

Object characteristics are defined in prototypical object shapes. A highly accurate model is not required for this high level of planning, unlike planning the fine movements of fingers, dealing with contact and stability of a grasp. In [13], 6 basic shapes are defined as combinations of 2 discrete shapes (flat, rounded) and 3 sizes (small, long, large). In our work we consider a set of 9 objects that are typical of everyday use (Fig.2).

Fig. 2. The set of representatve objects used for this work. The set of object is: key, mug handle, mug body, pen, phone, ball, ladle, book, bottle.

In general, a set of M object types is defined:

$$\mathscr{O} = o_m, m \in \{1, \cdots, M\} \tag{3}$$

Given the above definitions, we are ready to model our problem as an object dependent Markov Decision Process.

A. Markov Decision Processes

A Markov Decision Process (MDP) is a Dynamical Bayesian Network that models the evolution in time of the state of an agent according to the actions taken and the dynamics of the environment. In our problem we consider discrete states and actions. At each time step the agent receives a reward or penalty from the environment reflecting the successful (or not) accomplishment of tasks (see Fig.(3)). A MDP is usually denoted as a 4-tuple $(\mathscr{S}, \mathscr{A}, \mathbf{P}, R)$, that in our case is defined as:

- $\mathscr{S}$ is the set of possible states. At each time step t, the state of the agent is represented by a random variable S_t that can have values in $\mathscr{S}$. In our problem the set of possible states include the canonical grasp types g_i and a null state ϕ representing failures, infeasible or non-human like transitions. Thus, $\mathscr{S} = \mathscr{G} \cup \phi$.

- $\mathscr{A}$ is the set of available actions. In our case it is the same set as defined in (2). A random variable A_t represents the action taken at time t.
- **P** is a probabilistic transition model. It models the time evolution of the state according to the executed action. It is usually represented as a probability distribution over next states S_{t+1} give the current state S_t and action A_t: $P(S_{t+1}|S_t,A_t)$. In our case it also depends on the object being manipulated, therefore: $P(S_{t+1}|S_t,A_t,O)$, where O is the random variable taking values in the set $\mathscr{O}$.
- R is a reward signal collected by the agent at each time step. It depends on the MDP state, the action or both.

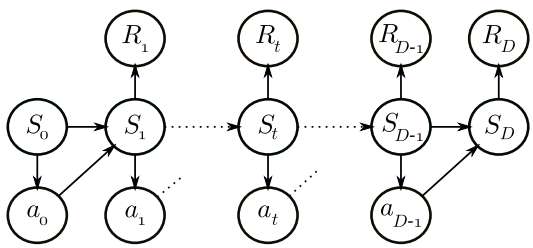

Fig. 3. A general MDP of duration D, with a reward signal.

The above definitions configure an object dependent MDP. In particular, for each object in our set, the probability transition model have different parameters. Given the reasonably low number of object types, this is not a very limiting fact of our model. In particular, due to the nature of our problem, there are several constraints on the probability transition model that limit the number of required parameters.

- Because some actions can only be applied in certain states, we have, $\forall i,j,k,l, k \neq i$ or $l \neq j$:

$$P(S_{t+1} = g_j | S_t = g_i, A_t = a_{kl}, O) = 0 \tag{4}$$

- As in the case above, $k \neq i$ or $l \neq j$, infeasible actions lead with certainty to the null state:

$$P(S_{t+1} = \phi | S_t = g_i, A_t = a_{kl}, O) = 1 \tag{5}$$

- The null state is an absorbing state, i.e. when the system is in the absorbing state, it will stay on that state forever[1]. $\forall i,j,k$:

$$P(S_{t+1} = \phi | S_t = \phi, A_t = a_{kl}, O) = 1 \tag{6}$$
$$P(S_{t+1} = g_i | S_t = \phi, A_t = a_{kl}, O) = 0 \tag{7}$$

- An action that is feasible and leads to the same state cannot fail:

$$P(S_{t+1} = g_i | S_t = g_i, A_t = a_{ii}, O) = 1 \tag{8}$$
$$P(S_{t+1} = \phi | S_t = g_i, A_t = a_{ii}, O) = 0 \tag{9}$$

[1]This can be interpreted as: if a grasp transition is not feasible, not human-like or fails, the whole sequence is unfeasible, not human-like or failed.

- Finally, feasible transitions actions can have two outcomes: either the transition succeeds or it fails[2]:

$$P(S_{t+1} = g_j | S_t = g_i, A_t = a_{ij}, O = o_m) = p_{ijm} \tag{10}$$
$$P(S_{t+1} = \phi | S_t = g_i, A_t = a_{ii}, O = o_m) = 1 - p_{ijm} \tag{11}$$

The values of p_{ijm} are thus the important parameters to define in our model. They encode the success likelihood of a transition for an object. There are at most $(N-1) \times (N-1) \times M$ such parameters. Learning from scratch all these parameters is too costly, therefore we use human empirical knowledge to initialize a coarse model, as presented in III-A.

The duration D of the MDP has to be known, so that $t \in \{0, \cdots, D\}$. We know from human observation that sequences of grasps for in-hand manipulation are never longer than 6 grasps, thus we fix the duration to $D = 5$. Shorter sequences fill the missing steps by nature, using actions that keep the system under the same state with an unconditional success (illustrated on Fig.(4)).

We want the MDP to be in the goal state $S_D = g_f$ at the last time step of the MDP, we then define the reward depending on the state and on the time step, $R_t(S_t)$: $R_D(g_f) = 1$, $R_t(S_t) = 0$ if $S_t \neq g_f$ or $t \neq D$. Giving a reward only on the final time step avoids that sequences remaining in the goal state for multiple time steps accumulate rewards.

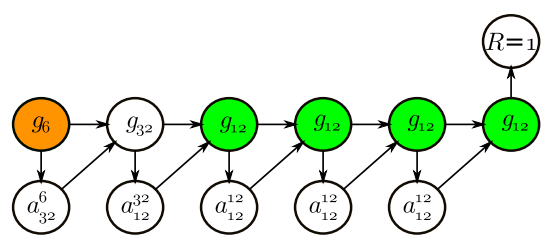

Fig. 4. An instantiation of a MDP modeling a grasp sequence, valid for a single object. The reward is 1 if the final state is the goal grasp. The grasp sequence illustrated here is shorter than the length of the MDP. Actions keeping the system under the same state fulfill the remaining time steps, only obtaining the reward on the last time step.

B. Tasks and Rewards

In-hand manipulation tasks consist in the application of forces and velocities in a certain object, whose performance depends on the the way the hand grasps the object. For instance the use of a spoon to stir coffee is more efficient when the spoon is held with a tip pinch; the use of a pen to write is more efficient with a writing tripod type of grip. In this paper we concentrate on in-hand action sequences required to achieve a desired grip on the object. Starting from an initial grasp type g_o, our objective is to determine the most human-like sequence of intermediate grasps to achieve the task: reaching g_f. This can be encoded in an MDP by defining a reward function structure that promotes the achievement of the task. As previously mentioned, without any further constraints on the task, this can be implemented

[2]failure is taken *latu sensu*, i.e. the object can fall, the next state is unstable or the transition leads to a non desired state.

by a reward function that assigns value 1 to the goal state in the end of the sequence and 0 to all others states ($R_D = 1$ if and only if $S_D = g_f$, where D is the duration of the MDP).

With the aforementioned constraints, and assuming an arbitrary sequence of actions $\mathbf{A} = \{A_0, \cdots, A_{D-1}\}$, the expected reward is given by:

$$\bar{R} = E\left[R(S_D)\right] = P(S_D = g_f | S_0 = g_o, \mathbf{A}, O) \tag{12}$$

This function has a recursive nature and can be more adequately represented through the definition of a Value function for each state, $V(g_o)$ representing the expected reward obtained if starting at an arbitrary state g_o.

$$V(g_o) = P(S_D = g_f | S_0 = g_o, \mathbf{A}, O)$$
$$= \sum_k P(S_1 = g_k | S_0 = g_o, A_0 = a_{ik}, O) V(g_k)$$

Because of the constraints on the state-action transitions defined in our formulation (equations (4)-(9)), i.e. action a_{ij} only succeeds in the transition from g_i to g_j, with probability p_{ijm}, the only action sequences that return rewards are the ones that form chains starting in the initial state and terminating in the goal state. We define a feasible chain, for initial state g_o and final state g_f, as the ones with $A_0 = a_{ok}$, $A_{D-1} = a_{lf}$ and whose end state of A_t is the initial state of A_{t+1}. Constraining the action sequence to a feasible chain, the sum on the previous recursion disappears:

$$V(g_o) = P(S_1 = g_k | S_0 = g_o, A_0 = a_{ok}, O) V(g_k) \tag{13}$$

Unfolding the recursion until the goal state is reached, we can verify that the state value function is given by the transition probabilities along the chain:

$$V(g_o) = P(S_1 = g_k | S_0 = g_o, A_0 = a_{ok}, O) \times$$
$$\times \prod_{t=1}^{D-2} P(S_{t+1} = g_j | S_t = g_i, A_t = a_{ij}, O) \times$$
$$\times P(S_D = g_f | S_{D-1} = g_l, A_{D-1} = a_{lf}, O)$$

Thus, given a feasible chain of actions $\mathbf{A} = \{A_0, \cdots, A_{D-1}\}$, we can compute the expected reward by simply multiplying the p_{ijm} probabilities along the chain.

C. Stochastic Policies

It is common to represent the decision making process as a policy function that chooses actions according to the current state. Given the desired object (O) and task (T), we write such a policy as $\pi^{OT}(A_t, S_t)$, and it represents a probability distribution over actions at each state. This reflects the probability that an agent performs some action A_t at state S_t, which depends on the current object and task:

$$\pi^{OT}(A_t, S_t) = P(A_t | S_t, O, T) \tag{14}$$

The rationale for the adoption of a stochastic policy model (probability distribution over actions) instead of a deterministic one (single choice on a given state) has to do with the fact that humans have a certain variability in the choices of actions that demand a probabilistic representation, not only for learning purposes but also to plan the recording of human trials. Under probabilistic representations, active learning techniques can be used to design experiments that maximize expected model improvement through learning (III-B.2).

The state value recursion, under a particular policy, is now an expected value over the action distribution induced by the policy :

$$V^{\pi^{OT}}(g_o) = \sum_k \pi^{OT}(a_{ok}, g_o) \times$$
$$\times P(S_1 = g_k | S_0 = g_o, A_0 = a_{ok}, O) V^{\pi^{OT}}(g_k)$$

As (14) suggests, this policy can be encoded as a Bayesian Network $P(A_t | S_t, T, O)$ (see Fig.(5)), where the action node depends on the parent nodes representing the current state, the task (goal state), and the manipulated object. This consists in a multinomial table with a set of weights:

$$q_{ijfm} = P(A_t = a_{ij} | S_t = g_i, T = g_f, O = o_m) \tag{15}$$

The process to initialize and learn this table is shown in (III).

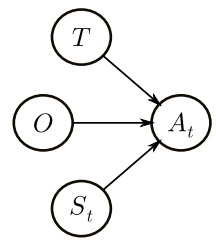

Fig. 5. Bayesian Network modeling the policy of the MDP generating modeling grasp sequences. A_t is the action at time t, S_t is the state at time t, O is the object and $T = g_f$ is the task.

D. Optimal Policies

Optimal policies are the ones that maximize the expected reward. At each time step, the agent should choose an action that leads to a sequence of states with maximal pay off, i.e. maximum state value:

$$a_{ok} = \text{argmax}_k P(S_1 = g_k | S_0 = g_o, A_0 = a_{ok}, O) V^{\pi^{OT}}(g_k)$$

This is the so called greedy approach, that actually leads to the optimal policy $\pi^{*^{OT}}$ and optimal state value function $V^{*^{OT}}$ [14]. We can define, for our setting, the optimal action-state value function $Q^*(A, S)$ representing the expected reward of choosing action A in state S:

$$Q^*(A_0 = a_{ok}, S_0 = g_o) =$$
$$P(S_1 = g_k | S_0 = g_o, A_0 = a_{ok}, O) \times \max\left(V^{*^{OT}}(g_k)\right)$$

The optimal policy should choose actions according to the values of $Q^{*^{OT}}$. In a stochastic setting, the higher the value of $Q^{*^{OT}}(A, S)$ is, the highest the probability of choosing action A in state S should be. In this work we choose:

$$\pi^{*^{OT}}(A, S) \sim Q^{*^{OT}}(A, S) \tag{16}$$

Using this policy, grasp sequences with highest probability of reaching the goal state are generated.

III. Initialization and learning

A. Initializing the policy

For initializing the policy, we use the Markov Decision Process (MDP) defined in II-A. A human expert estimates empirically (III-A.1) the state transition probability values, p_{ijm}, and the policy is generated accordingly (III-A.2).

1) Empirical estimation of the state transition probability: To give coarse values to p_{ijm}, we realize an incremental approach, starting by averaging out the object influence, creating object independent model, with parameters p_{ij}. The obtained model is used in subsequent work to initialize the multinomial tables depending on the object, creating a complete model with parameters p_{ijm}. To give the object independent model estimations, we first classify the transitions in terms of difficulty, or probability of success. A human expert is considering every transition from grasp g_i to grasp g_j, and assesses the difficulty using three classes: possible and easy (class a), possible and complex (class h), or impossible (class N).

As the generic model is object independent, we have to avoid object influence. Thus, we use a set of objects, but every possible combination on any object is considered, and the easiest will determine the class of the transition, to avoid restricting the possibilities. For example, if a transition can be performed with a complex combination of actions with object A, and can be performed with a complex and an easy combination of actions with object B, the transition is classified as easy, so that no unwanted limitation appears.

To evaluate the difficulty of a transition, the number and type of sub-actions composing it are considered. We refer to the canonical in-hand movements defined in [15]. It presents an exhaustive list of in-hand sub-actions, to which we add finger gaiting, consisting in repositionning a finger, making it stop contacting the object. The rules established for the classification process are simple and intuitive. To convert this classification into probabilities, we choose to give a sensible probability of success to difficulty classes. A transition is:

- impossible: if no direct transition is possible, i.e. any tentative of transition from grasp g_i to grasp g_j uses an intermediate grasp (as defined in the grasps list), or if the object and hand contact is stopped. Probability of success: $p_{ij} = 0$
- easy: if the direct transition from grasp g_i to grasp g_j uses a single action. Probability of success: $p_{ij} = 0.8$
- complex: if the direct transition from grasp g_i to grasp g_j uses a combination of two actions or more. During the transition, if the object is in contact with the hand on one plane surface, even briefly, and then in an unstable state, the transition is considered as complex: $p_{ij} = 0.2$

On a second step, the influence of the object is taken into account. It appeared during the preliminary analysis that the object influences the possible grasps in a binary way: a grasp is possible to perform on an object, or not possible at all. This allowed the following analysis: impossible grasp classes for every object has been listed, and the probability of success of transitions using these grasps set to 0.

Now that the probability of success of every transition has been assessed, we can generate a policy accordingly.

2) Policy generation: The policy is made of multinomials (15): $q_{ijfm} = P(A_t = a_{ij}|S_t = g_i, T = g_f, O = o_m)$. For every set of parents (S_t, T, O), we try every feasible chain of actions (13), and compute the $Q^{*OT}(A,S)$ (16) using the MDP with the values of p_{ijm} obtained in (III-A.1). We update q_{ijfm} as $Q^{*OT}(a_{ij}, s_i)$. To have a multinomial distribution, these values are normalized over A_t, so that $\sum_j (q_{ijfm}) = 1$.

We obtain a policy that is used as initial policy, and refine it using learning from the observation of human policy.

B. Learning the policy from humans

1) Learning: To update the parameters with experiments, we use multinomial updates with Dirichlet priors [16]. The method consists in counting how many times an action a_{ij} has been observed: $N_{ij}(S_t, T, O)$ in a set of $N_i(S_t, T, O)$ when in state S_t, with task T and object O. A set of positive hyperparameters $h_{ij}(S_t, T, O)$ ($h_i(S_t, T, O) = \sum_j h_{ij}(S_t, T, O)$) defines the prior knowledge on the process, in our case, the initialization: $h_{ij}(g_i, g_f, o_m) = q_{ijfm} \cdot h_i(g_i, g_f, o_m)$. Intuitively, $h_i(S_t, T, O)$ is the number of imaginary cases in which each set of parents (S_t, T, O) has been observed, and can be used by the human expert to encode its certainty on the process. If very certain, this number should be high, otherwise, low. Then the probability of a transition is given by:

$$q_{ijfm} = P(a_{ij}|g_i, g_f, o_m) = \frac{h_{ij}(g_i, g_f, o_m) + N_{ij}(g_i, g_f, o_m)}{h_i(g_i, g_f, o_m) + N_i(g_i, g_f, o_m)} \tag{17}$$

We can update the policy using (17), choosing the importance of the empirical knowledge compared to the knowledge obtained from learning by setting the number of imaginary samples corresponding to the initialization $h_i(S_t, T, O)$, $\forall i$.

2) Active learning: This can be done through an arbitrary batch of human recordings, however it is not efficient, and given that each human recording is time-consuming, we choose another solution. We use the possibility of choosing what values the parent variables ($S_t = g_i$,$T = g_f$ and $O = o_m$) will be in the samples. The learning process is therefore guided by the information obtained previously. Active learning consists in choosing the training that would most improve the system knowledge by learning. An interesting method to do this is presented in [17]. A function representing the probable reduction of risk of loss of information is calculated for each set of parents (a query), and the one most reducing the risk is selected and sampled in the next experiment.

We use a similar technique, but not using the same function. Our previous knowledge is obtained from an empirical analysis, assumed sufficient for giving the appropriate policy for obvious cases, where one solution is highly more likely to succeed than the others. Learning is preferably used to finely tune the policy in the cases where it is unclear whether appropriate solution should be chosen. The concept of entropy is used to describe this uncertainty. For every set of parents, we compute the entropy over feasible actions a_{ij}:

$$H(g_i, g_f, o_m) = \sum_j -(q_{ijfm}) \cdot \ln(q_{ijfm}) \tag{18}$$

To make the entropies comparable between sets of possible actions of different size, we normalize the entropies using the entropy of the uniform ditribution of each size, and obtain comparable entropy parameters (after simplification):

$$H'(g_i, g_f, o_m) = \frac{H(g_i, g_f, o_m)}{ln(1/j)} \tag{19}$$

The sets of parents of highest entropy parameters H' are the next samples to record from human demonstration.

IV. Results

In this part is described the work that has been done following the method presented previously. The empirical initialization explained in (III-A) has been achieved. At first, we performed the expert guess of grasp transition probabilities of success (III-A.1), independently of the object. This gives a Grasp Transition Graph (Fig.(6)) showing an interesting fact: four groups of grasps have been identified.

- groupA : each grasp from groupA is connected to grasps from every other groups. These grasps are likely to be key grasps for in-hand manipulation.
- groupB: grasps from groupB are all connected to each other, and only connected with grasps of groupA.
- groupC and groupD: grasps these groups are not connected to grasps from other groups, exept from groupA.

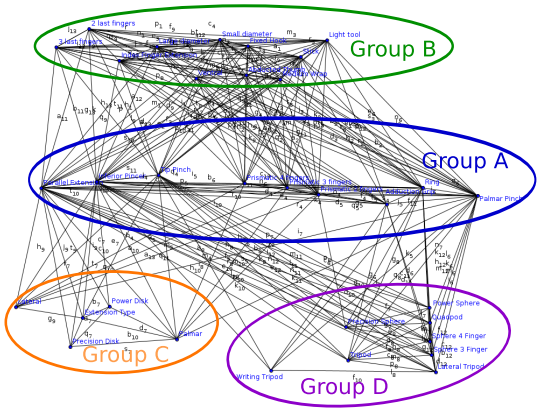

Fig. 6. Object independent Grasp Transition Graph resulting from the empirical analysis. Each transition has an associate probability.

We conducted the analysis of the possible grasps for each object, extending the validity of the probabilities to the whole set $\mathscr{O}$. The estimated values have been used as state transition probabilities of the MDP described in (II-A), in order to compute an initialized policy, following the method presented in (III-A.2). Using this policy, grasp sequences can be generated, as shown on Fig.(7).

We performed the active learning technique described in (III-B). The effects of the learning on the policy is a good estimation of the quality of the initialization. We used the value for the imaginary number of samples associated to the empirical knowledge: $h_i(S_t, T, O) = 30, \ \forall i$. The learning process has been performed for an object of the set: the ball. We used the 20 sets of parents of highest entropy, as

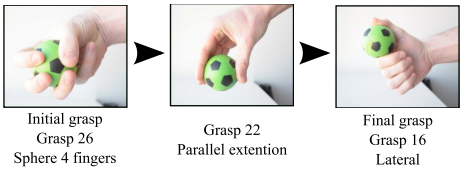

Fig. 7. A generated sequence for a specific object and goal grasp, from an initial grasp, using the initialization policy.

explained in (III-B.2), and recorded movements of 5 human subjects for these sets. After learning from these samples, the entropy of every set of parents has been updated, and the variation of entropy before and after learning ($\Delta_{entropy}$) observed. These results are shown on the Fig.(8).

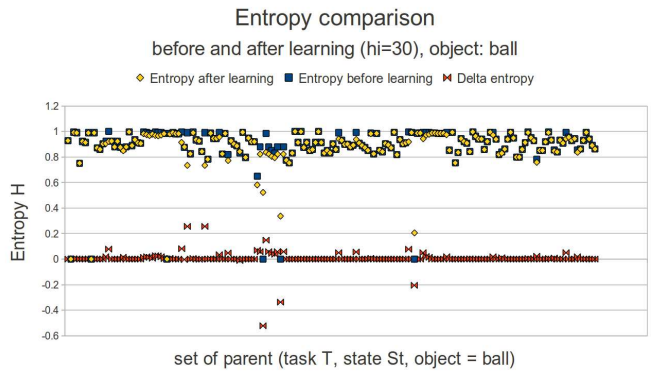

Fig. 8. Histogram of the entropy of every mission for the object: "ball", after the initialization policy. The blue values are before learning, and the yellow values are after learning the selected samples. The red values are the difference of entropy. More than 20 sets of parents are modified, as every transitions in a sequence modify its corresponding set of parents.

The entropy is slightly decreased by the learning process for most of the sets of parents, confirming the policy in its choice. The entropy is increased ($\Delta_{entropy} < 0$) for three sets, whose entropy was 0 after initialization. These sets had only one possible next action before. The learning process has shown other possible actions, increasing the entropy, but giving the system new actions to consider. These sets were updated through intermediate transitions that update their respective sets of parents. Except for these transitions, the learning process has proven that the initialized policy is human-like, allowing us to use it efficiently for planning in-hand manipulation. Some learning is however profitable, as shown by the entropy increase of a few set of parents. As an example to illustrate the effect of learning, we show one of the few sequences which generation has been modified by the learning process in Fig.(9).

Some sequences have been executed with a anthropomorphic robotic hand and arm. The lower level planning the in-hand actions is for now really simple: it is an interpolation of the joints angles from a grasp to another. Due to the simplicity of the transition planning, failures are frequent, but integrating a state of the art low level planner would improve greatly this demonstration. The photos shown in Fig.(10) illustrate an experimental sequence execution. The

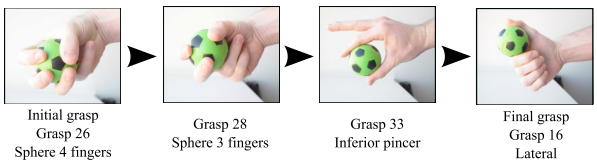

Fig. 9. A generated sequence for the same object, goal grasp and initial grasp as shown in Fig.(7), but using the policy after learning.

generation of a grasp sequence using the policy is extremely fast, planning and replanning grasp sequences online during an in-hand manipulation action are consequently possible.

V. Conclusion

We have presented a model for the high level planning of human-like in-hand manipulation activity for a dexterous robotic hand. A sequence of grasp taypes is autonomously generated to reach the goal grasp from any initial grasp, and in a negligible computation time. The choice of the successive grasps is modeled as a Markov Decision Process, object dependent and valid for any task, using the probability of success of the transitions between canonical grasp types to generate a policy. The work presented in this paper shows how the policy is learnt from demonstration of human movements, assessed as optimal. The policy is modeled as a Bayesian network, used to generate the sequences, for any task and object among a set of 9 objects. An initialization of the policy from an empirical analysis of the probability of success of transitions is provided to avoid starting from scratch. Then, an active learning technique selects the next learning samples with highest interest for the system.

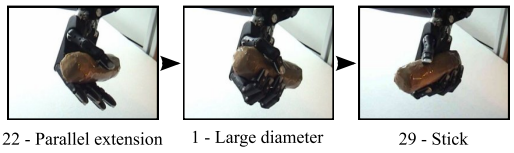

Fig. 10. A grasp sequence generated automatically from grasp 22 to 29, performed by the Shadow hand

For now, we have started the learning process for only one object and a single set of samples, in order to prove its concept and evaluate its efficiency. The results obtained with the learning set highlight the quality of the initialization, and confirm the model improvement. Human plausible grasp sequences are generated instantly, illustrating the efficiency and interest of the method. In a further work, the learning will be continued, tuned and extended to other objects. An analysis of the learning progress would allow to estimate when it should be stopped. A low-level planner able to control the hand from a grasp to another is required to successfully perform a movement. Such a work combined with the method presented in this paper would make a robot able to perform in-hand manipulation autonomously, provided the task is know.

On a different way, the state transition probabilities of the MDP can be found, allowing to generate the policy, and not use a predefined one. This can be done from the robot's own experience, trying to perform every transitions. This would avoid the gap between a robotic anthropomorphic hand and a human hand, with the drawbacks of a heavy experimental process. Supervised learning from humans for state transition probabilities is impossible to do directly because probability of success of transitions cannot be observed from humans, but it would be interesting to investigate indirect techniques to infer state transition probabilities from the human policy. Using the MDP, the generation of a sequence can even be faster if the results are precomputed and stored. By then, the formulation as a MDP allows the extension of the method by enhancing the state used, now only the grasp and object identification numbers. The reward, now goal oriented, can be used to take into account other task related constraints.

References

[1] M. Cherif and K. K. Gupta, "Planning quasi-static fingertip manipulations for reconfiguring objects," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 837–848, 1999.

[2] T. Feix, "The generation of a comprehensive grasp taxonomy," KTH Royal Institute of Technology, Tech. Rep., 2009. [Online]. Available: http://web.student.tuwien.ac.at/ e0227312/

[3] H. Zhang, K. Tanie, and H. Maekawa, "Dexterous manipulation planning by grasp transformation." IEEE International Conference on Robotics and Automation, 1996.

[4] Y. Guan and H. Zhang, "Kinematic feasibility analysis of 3-d multifingered grasps," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 507–513, 2003.

[5] G. Vass, "Object manipulation planning for dextrous robot systems," Ph.D. dissertation, Budapest University of Technology and Economics, 2005.

[6] B. Goodwine, "Stratified motion planning with application to robotic finger gaiting," in *IFAC World Congress*, 1999.

[7] Y. Wie and B. Goodwine, "Stratified motion planning on nonsmooth domains with robotic applications," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 128–132, 2004.

[8] I. Harmati, B. Lantos, and S. Payandeh, "On fitted stratified and semi-stratified geometric manipulation planning with fingertip relocations," *The International Journal of Robotics Research*, vol. 21, no. 5-6, pp. 489–510, 2002.

[9] J. Xu, T.-K. J. Koo, and Z. Li, "Sampling-based finger gaits planning for multifingered robotic hand," *Autonomous Robots*, vol. 28, no. 4, pp. 385–402, 2010.

[10] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Iowa State University, Tech. Rep., 1998.

[11] J.-P. Saut, A. Sahbani, S. El-Khoury, and V. Perdereau, "Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 2907–2912.

[12] M. Yashima, "Manipulation planning for object re-orientation based on randomized techniques," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2004, pp. 1245–1251.

[13] D. Lyons, "A simple set of grasps for a dextrous hand," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, Mar. 1985, pp. 588 – 593.

[14] C. Szepesvári, "Algorithms for reinforcement learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 4, no. 1, pp. 1–103, Jan. 2010.

[15] J. Elliot and K. Connolly, "A classification of manipulative hand movements," *Developmental Medicine & Child Neurology*, vol. 26, pp. 283–296, 1984.

[16] D. Heckerman, "A tutorial on learning with bayesian networks," Microsoft Research, Tech. Rep. MSR-TR-95-06, March 1995.

[17] S. Tong and D. Koller, "Active learning for parameter estimation in bayesian networks." NIPS, 2001, pp. 647–653.