

Learning Pouring Skills from Demonstration and Practice

Akihiko Yamaguchi^{1,2}, Christopher G. Atkeson¹, Scott Niekum¹, and Tsukasa Ogasawara²

Abstract—This paper focuses on improving performance with practice for tasks that are difficult to model or plan, such as pouring (manipulating a liquid or granular material such as sugar). We are also interested in tasks that involve the possible use of many skills, such as pouring by tipping, shaking, and tapping. Although our ultimate goal is to learn and optimize skills automatically from demonstration and practice, in this paper, we explore manually obtaining skills from human demonstration, and automatically selecting skills and optimizing continuous parameters for these skills. Behaviors such as pouring, shaking, and tapping are modeled with finite state machines. We unify the pouring and the two shaking skills as a general pouring model. The constructed models are verified by implementing them on a PR2 robot. The robot experiments demonstrate that our approach is able to appropriately generalize knowledge about different pouring skills and optimize behavior parameters.

I. INTRODUCTION

An important robotics challenge is to enable robots to help with household chores, such as bringing a beer bottle from a refrigerator to a human, loading dishes into a dishwasher [1], making pancakes [2], and folding towels [3]. We are taking a learning from demonstration (LfD) approach to this problem [4]. However, a key issue for robots to do household chores is how to treat different strategies of each task. Consider an opening task. There are a number of ways to open containers: rotating a cap on a plastic bottle, pulling a hinge cap of a ketchup bottle, pulling a pop-tab of a beer can, tearing a bag of potato chips, and so on. In addition, when opening a tight jar, we will use a different way to open it, like holding a cap with a wet towel. We call these methods *skills*. Learning these skills is essential for robots to fully handle tasks.

In this research, we examine a pouring task to study skill involving many strategies. The purpose of pouring is to move material from a source container to a receiving container. Humans use many skills to pour, such as shaking a bottle to pour viscous liquid like ketchup, tapping a bottle to pour a little amount of coffee powder, squeezing a shampoo bottle, and pushing a soap pump. Each of these versions of the task are hard to model and plan from first principles. Thus, the pouring task is a good example for robots to learn skills.

The goal of this research is making a general pouring model from human demonstrations with which a robot can pour a wide variety of materials using a wide range of containers. This problem is decomposed into four sub-problems: (1) Deriving a model of a skill from human demonstrations to reproduce the skill. Each model will have some parameters to adapt to a specific situation like container size and material. (2) Storing skill models and pairs of a situation description and the parameters for the situation. (3) Selecting a skill for a new situation with estimated parameters. (4) Correcting the selection of a skill if necessary, and adjusting the parameters to the new situation through actual executions.

In this paper, we focus on (4), the learning from practice step, with manually derived skills from human demonstration. Though a number of methods are proposed so far in LfD research, we do not see a practical method that covers (1) to (4). The issue is that we have not formulated this kind of problem yet. Therefore, we start from manually obtained skills and provide a way to improve them through experiments.

Concretely, we model pouring, two shaking, and tapping skills. In addition, we develop a general pouring model that unifies the pouring and the shaking skills. This general pouring model has a wider generalization ability than the individual skills. For learning from practice, we also introduce a parameter optimization architecture for both discrete and continuous parameters of the skills. Finally, we implement the models on a PR2 robot, and conduct experiments on the real robot. In the experiments, we show the generalization ability of these models in terms of the target amount, the source container shapes, and the material kinds.

This paper’s approach is close to the “learning from observations and practice using behavioral primitives” framework proposed by Bentivegna [5]. Another similar approach is found in the “learning parameterized skills” framework [6]. There are several attempts for robots to learn pouring from human demonstrations [7], [8], [9], [10]. However, they are focusing on a part of the entire pouring problem. For example, the method in [8] optimizes the source container trajectory and its goal position for a pouring task, that will have a generalization ability at a certain level, but we cannot expect the generalization to the source container shape and the material kinds. The method in [10] teaches a robot to pour using force information, and models the human demonstration with a parametric hidden Markov model (HMM). Using HMM-like models is useful to encode a human demonstration automatically, but it is just a part of the entire pouring problem. To achieve our goal, we need a method with the capability to learn different strategies for

*A. Yamaguchi was funded in part by Japan Society for the Promotion of Science under the Strategic Young Researcher Overseas Visits Program for Accelerating Brain Circulation G2503. S. Niekum was funded in part by the US National Science Foundation under grant IIS-1208497.

¹A. Yamaguchi, C. G. Atkeson, and S. Niekum are with The Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213-3890, United States ay@akiyam.sakura.ne.jp

²A. Yamaguchi and T. Ogasawara are with Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5, Takayama, Ikoma, Nara 630-0192, Japan

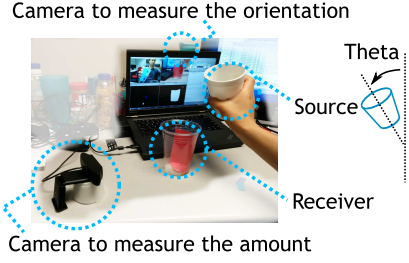


Fig. 1. Setup to measure a human demonstration.

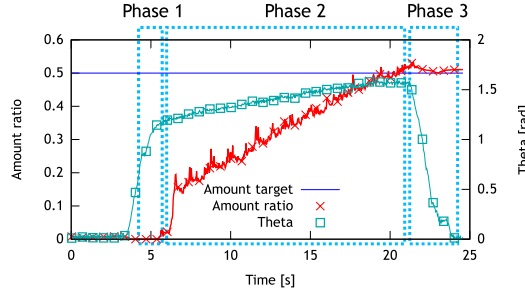


Fig. 2. Result of a human demonstration. The amount trajectory and the orientation (Theta) trajectory are plotted.

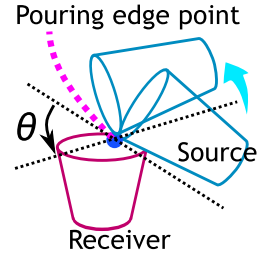


Fig. 3. Illustration of pouring model.

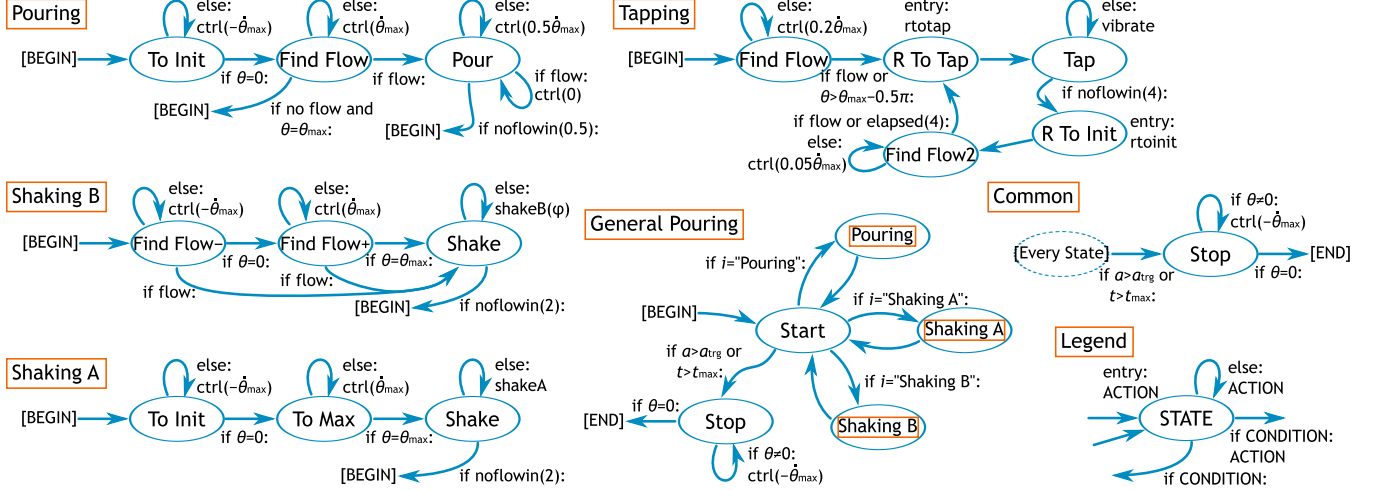


Fig. 4. Finite state machines of pouring, shaking A and B, tapping, and general pouring. See “Legend” to understand the state machines; *else* denotes a condition that is satisfied if the other conditions are false, *entry* denotes an entry action that is performed when entering the state, which is not executed if the next state is the original state. In state machines of pouring, shaking A and B, and tapping, a common condition and the next state are omitted from every state, which is shown as “Common”. Note that in the tapping state machine, the “Stop” state has an entry action: *rtointit*. There are several utility functions; *ctrl*($\dot{\theta}$): control θ with its velocity $\dot{\theta}$, *flow*: true if flow observed, *noflow*: true if no flow observed, *noflowin*(Δt): true if no flow observed in Δt , *shakeA*: perform shaking A motion, *shakeB*(ϕ): perform shaking B motion with the axis parameter ϕ , *rtotap*: move the right gripper to the tapping pose, *rtointit*: move the right gripper to the initial pose, *elapsed*(Δt): true if the elapsed time after entering the state is greater than Δt , *vibrate*: perform vibrating motion. There are several constants and variables; $\dot{\theta}_{max}$: maximum velocity of θ , a_{trg} : target amount, t_{max} : timeout duration, i : selected skill.

each task, improve the strategies from practice, and have an extensible generalization ability. Although our approach relies on manual skill implementation, the obtained method has good generalization ability. This result will provide guidance for robotics, machine learning, and artificial intelligence to develop more practical methods.

In Section II, we model skills from human demonstrations. In Section III, we describe the parameter optimization architecture. In Section IV, we implement skills on a PR2 robot. Section V concludes this paper.

II. LEARNING SKILLS FROM HUMAN DEMONSTRATIONS

A. Human Demonstration of Pouring

First, we observe human demonstrations of pouring. We use the setup shown in Fig. 1 to track the human demonstrations. A human subject will pour from a source container (source) to a receiver container (receiver) where the orientation of the source and the amount of material in the receiver are measured by RGB cameras. The material in

the source is dried *peas* which behave like water, but are more convenient for measuring the amount and for real robot experiments especially when spills occur. The human subject pours the material to a target amount of 0.5 which is half of the receiving container volume.

Fig. 2 shows a demonstration of pouring where the amount in the receiving container and the orientation are plotted. From this demonstration, we can see that there are three phases in pouring. Phase 1: rotating the source container quickly until flow is observed. Phase 2: after flow starts, the human rotates the source container slowly until the amount reaches the target. We found that once the flow starts, it continues without rotating the container much, and we infer the human was more careful. Phase 3: after reaching the target amount, the human moves the source container to the initial pose.

B. Modeling Pouring

Based on the demonstration, we discuss how to model the pouring behavior for a robot. The whole pouring task

consists of grasping a source container, moving it to the receiving container, pouring the material from the source to the receiver, moving the source to an initial location, and releasing the source. For simplicity, here we assume that the robot starts to pour when the robot is grasping a source container and holding it near the receiver.

From the human demonstration, we found that during pouring, the mouth edge of the source was moving little; while, the grasping point was moving more widely. Thus, we think that modeling the movement of a point on the mouth edge is easier than modeling the movement of the gripper. We assume that during pouring, the edge point of the source takes a constant value, and only the orientation changes to control the flow, as illustrated in Fig. 3. The source container moves mostly in a 2-dimensional plane, so the orientation is modeled by a 1-dimensional variable, θ . When the robot is grasping a source container, the position of the edge point of the source is constant in the local frame of the gripper. Thus, controlling the edge point is achieved by a standard inverse kinematics solver.

As we mentioned above, we found three phases in the human demonstration. A simple way to model this kind of behavior is using a finite state machine. When no flow is observed, the robot increases θ (Phase 1). If flow is observed, the robot slows down the movement (Phase 2). If the target amount is achieved, the robot moves θ to the initial value (Phase 3). Through some demonstrations, we found that if the material starts flowing, it continues to flow without increasing θ . Thus, in Phase 2, we increase θ only when the flow is not observed, and keep the same value when the flow is observed.

A state machine for pouring is illustrated in Fig. 4. We assume $\theta = 0$ at the initial pose, and θ should be less than θ_{\max} where the source container is upside down. The “To Init” state is used when θ is not zero at the state. This happens when the robot tries again from the initial pose due to jamming, or the pouring skill is combined with the others. The other states, “Find Flow”, “Pour”, and “Stop”, correspond to the Phase 1 to 3 respectively.

Note that we can expect generalization abilities of this state machine at a certain level, in terms of the target amount, the initial amount in the source, the source container type, and the material kind. In the experiments, we investigate these generalization abilities.

C. Shaking

Humans sometimes shake a bottle to pour when the material is jamming inside the bottle or the material is a viscous liquid. Through human demonstrations, we found that there are some variations in shaking behaviors; for example, shaking vertically, shaking at an angle, shaking linearly, and shaking rotationally. We model two types of shaking since they have good pouring performance. These demonstrations are shown in Fig. 5(a) and 5(b), which we refer to as shaking A and B respectively. Shaking A is a vertical motion while holding the bottle upside down.

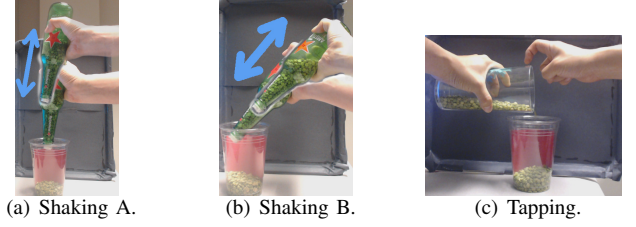


Fig. 5. Human demonstrations of shaking and tapping.

Shaking B is a shaking at an angle where the flow is maximized.

Shaking A and B are modeled with finite state machines as illustrated in Fig. 4. In this figure, shakeA denotes the shaking A motion where the source is moved in vertical direction, and $\text{shakeB}(\phi)$ denotes the shaking B motion where the shaking direction is decided by the parameter ϕ as $[\sin(\phi), 0, -\cos(\phi)]$ in the source frame. The parameter ϕ is chosen to be suitable for the current situation.

D. Tapping

Tapping is used to pour material accurately (see Fig. 5(c)); for example, pouring coffee powder. Reproducing such a motion with a robot is a bit difficult unless the robot can move a gripper or finger rapidly. The PR2 can “tap” by touching the right gripper to the source container held by the left gripper, and then vibrating the right gripper.

The tapping is also modeled with a finite state machine as illustrated in Fig. 4. This state machine is more complicated compared to the others since the tapping includes dual-gripper motions. In preliminary experiments, we found that the initial flow is much larger than the amount poured by the tapping. Thus, we design the state machine so that it can find the initial flow carefully.

E. General Pouring

Finally, we model a general pouring where the pouring, and shaking A and B skills are unified. Each skill is modeled by a finite state machine, so the general pouring is also modeled by a state machine, as illustrated in Fig. 4. This is a higher-level state machine, where the individual state machines are used as sub-state machines. Basically, [BEGIN] and [END] of each sub-state machine are connected to the “Start” state of the general pouring state machine. The “Stop” state of each sub-state machine is removed and the terminal condition is directly connected to [END]. This is because every sub-state machine is designed to be able to start at any value of θ . In a known situation, namely, a known source container type and material kind, the suitable skill can be selected easily; in an unknown situation, the skill selection is optimized through trial and error as described in the next section.

III. LEARNING FROM PRACTICE

The task of the human demonstration shown in Fig. 5(a) and 5(b) was a bit difficult for the human, where we found several interesting behaviors. First, the human tried some

different skills that we are referring to as shaking A and B. Since the human did not know the most suitable skill for the task, the human investigated the best one through trial and error. The human also adjusted some skill-related parameters like shaking axis and speed.

These are on-line parameter optimizations. The former one is a discrete parameter optimization (skill selection), and the latter one is a continuous parameter optimization. We introduce a parameter optimization architecture into the skill models so that the robot can widen the variety of pouring task scenario. In the following, we describe the problem setup, optimization methods, and the architecture to introduce the optimization methods into the skill models.

A. Problem Specification

Though there are several objectives in the pouring task such as pouring as fast as possible and avoiding spills, we focus on the pouring speed optimization. The optimized parameters are, for example, skill selection, the shaking axis, and the shaking speed. The problem is to maximize the pouring speed, namely a score function, with respect to these parameters. We do not have an analytical model between the parameters and the score; the score can be obtained through actual execution using the parameters.

There are two kinds of parameters to be optimized. One is a discrete parameter that is selected from a set of options (e.g. a set of skills). The other is a continuous parameter that is selected from a single or multi dimensional continuous space (e.g. a shaking axis).

B. Optimization Method

Since score functions like the pouring speed are noisy, we need to choose a robust optimization method. For continuous parameter optimization, we use the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) proposed and implemented by Hansen [11]. CMA-ES is an evolutionary algorithm that does not require the gradient of the score function.

For the discrete parameter optimization problem, we use Boltzmann selection (as known as the softmax selection) [12] to select an option where each option is evaluated with an upper confidence bound (UCB). Though there are several versions of the UCB, we use the sum of the expected score μ and its standard deviation σ similarly to [13]. μ and σ are updated by an exponential moving average scheme. The details can be found in [14].

There are many other choices to optimize the parameters, for example, using a reinforcement learning method (e.g. [15]). Rather than using or developing a state-of-the-art method, we are focusing on showing an entire solution to the pouring problem. Thus, we choose simple but practical methods.

C. Architecture

We use these optimization methods in an on-line manner. An on-line parameter optimization consists of three steps: (1) Selecting a parameter to be used, (2) Using the parameter

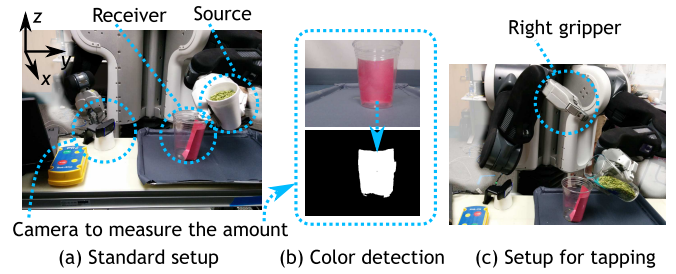


Fig. 6. Setup of the experiments.

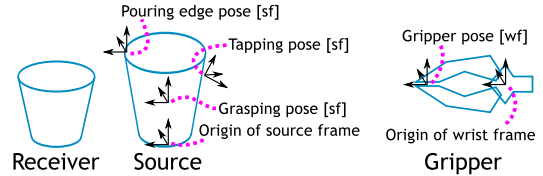


Fig. 7. Object specific vectors. A pose means a xyz position and a quaternion. [sf] and [wf] denote a vector defined in the source container and the wrist frames respectively.

and obtaining the score, and (3) Updating based on the score. A natural way to integrate these steps in a state machine is treating them as actions of the state machine, since these are a kind of action.

The optimization method for the skill selection is introduced to decide the skill index i in the general pouring state machine (Fig. 4). In this case, the selecting and the updating steps are executed as the entry action of the “Start” state. The score is the difference of the amount divided by the execution duration.

Note that the general pouring state machine supports trial and error learning during a single pouring trial. For example, if the material jams in the shaking A state machine, it is detected by the `noflowin(2)` condition, and the state moves back to the “Start” of the general pouring. Thus, the skill selection can be updated and a new skill selected.

In this paper, we apply continuous parameter optimization to ϕ of the shaking B. In this case, the selecting and the updating steps are executed right before and right after the `shakeB(ϕ)` action.

IV. EXPERIMENTS

We implement the pouring skills modeled in the previous sections on a robot, and investigate their capabilities. In the experiments, first, we show the generalization ability of the pouring in terms of the target amount. Second, we compare the shaking A and B skills. Third, we show how learning from practice works. Fourth, we demonstrate the generalization ability of the general pouring model in terms of the source container shapes and the material. Finally, we investigate the tapping skill. The accompanying video is available at <http://youtu.be/cGci9F0l680>.

We use a PR2 robot that has two 7-degrees of freedom arms with grippers. Fig. 6(a) shows the setup of the robot and containers. In order to measure the amount of poured



Fig. 8. Source containers and poured materials. Each photo contains a coin of diameter 0.955 inches as a scale. BBs are copper-coated bullets for toy guns. The container B3 and B4 are specially designed for the experiments; these holes are designed to be small in order to produce jamming.

material, we use an RGB camera and detect specific colors as shown in Fig. 6(b). The ratio of colored areas is used as the amount. For this purpose, as the receiving container, we use a transparent plastic container whose back half is colored. We use ROS packages for the PR2 to implement low-level control and inverse kinematics solver of the grippers.

As mentioned in Section II-B, in order to control the pose of the pouring edge point with the 1-dimensional variable θ , we define several object specific vectors as illustrated in Fig. 7. The pouring edge pose and the grasping pose are constant vectors in the source container frame; these are defined for each source container. Since during grasping, the gripper pose corresponds to the grasping pose, we can compute the pouring edge pose in the wrist frame. An inverse kinematics solver for the wrist link is implemented in a ROS package, thus we can control the pouring edge pose. From the initial pose of the source container, we can estimate the rotation axis.

In order to investigate the generalization ability in terms of the source containers and the poured materials, we prepare 14 containers and 5 materials as shown in Fig. 8. Though we use only dry materials to avoid hardware damage by liquid, they behave similarly to a viscous or runny liquid. The object specific vectors shown in Fig. 7 are manually measured for each container. We plan to totally automate perception in future work.

A. Pouring

Fig. 9 shows a typical result of pouring where the target amount is 0.5, the source container is B1, and the poured material is the dried peas. The flow started around 2 [s], and the robot slows down. Compared to the human demonstration (Fig. 2), the robot behavior has similar structure; namely, there are three phases. However, we can find some differences. For example, the human slows down the angular velocity before the flow starts. One possible reason is that the human tries to keep the initial flow small. Estimating the orientation where the flow starts is necessary to reproduce this

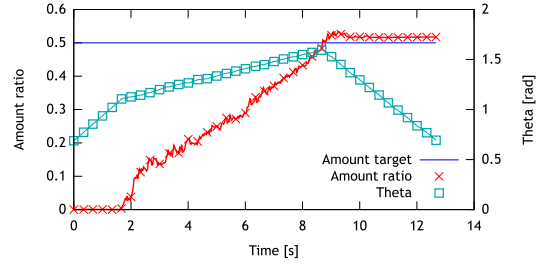
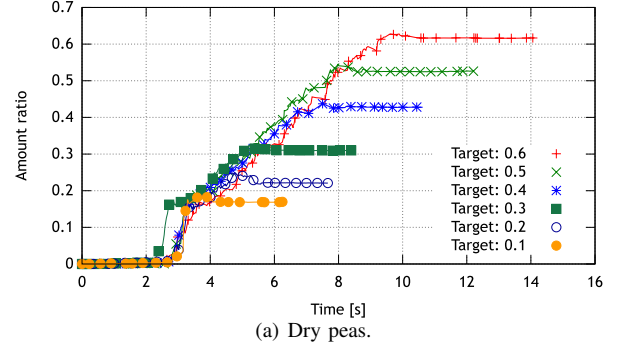
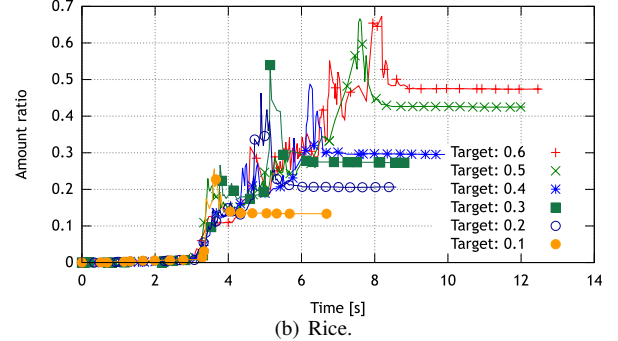


Fig. 9. Typical result of pouring. The setup corresponds to that of the human demonstration (Fig. 2); the source container is B1, and the poured material is the dried peas. For the consistency with Fig. 2, θ (Theta) is shifted so that their ranges match with each other.



(a) Dry peas.



(b) Rice.

Fig. 10. Generalization in terms of target amount.

behavior; humans are using visual information and/or force information. We have not yet implemented this behavior.

Next, we investigate the generalization ability of the pouring skill in terms of the target amount. We use B1 as the source container, the dry peas and the rice as the poured material, and change the target amount from 0.1 to 0.6. Fig. 10(a) and 10(b) show the results of using the peas and the rice respectively. In Fig. 10(a), the case of the largest error is at the target of 0.1. The reason is that the amount of initial flow (around 3 [s]) was large; the initial flow poured more than the target amount. In the other cases, the target amount is achieved well.

On the other hand, the results of the rice case (Fig. 10(b)) seem to be noisy. Each amount trajectory overshoots. This was caused by the vision system being confused by the stream of material during the pouring. Compared to the peas, the flow of rice spread more widely and was more visible to the camera. There are several ways to reduce this problem:

TABLE I
COMPARISON OF SHAKING A AND SHAKING B.

Src.	Method	# of failures	Avg. duration [s]	SD [s]
B4	Shaking A	0	19.06	1.27
	Shaking B(0)	2	45.76	N/A
	Shaking B($\pi/4$)	0	31.03	6.79
B3	Shaking A	2	94.29	N/A
	Shaking B(0)	3	N/A	N/A
	Shaking B($\pi/4$)	0	38.87	0.54

using a better vision system, adjusting pouring parameters, using force information like [10], and so on. Anyway, we can say that the pouring skill has some generalization ability in terms of the target amount.

B. Comparison of Shaking Skills

We investigate two versions of the shaking skills, shaking A and B, and make clear the necessity of both versions. We use B4 and B3 as the source containers. The poured material is the dried peas. For each container, we apply the shaking A, shaking B with the axis $\phi = 0$ and $\phi = \pi/4$.

Table I shows the results of the B4 and the B3 cases. The table describes the number of failures out of 3 runs, average pouring duration and its standard deviation. The failure is a timeout case; t_{\max} is about 40 [s] in the B4 case, and is about 80 [s] in the B3 case, since the latter case is more difficult. Obviously, in the B4 case, shaking A outperforms the others; meanwhile in the B3 case, shaking B with $\phi = \pi/4$ is the best. Both shaking versions are necessary to cover the wide range of source containers.

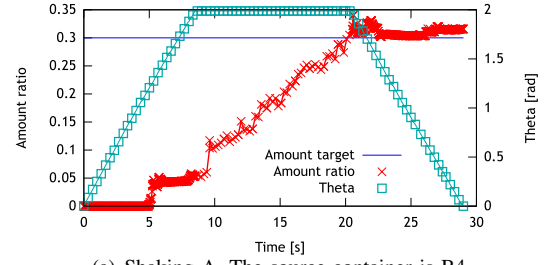
Fig. 11(a) shows a result of shaking A for the B4 container. Fig. 11(b) shows a result of shaking B with $\phi = \pi/4$ for the B3 container. While the orientation θ takes a constant value, the shaking motion is performed. In Fig. 11(a), we can see a little flow before starting shaking (around 5 [s]) but the flow stops due to the jammed material, so standard pouring does not work any more. During shaking, we can see the amount is increasing. Thus, shaking is a possible way for the robot to solve jamming. Compared to the B4 case, it takes more time to pour the target amount in the B3 case. The B3 container is also difficult for humans to pour.

C. Learning from Practice

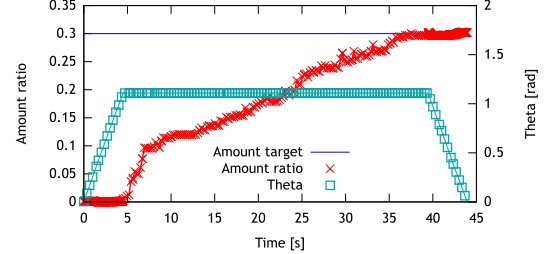
We demonstrate how the parameter optimization architecture works. Here, we investigate separately discrete parameter optimization (skill selection) and continuous parameter optimization.

1) *Skill Selection Optimization*: There are three choices: pouring, shaking A, and shaking B. The axis of shaking B is fixed to $\phi = \pi/4$. We initialize the expected scores of the three options as 1.0, 0.5, 0.5 respectively, which means that the robot will use standard pouring initially.

Fig. 12(a) shows the result of the B3 case, and Fig. 12(b) shows the result of the B4 case. The dried peas are used in both cases. Several trials are sequentially done in each case; six trials in the B3 case, and seven trials in the B4 case. In each graph, the selected option is plotted on the amount trajectory.

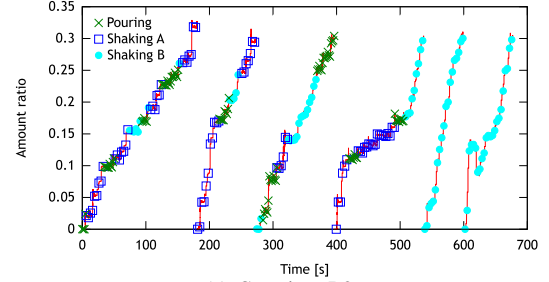


(a) Shaking A. The source container is B4.

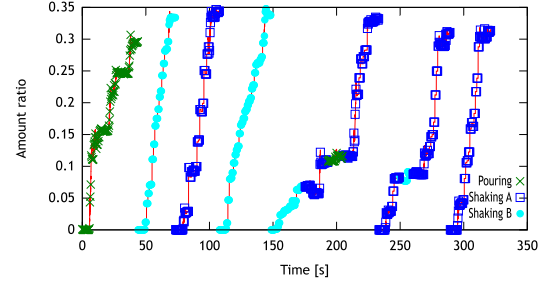


(b) Shaking B($\pi/4$). The source container is B3.

Fig. 11. Results of shaking A and B. The material is the dried peas.



(a) Container B3.

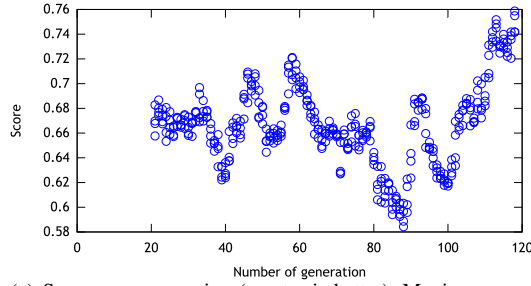


(b) Container B4.

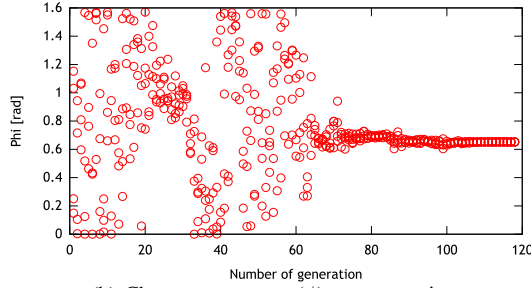
Fig. 12. Learning process of skill selection optimization.

In the first trial of Fig. 12(a), we can see the three options were tried. First, the robot applied pouring, but since it did not work, the robot switched to shaking A (recall that this is an on-line parameter optimization). In the first and second trials, shaking A seems to have been dominant. However, shaking A got stuck due to jamming in the 4-th trial. Eventually, shaking B was selected.

In the first trial of Fig. 12(b), only pouring was used though the selection was done several times. As we could see a little flow before starting shaking in Fig. 11(a), the standard pouring works in this setup initially. Actually in the first trial, repeating alternately pouring and going back to the initial achieved the target amount. Thus, it took several times to learn that the performance of pouring was not good. In the



(a) Scores per generation (greater is better). Moving average filter is applied, and the data of first 20 generations are omitted because of inadequate data for the filter.



(b) Chosen parameters (ϕ) per generation.

Fig. 13. Learning process of parameter ϕ optimization in shaking B.

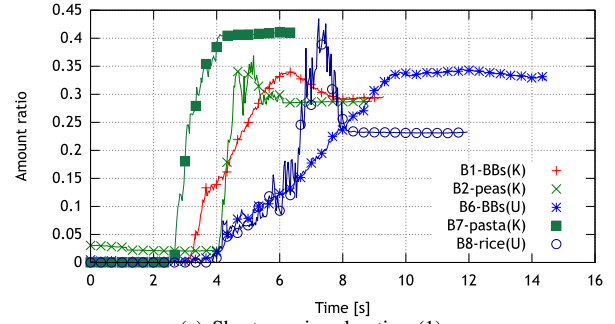
second and the third trials, the robot used the shaking B and A respectively. Since the robot could pour continuously with each skill, it did not change the skill in each trial. In 5-th and 6-th trials, the robot experienced jamming with the shaking B. Eventually, the robot decided to use the shaking A for this situation.

Therefore, in the both B3 and B4 cases, we obtain the corresponding results with the previous experiment.

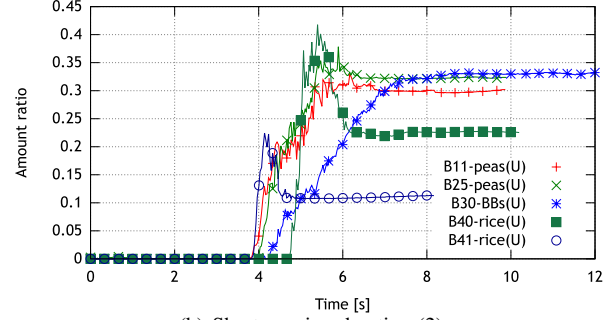
2) *Continuous Parameter Optimization:* Next, we optimize the parameter ϕ to decide the axis of the shaking B. The initial mean and standard deviation are $\phi = \pi/4$ and 1 respectively. In the early stage of optimization, the robot will choose the parameter almost randomly with this configuration. ϕ is limited in $[0, \phi/2]$.

We ran 11 trials sequentially. Fig. 13(a) shows the scores in each generation. Fig. 13(b) shows the parameters in each generation of the CMA-ES; there are 4 individuals (different parameters) in each generation. We can see that the parameter converges to around 0.6 in Fig. 13(b), and the score is improved in Fig. 13(a). The pouring duration was improved from 60.71 [s] of the first trial to 42.47 [s] of the last trial. In the previous experiment, we compared $\phi = 0$ and $\phi = \pi/4$ in the same setup, and found that $\phi = \pi/4$ is better. In this experiment, the robot could find the optimal parameter automatically, and the found parameter is close to the manual optimization result, $\phi = \pi/4$. Thus, the CMA-ES could find an appropriate solution.

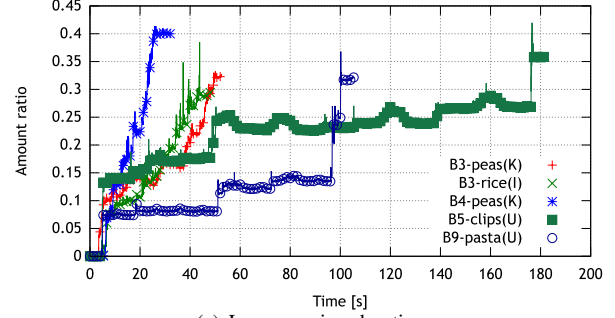
On the other hand, the score function was very noisy. Even taking the same parameter, the resulted scores are different. The shaking result is affected by the previous shaking motion. Due to these effects, in Fig. 13(b), the parameter seems to have almost converged around 30 [s], but the score of the corresponding generation was not so



(a) Short pouring duration (1).



(b) Short pouring duration (2).



(c) Long pouring duration.

Fig. 14. Results of generalization test in terms of the source container and the poured material. Each curve shows a result of SOURCE-MATERIAL combination. (K) shows the parameters are known, (U) shows unknown, and (I) shows incorrect.

high. After that, it increased the search deviation and found a parameter with a better score. Thus, CMA-ES is practically useful.

D. Generalization Ability

We investigate the generalization ability in terms of the source container and the poured material. Each situation is described as a SOURCE-MATERIAL format; e.g. B1-BBS. We prepare 15 combinations from the containers and the materials shown in Fig. 8. For some of them, we manually assigned the parameters of the skill selection and the shaking axis. We assigned incorrect parameters for a B3-rice case where we use the same parameters as those of B3-peas treated in the previous experiments. We assume the parameters of the other cases are unknown. For these cases, we initialize the parameters as was done in the parameter optimization experiments. The target amount is 0.3 except for a B41-rice case; in the B41-rice case, the target amount is 0.1 since B41 is a small container.

Fig. 14 shows the results where the combinations are categorized into long pouring duration ones (Fig. 14(c)) and short ones (the others). In the B8-rice, the B40-rice, and the B41-rice cases, the overshoot problem happened similarly to the previous experiment. In the B7-pasta case, the poured amount exceeds the target significantly. This is because the friction between the source container and the material is pretty low in this combination. In the B6-BBs case, the poured amount also exceeds the target. The reason is that since the hole of the container B6 is small, the robot rotated the container more than the others, which increased the material flow during moving the container back to the initial orientation. For the other cases in the short pouring duration category, the target amount was almost achieved.

In the long pouring duration cases in Fig. 14(c), the known cases, B3-peas and B4-peas, were poured relatively quickly. Though incorrect parameters were given, the B3-rice case was also poured fast. This is because the problem setup was similar; the difference was the dried peas and the rice only. The B5-clips and the B9-pasta cases took longer time. Obviously, these problems were difficult since the material particle was large compared to the containers' holes. Since these were unknown setups, the skill selection was optimized during the execution (we executed only one trial).

Though there is room for improvement, we have achieved the generalization ability at a certain level.

E. Tapping

Next, we investigate the performance of tapping. This skill uses the right gripper to tap, so we start from the setup shown in Fig. 6(c). The robot grasps a source container with the left gripper while the right gripper stays above the receiving container. In order to touch the right gripper to the source container, we define a tapping pose as illustrated in Fig. 7, which is a constant vector in the source frame. We use the B25 container and the BBs.

Fig. 15 shows a part of trajectories of the tapping where the result of B25-peas in the previous experiment is shown as the comparison. We can see that using tapping, the robot can pour the material very slowly. Thus, tapping enables the robot to pour accurately. However, there is another difficulty; in the slow pouring setup, the amount of initial flow is comparably large, which dominates the total amount. Thus, without an accurate controller for the initial flow, we cannot achieve accurate pouring with respect to the total amount.

V. CONCLUSION

In this paper, we investigated a way for robots to learn pouring from demonstration and practice. A pouring task involves the possible use of many skills; we manually modeled pouring by tipping, shaking, and tapping using finite state machines. We unified the pouring and the two shaking skills as a general pouring model. For learning a skill selection and skill parameters from practice, we introduced a discrete and continuous parameter optimization architecture into the models. The constructed models were verified by implementing them on a PR2 robot. The robot experiments

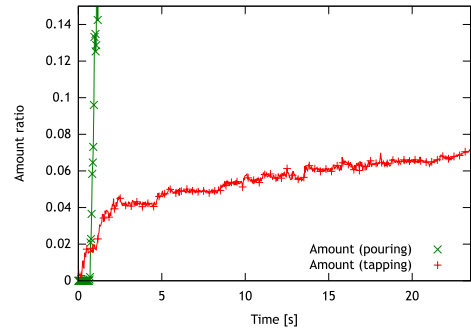


Fig. 15. Comparison of the standard pouring and tapping.

demonstrated that our approach could appropriately generalize knowledge about different pouring skills and optimize skill parameters. Future work is expanding the scenario to the whole pouring process including a pick-and-place of the container, and using the pouring skills in higher level tasks like cooking, as well as increasing the number of skills.

REFERENCES

- [1] T. Asfour, P. Azad, N. Vahrenkamp, K. Regenstein, A. Bierbaum, K. Welke, J. Schröder, and R. Dillmann, "Toward humanoid manipulation in human-centred environments," *Robotics and Autonomous Systems*, vol. 56, no. 1, pp. 54–65, 2008.
- [2] P. Kormushev, S. Calinon, and D. G. Caldwell, "Robot motor skill coordination with EM-based reinforcement learning," in the *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3232–3237.
- [3] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," in the *IEEE International Conference on Robotics and Automation*, 2010, pp. 2308–2315.
- [4] A. Billard and D. Grollman, "Robot learning by demonstration," *Scholarpedia*, vol. 8, no. 12, p. 3824, 2013.
- [5] D. C. Bentivegna, "Learning from observation using primitives," Ph.D. dissertation, Georgia Institute of Technology, 2004.
- [6] B. da Silva, G. Konidaris, and A. Barto, "Learning parameterized skills," in the *29th International Conference on Machine Learning*, 2012.
- [7] M. Mühlh, M. Gienger, S. Hellbach, J. J. Steil, and C. Goerick, "Task-level imitation learning using variance-based movement optimization," in the *IEEE International Conference on Robotics and Automation*, 2009, pp. 1177–1184.
- [8] M. Tamosiunaite, B. Nemec, A. Ude, and F. Wörgötter, "Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives," *Robotics and Autonomous Systems*, vol. 59, no. 11, pp. 910–922, 2011.
- [9] K. Kronander and A. Billard, "Online learning of varying stiffness through physical human-robot interaction," in the *IEEE International Conference on Robotics and Automation*, 2012, pp. 1842–1849.
- [10] L. Roza, P. Jiménez, and C. Torras, "Force-based robot learning of pouring skills using parametric hidden markov models," in the *IEEE-RAS International Workshop on Robot Motion and Control*, 2013.
- [11] N. Hansen, "The CMA evolution strategy: a comparing review," in *Towards a new evolutionary computation*, J. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Eds. Springer, 2006, pp. 75–102.
- [12] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
- [13] O. Kroemer, R. Detry, J. Piater, and J. Peters, "Combining active learning and reactive control for robot grasping," *Robotics and Autonomous Systems*, vol. 58, no. 9, pp. 1105–1116, 2010.
- [14] A. Yamaguchi, J. Takamatsu, and T. Ogasawara, "Learning strategy fusion to acquire dynamic motion," in the *11th IEEE-RAS International Conference on Humanoid Robots*, Bled, Slovenia, 2011, pp. 247–254.
- [15] J. Kober, A. Wilhelm, E. Oztog, and J. Peters, "Reinforcement learning to adjust parametrized motor primitives to new situations," *Autonomous Robots*, vol. 33, pp. 361–379, 2012.