Robot Learning: Some Recent Examples

G. Konidaris, S. Kuindersma, S. Niekum, R. Grupen, and A. Barto

Abstract— **This paper provides a brief overview of three recent contributions to robot learning developed by researchers at the University of Massachusetts Amherst. The first is the use of policy search algorithms that exploit new techniques in nonparameteric heteroscedastic regression to directly model policy-dependent distribution of cost [12]. Experiments demonstrate dynamic stabilization of a mobile manipulator through learning flexible, risk-sensitive policies in very few trials. The second contribution is a novel method for robot learning from unstructured demonstrations [19] that permits intelligent sequencing of primitives to create novel, adaptive behavior. This is demonstrated on a furniture assembly task using the PR2 mobile manipulator. The third contribution is a robot system that autonomously acquires skills through interaction with its environment [6]. Material in this paper has been published previously in refs. [8, 10, 11, 13, 14, 15, 19, 20] from which additional details are available.**

I. INTRODUCTION

Recent contributions to robot learning developed by researchers at the University of Massachusetts Amherst illustrate new methods for learning and exploiting behavioral modularity. The first is the use of policy search algorithms that exploit new techniques in nonparameteric heteroscedastic regression to directly model policy-dependent distribution of cost [12]. The learned cost model is used as a critic for performing risk-sensitive gradient descent. Experiments are presented in dynamic stabilization and manipulation with a mobile manipulator that demonstrate learning of flexible, risk-sensitive policies in very few trials. The second contribution is a novel method for robot learning from unstructured demonstrations [19]. This method uses a Beta Process Autoregressive Hidden Markov Model to automatically segment demonstrations into motion categories, which are then further subdivided into semantically grounded states of a finite-state automaton to permit intelligent sequencing of primitives to create novel, adaptive behavior. This is demonstrated on a furniture assembly task using the PR2 mobile manipulator. The third contribution is a robot system that autonomously acquires skills through interaction with its environment [6]. The robot learns to sequence the execution of a set of innate controllers to solve a task, extracts and retains components of that solution as portable skills, and then transfers those skills to reduce the time required to learn to solve a second task.

II. BAYESIAN OPTIMIZATION FOR VARIABLE RISK CONTROL

Experiments on physical robot systems are typically associated with significant practical costs, such as experimenter time, money, and robot wear and tear. However, such experiments are often necessary to refine controllers that have been hand designed or optimized in simulation. For many nonlinear systems, it can even be infeasible to perform simulations or construct a reasonable model. Consequently, model-free policy search methods have become one of the standard tools for constructing controllers for robot systems. These algorithms are designed to minimize the expected value of a noisy cost signal by adjusting policy parameters. By considering only the expected cost of a policy and ignoring cost variance, the solutions found by these algorithms are by definition *risk-neutral*. However, for systems that operate in a variety of contexts, it can be advantageous to have a more flexible attitude toward risk. Bayesian optimization is a promising approach to this problem.

A. Variational Bayesian Optimization

Bayesian optimization algorithms are a family of global optimization techniques that are well suited to problems where noisy samples of an objective function are expensive to obtain [2]. Recently there has been increased interest in applying Bayesian optimization algorithms to solve model-free policy search problems [13, 18, 22]. In contrast to well-studied policy gradient methods [21], Bayesian optimization algorithms perform policy search by modeling the distribution of cost in policy parameter space and applying a selection criterion to *globally* select the next policy. Selection criteria are typically designed to balance exploration and exploitation with the intention of minimizing the total number of policy evaluations.

Previous implementations of Bayesian optimization for policy search have assumed that the variance of the cost is the same for all policies in the search space, which is not true in general. Kuindersma [12] (see also ref. [15]) introduced a new Bayesian optimization algorithm, the Variational Bayesian Optimization (VBO) algorithm, that relaxes this assumption and efficiently captures both the expected cost and cost variance during the optimization. Specifically, Kuindersma extended a variational Gaussian Process regression method for problems with input-dependent noise (or *heteroscedasticity* [17]) to the optimization case by deriving an expression for expected risk improvement, a generalization of the commonly used expected improvement (EI) criterion for selecting the next policy, and incorporating log priors into the optimization to improve numerical performance. This

G. Konidaris and S. Kuindersma are with the MIT Computer Science and Artificial Intelligence Laboratory: gdk@csail.mit.edu, scottk@csail.mit.edu

S. Niekum, R. Grupen, and A. Barto are with the School of Computer Science, University of Massachusetts Amherst: sniekum@cs.umass.edu, grupen@cs.umass.edu, barto@cs.umass.edu.

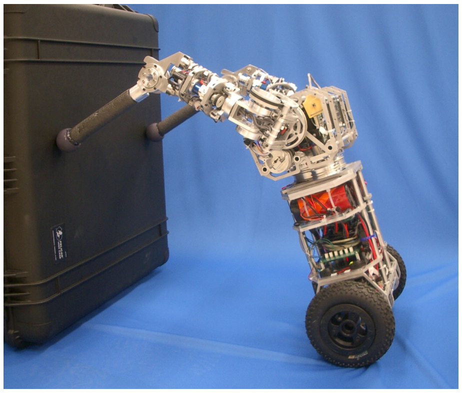


Fig. 1. The uBot-5 demonstrating a whole-body pushing behavior.

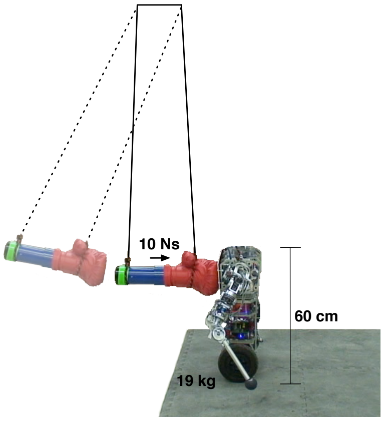


Fig. 2. The uBot-5 situated in the impact pendulum apparatus.

selection criterion includes a parameter, κ, that controls the system's risk sensitivity by weighting the standard deviation of the cost function in the risk-sensitive objective, with larger values of κ indicating more sensitivity to risk. Confidence bounds were also considered to produce *runtime* changes to risk sensitivity, yielding a generalized expected risk improvement criterion that balances exploration and exploitation in risk-sensitive setting.

B. Balance Recovery with the uBot-5

The uBot-5 (Fig. 1) is an 11-DoF mobile manipulator developed at the University of Massachusetts Amherst [3, 16]. The uBot-5 has two 4-DoF arms, a rotating trunk, and two wheels in a differential drive configuration. The robot stands approximately 60 cm from the ground and has a total mass of 19 kg. The robot's torso is roughly similar to an adult human in terms of geometry and scale, but instead of legs, it has two wheels attached at the hip. The robot balances using a linear-quadratic regulator (LQR) with feedback from an onboard inertial measurement unit to stabilize around the vertical fixed point. The LQR controller has proved to be very robust throughout five years of frequent usage and it remains fixed in the experiments described here.

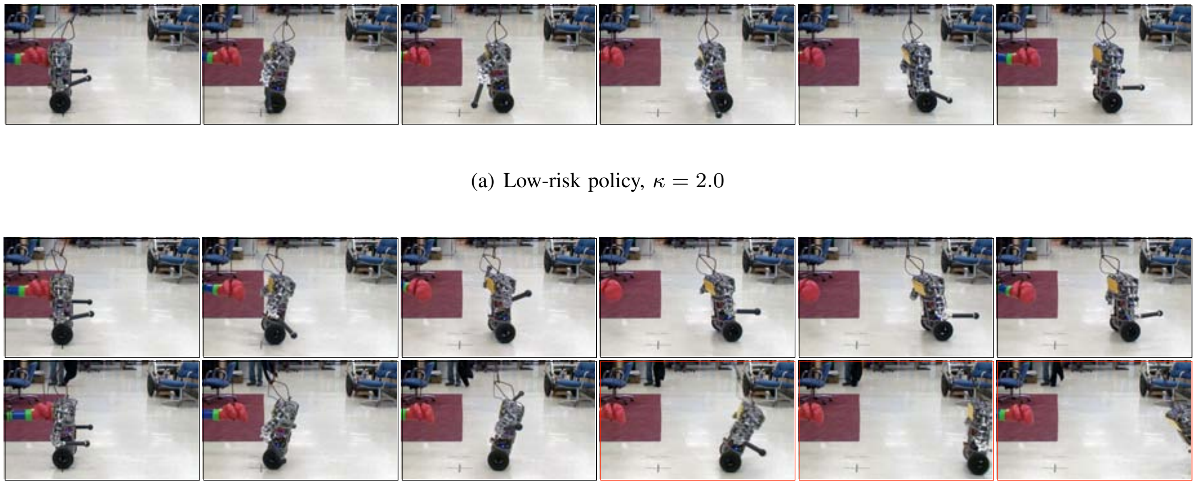
In previous experiments [13], the energetic and stabilizing effects of rapid arm motions on the LQR stabilized system were evaluated in the context of recovery from impact perturbations. One observation made was that high energy impacts caused a subset of possible recovery policies to have high cost variance: successfully stabilizing in some trials, while failing to stabilize in others. Kuindersma [12] extended these experiments by considering larger impact perturbations, increasing the set of arm initial conditions, and defining a policy space that permits more flexible, asymmetric arm motions.

The robot was placed in a balancing configuration with its upper torso aligned with a 3.3 kg mass suspended from the ceiling (Fig. 2). The mass was pulled away from the robot to a fixed angle and released, producing a controlled impact between the swinging mass and the robot, resulting in an impact force approximately equal to the robot's total mass. The robot was consistently unable to recover from this perturbation using only the wheel LQR (see the rightmost column of Fig. 3). (The robot was attached to the ceiling with a loose-fitting safety rig designed to prevent it from falling completely to the ground, while not affecting policy performance.)

This problem is well suited for model-free policy optimization since there are several physical properties, such as joint friction, wheel backlash, and tire slippage, that make the system difficult to model accurately. In addition, although the underlying state and action spaces are high dimensional (22 and 8, respectively), low-dimensional policy spaces that contain high-quality solutions are relatively straightforward to identify. In particular, the policy controlled each arm joint according to a parameterized exponential trajectory. The pitch (dorsal) motions were specified separately for each arm and the lateral motions were mirrored, which reduced the number of policy parameters to 3. After each trial, the arms were retracted to a nominal configuration using a fixed, low-gain linear position controller. The cost function was designed to encourage energy efficient solutions that successfully stabilized the system.

After 15 random initial trials, VBO was applied with EI selection for 15 episodes and randomized confidence bound (CB) selection for 15 episodes resulting in a total of 45 policy evaluations (approximately 2.5 minutes of total experience). After training, four policies were evaluated with different risk sensitivities selected by minimizing the CB criterion with $\kappa = 2$, $\kappa = 0$, $\kappa = -1.5$, and $\kappa = -2$. Each selected policy was evaluated 10 times, and the results are shown in Fig. 3. The sample statistics confirm the algorithmic predictions about the relative riskiness of each policy. In this case, the risk-averse and risk-neutral policies were very similar (no statistically significant difference between the mean or variance), while the two risk-seeking policies had higher variance (for $\kappa = -2$, the differences in both the sample mean and variance were statistically significant).

For $\kappa = -2$, the selected policy produced an upward laterally-directed arm motion that failed approximately 50% of the time. A slightly less risk-seeking selection ($\kappa = -1.5$) yielded a policy with conservative low-energy arm movements that was more sensitive to initial conditions than


(a) Low-risk policy, $\kappa = 2.0$

(b) High-risk policy, $\kappa = -2.0$

Fig. 4. Time series (time between frames is 0.24 seconds) showing (a) a trial executing the low-risk policy and (b) two trials executing the high-risk policy. Both policies were selected using confidence bound criteria on the learned cost distribution. The low-risk policy produced an asymmetric dorsally-directed arm motion with reliable recovery performance. The high-risk policy produced an upward laterally-directed arm motion that failed approximately 50% of the time.

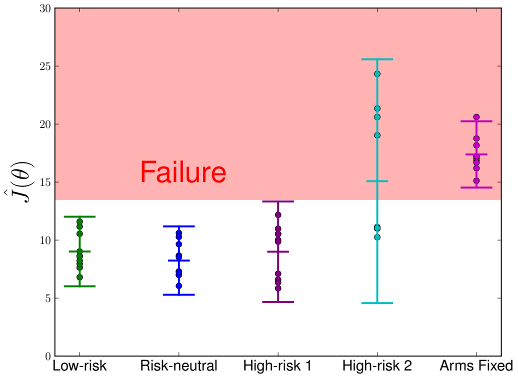


Fig. 3. Data collected over 10 trials using policies identified as risk-averse, risk-neutral, and risk-seeking after performing VBO. The policies were selected using confidence bound criteria with $\kappa = 2$, $\kappa = 0$, $\kappa = -1.5$, and $\kappa = -2$, from left to right. The sample means and two times sample standard deviations are shown. The shaded region contains all trials that resulted in failure to stabilize. Ten trials with a fixed-arm policy are plotted on the far right to serve as a baseline.

the lower risk policies. This exertion of minimal effort can be viewed as a kind of gamble on initial conditions. Fig. 4 shows example runs of the risk-averse and risk-seeking policies.

Varying risk sensitivity based on runtime context is a potentially powerful way to generate flexible control in robot systems. Kuindersma [12] considered this problem in the context of model-free policy search, where risk-sensitive parameterized policies can be selected based on a learned cost distribution. The experimental results suggest that VBO is an efficient and plausible method for achieving variable risk control.

III. LEARNING FROM UNSTRUCTURED DEMONSTRATIONS

Robot learning from demonstration (LfD) [1] has become a popular way to program robots. LfD allows users to teach a robot by example, often eliminating the need for specialized knowledge of the robotic system and taking much less time than it would take an expert to design a controller by hand. While much LfD research has focused on tasks that can be represented by monolithic policies, some recent work has focused on automatically segmenting demonstrations into simpler primitives that can be sequenced to perform complex, multi-step tasks [5, 11, 20]. Such segmentations can be performed by humans, but this may require specialized knowledge, such as the robot's internal representations and kinematic properties. Furthermore, manually managing, memorizing, and reusing a library of primitives becomes intractable for a human user as the library grows in size. Thus, it is advantageous for primitives to be automatically segmented and managed.

Niekum et al. [19, 20] developed a novel method to sequence automatically discovered primitives that makes minimal assumptions about the structure of the task and can sequence primitives in previously unseen ways to create new, adaptive behaviors. Specifically, a Beta Process Autoregressive Hidden Markov Model (BP-AR-HMM) [4] is used to segment continuous demonstration data into motion categories with associated coordinate frames. Tests are then performed on the motion categories to further subdivide them into semantically grounded movement primitives that are used to create a finite-state representation of the task. In this representation, each state has an associated set of exemplars of the relevant movement primitive, plus a trained classifier used to determine state transitions. The resulting finite-state automaton (FSA) can then be used to replay a complex, multi-step task. Further, the method allows the user to provide new demonstrations that can fill in the gaps in the robot's knowledge through interactive corrections at the time of failure. Together, this allows for iterative, incremental learning and improvement of a complex task from unsegmented demonstrations. Niekum et al. [19] illustrated the utility of this system on a complex furniture assembly task using a PR2 mobile manipulator.

A. Demonstration, Segmentation, and FSA Construction

Task examples are provided to the robot via kinesthetic demonstrations, in which the teacher physically moves the robot to perform the task. After a set of demonstrations has been collected in various configurations, the robot pose information is segmented and labeled by the BP-AR-HMM. The segmentation process provides a set of segment lists and corresponding label vectors. Each integer label corresponds to a unique motion category discovered by the BP-AR-HMM segmentation. The clustering method described in Niekum et al. [20] is used to automatically discover coordinate frame assignment lists.

An FSA that represents the task can begin to be constructed by creating nodes that correspond to the labels. Each node is assigned the set of all exemplars that have the same label, and the labels of the previous and next segments are also recorded. A transition matrix is then constructed, where each entry is set to 1 if there exists a corresponding directed transition and 0 otherwise. When the structure of the FSA is finalized, a classifier is trained for each node that has multiple descendants. This is used as a transition classifier to determine which node to transition to next, once execution of a primitive at that node has taken place. Given a novel situation, the FSA can be used to replay the task. The current observation is classified to determine which node to transition to next.

At any time during execution, the user can push a button on the joystick to stop the robot so that an interactive correction can be made. The robot immediately stops execution of the current movement and switches modes to accept a kinesthetic demonstration from the user. From the beginning of execution, the robot has been recording pose data in case of an interactive correction, and it continues to record as the user provides a demonstration of the remainder of the task. After any number of replays and interactive corrections have taken place, the corrections are integrated with the existing data for improved performance.

B. Experiment: demonstrations, corrections, and replay

Niekum et al. [19] evaluated the system on a furniture assembly task, using a PR2 mobile manipulator to partially assemble a small off-the-shelf table. The table consists of a tabletop with four pre-drilled holes and four legs that each have a screw protruding from one end. Eight kinesthetic demonstrations of the assembly task were provided, in which the tabletop and one leg were placed in front of the robot in various positions. In each demonstration, the robot was made to pick up the leg, insert the screw-end into the hole in the tabletop, switch arms to grasp the top of the leg, hold the tabletop in place, and screw in the leg until it is tight. An example of this progression is shown in Fig. 5.

The demonstrations were then segmented and and an FSA was built. At this stage, task replay was sometimes successful, but several types of errors occurred intermittently. Two particular types of errors that occurred were (a) when the table leg was at certain angles, the robot was prone to missing the grasp, and (b) when the leg was too far from the robot, it could not reach far enough to grasp the leg at the desired point near the center of mass. In both cases interactive corrections were provided to recover from these contingencies. In the first case, a re-grasp was demonstrated, and then the task was continued as usual. In the second case, the robot was shown how to grasp the leg at a closer point, pull it towards itself, and then re-grasp it at the desired location.

After the interactive corrections were collected, the old data were re-segmented with the two new corrections and used to re-build the FSA. Using this new FSA, the robot was able to recover from two types of errors in novel situations. Finally, Fig. 6 shows a full successful execution of the task without human intervention, demonstrating that these error recovery capabilities did not interfere with smooth execution in cases where no contingencies were encountered.

Flexible discovery and sequencing of primitives is essential for tractable learning of complex robotic tasks from demonstration. Sequencing primitives with an FSA allows exemplars of movement primitives to be grouped together in a semantically meaningful way that attempts to maximize data reuse, while minimizing the number of options that the agent must choose amongst at each step. This approach makes the sequencing classification task easier, while also providing a mechanism for semantically grounding each primitive based on state visitation history and observed characteristics like coordinate frame, length, and successor state. In the furniture assembly task using a PR2 mobile manipulator, it was shown that the robot could learn the basic structure of the task from a small number of demonstrations, which were supplemented with interactive corrections as the robot encountered contingencies that would have lead to failure. The corrections were then used to refine the structure of the FSA, leading to new recovery behaviors when these contingencies were encountered again, without disrupting performance in the nominal case.

IV. AUTONOMOUS SKILL ACQUISITION

A core research goal in robot learning is the development of skill discovery methods whereby agents can acquire their own high-level skills through interaction with the environment in the context of solving problems and in the absence of explicit instruction or demonstration. Konidaris et al. [9] described an algorithm called CST that was able to acquire skills from demonstration trajectories on a mobile robot. A precursor of the method of Niekum et al. [19] described above, CST segments trajectories into chains of

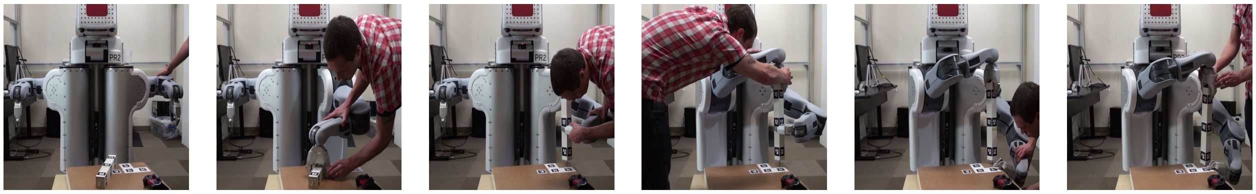


Fig. 5. A kinesthetic demonstration of the table assembly task.

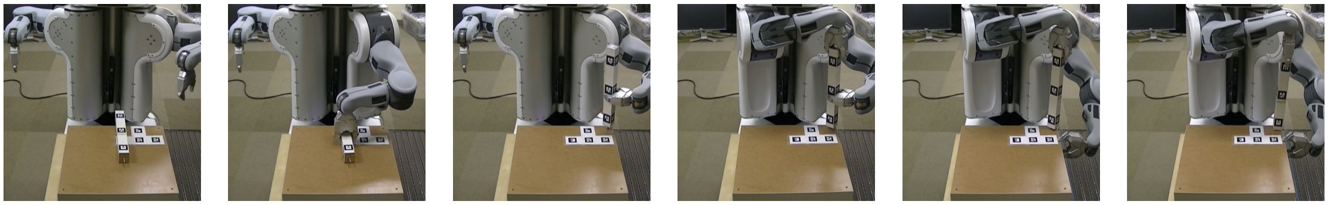


Fig. 6. A full successful execution of the task without any human intervention.

skills, allocating each its own abstraction (out of a library of available abstractions), and merges chains from multiple trajectories into a skill tree. Konidaris [6] (see also ref. [10]) used CST as a component of a robot system that learned to sequence the execution of a set of innate controllers to solve a task and then used the resulting solution trajectories as input to CST, thereby autonomously acquiring new skills through interaction with its environment. This work further demonstrated that the robot was able to reduce the time required to solve a second task by transferring the acquired skills.

A. CST

CST segments each trajectory into a chain of skills—allocating each skill its own abstraction—and merges chains from multiple trajectories into a single skill tree; this is accomplished incrementally and online. CST uses a library of abstractions, and segments each trajectory by automatically detecting when either the most relevant abstraction changes, or when a segment becomes too complex to represent using a single linear value function approximator.

Each skill's initiation set (the set of states in which each skill can be initiated) is obtained using a classifier: states in its segment are positive examples and all other states are negative examples. Each skill termination condition is the initiation set of the skill that follows it (or the target of the trajectory, in the case of the final skill), resulting in a chain of options that can be executed sequentially to take the robot from its starting position to the goal.

CST is suitable for skill acquisition in mobile robots because it is online, and given an abstraction library it segments demonstration trajectories into sequences of skills that are each represented using a small state space. This use of skill-specific abstractions is a key advantage of the approach because it allows problems that are high-dimensional when considered monolithically to be adaptively broken into subtasks that may themselves be low-dimensional [8]. Additionally, a change in abstraction is a useful measure of subtask boundaries, and the use of agent-centric abstractions facilitates skill transfer [7].

B. The Red Room Tasks

With the uBot-5 shown in Fig. 1 equipped with two cameras mounted on a pan/tilt unit, Konidaris [6] used a pair of tasks to demonstrate the feasibility of autonomous robot skill acquisition and the effect of acquired skills. In the first, the robot learned to sequence the execution of a set of innate controllers to solve a mobile manipulation task and then extracted skills from the resulting solution. Performances of the robot with and without the acquired skills in a second, similar, task were compared.

The first task consisted of a small room containing a button and a handle. When the handle was pulled after the button had been pressed a door in the side of the room opened, allowing the uBot access to a compartment containing a switch. The goal of the task was to press the switch. Fig. 7 shows a drawing and photographs of the first task.

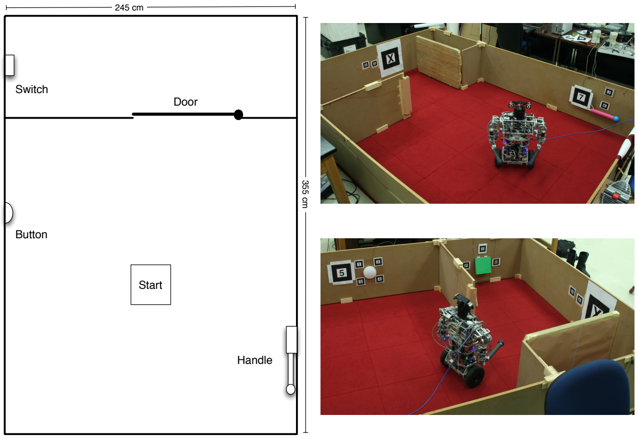


Fig. 7. The first task in the Red Room Domain.

The second Red Room task was similar to the first: the robot was placed in a room with a group of manipulable objects and a door. In this case, the robot had to first push the switch, and then push the button to open the door. Opening the door hid a button in the second part of the room. The robot had to navigate to the second part of the room and pull a lever to close the door again. This revealed the second button, which it had to press to complete the task. Since this room contained the same object types as the first task, the robot was able to apply its acquired skills to manipulate them.

To solve each task, the uBot learned a model of the task as a Markov decision process (MDP). This allowed the uBot to plan online using dynamic programming, resulting in a policy that sequenced its innate controllers. It had to learn both how to interact with each object and in which order interaction should take place. The robot was able to acquire the optimal controller sequence in 5 episodes, reducing the time taken to solve the task from approximately 13 minutes to around 3. The resulting optimal sequence of controllers were then used to generate 5 demonstration trajectories for use in CST. CST extracted skills that corresponded to manipulating objects in the environment, and navigating towards them.

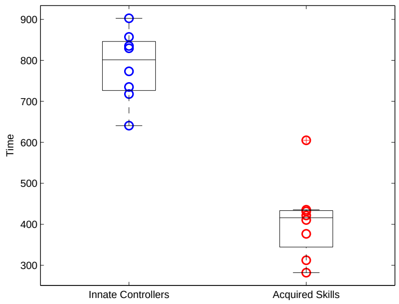


Fig. 8. The time required for the uBot-5 to first complete the second task, given innate controllers or acquired skills.

Figure 8 shows the time required for the uBot's first episode in the second task, given either its original innate controllers or, additionally, the manipulation skills acquired in the first Red Room task. The presence of acquired skills nearly halved the mean time to completion (from 786.39 seconds to 409.32 seconds).

V. Conclusion

In addition to their separate innovations, these illustrations show how behavioral modularity can be exploited to facilitate robot learning. Risk sensitive Bayesian optimization (Sec. II) permits rapid refinement of an identified behavior; automatic segmentation from demonstrations (Sec. III) is a practical way to identify behavioral modules that can be flexibly exploited; and the Red Room tasks (Sec. IV) illustrate how modules can be autonomously identified and refined to permit effective transfer across related tasks. Fully integrating these methods is a subject of ongoing research.

Videos of these demonstrations are available at `http://people.csail.mit.edu/scottk/`, `http://people.cs.umass.edu/~sniekum/`, and `http://people.csail.mit.edu/gdk/arsa.html`.

References

[1] B. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[2] E. Brochu, V. Cora, and N. de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR*, abs/1012.2599, 2010.

[3] P. Deegan. *Whole-Body Strategies for Mobility and Manipulation*. PhD thesis, University of Massachusetts Amherst, 2010.

[4] E. Fox, E. Sudderth, M. Jordan, and A. Willsky. Sharing features among dynamical systems with beta processes. *Advances in Neural Information Processing Systems 22*, pages 549–557, 2009.

[5] D. Grollman and O. Jenkins. Incremental learning of subtasks from unsegmented demonstration. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 261–266, 2010.

[6] G. Konidaris. *Autonomous Robot Skill Acquisition*. PhD thesis, Computer Science, University of Massachusetts Amherst, 2011.

[7] G. Konidaris and A. Barto. Building portable options: Skill transfer in reinforcement learning. In M. Veloso, editor, *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 895–900, 2007.

[8] G. Konidaris and A. Barto. Efficient skill learning using abstraction selection. In C. Boutilier, editor, *Proceedings of the Twenty First International Joint Conference on Artificial Intelligence*, pages 1107–1112, 2009.

[9] G. Konidaris, S. Kuindersma, A. Barto, and R. Grupen. Constructing skill trees for reinforcement learning agents from demonstration trajectories. In *Advances in Neural Information Processing Systems 23*, pages 1162–1170, 2010.

[10] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto. Autonomous skill acquisition on a mobile manipulator. In *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence (AAAI-11)*, pages 1468–1473. AAAI Press, 2011.

[11] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto. Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 31(3):360–375, 2012.

[12] S. Kuindersma. *Variable Risk Policy Search for Dynamic Robot Control*. PhD thesis, University of Massachusetts Amherst, September 2012.

[13] S. Kuindersma, R. Grupen, and A. Barto. Learning dynamic arm motions for postural recovery. In *Proceedings of the 11th IEEE-RAS International Conference on Humanoid Robots*, pages 7–12, Bled, Slovenia, October 2011.

[14] S. Kuindersma, R. Grupen, and A. Barto. Variable risk dynamic mobile manipulation. In *RSS 2012 Mobile Manipulation Workshop*, Sydney, Australia, July 2012.

[15] S. Kuindersma, R. Grupen, and A. Barto. Variational Bayesian optimization for runtime risk-sensitive control. In *Robotics: Science and Systems VIII (RSS)*, Sydney, Australia, July 2012.

[16] S. Kuindersma, E. Hannigan, D. Ruiken, and R. Grupen. Dexterous mobility with the uBot-5 mobile manipulator. In *Proceedings of the 14th International Conference on Advanced Robotics*, Munich, Germany, June 2009.

[17] M. Lázaro-Gredilla and M. Titsias. Variational heteroscedastic Gaussian process regression. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2011.

[18] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. A. Castellanos, and A. Doucet. A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, 27:93–103, 2009.

[19] S. Niekum, S. Osentoski, S. Chitta, B. Marthi, and A. G. Barto. Incremental semantically grounded learning from demonstration. In *Robotics: Science and Systems 9*. 2013. To appear.

[20] S. Niekum, S. Osentoski, G. Konidaris, and A. Barto. Learning and generalization of complex tasks from unstructured demonstrations. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5239–5246, 2012.

[21] J. Peters and S. Schaal. Policy gradient methods for robotics. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 2219–2225, 2006.

[22] M. Tesch, J. Schneider, and H. Choset. Using response surfaces and expected improvement to optimize snake robot gait parameters. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, 2011.