Intrinsically Motivated Affordance Learning

Stephen Hart   Rod Grupen
Laboratory for Perceptual Robotics
University of Massachusetts Amherst
Amherst, MA 01003
{shart,grupen}@cs.umass.edu

I. Introduction

This paper presents an intrinsic motivation function called the *multi-modal imperative* (MMI) that can be used by sensorimotor systems to learn deep control knowledge about behavioral affordances [1]. It builds upon the *control basis framework* and has been used to teach the bimanual robot Dexter (Figure 1) general purpose manipulation skills and commonsense knowledge about objects.

In the following sections, we describe the control basis and summarize existing and new work using the multi-modal imperative. In particular, we demonstrate how the MMI can govern the acquisition of a series of hierarchical manipulation skills [3], generalize those skills to new situations [2], and govern the exploration of affordance-based memory structures concerning objects.

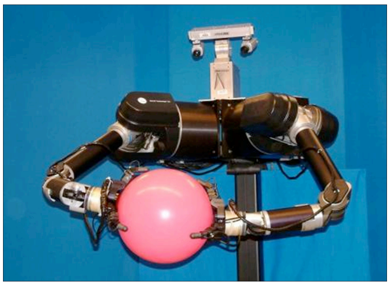

Fig. 1. The bimanual robot Dexter.

II. The Control Basis

The *control basis* framework provides a combinatoric means of constructing hierarchical and multi-objective closed-loop programs from a robot's sensory and motor resources, and is diagrammed in Figure 2. The control basis framework supports principled mechanisms for the following:

- **Parameterizable Control Actions:** Primitive actions in the control basis framework are closed-loop feedback controllers constructed by combining a potential function $\phi \in \Omega_\phi$, with a feedback signal $\sigma \in \Omega_\sigma$, and motor variables $\tau \in \Omega_\tau$ into a control action $c(\phi, \sigma, \tau)$. In any such configuration, $\phi(\sigma)$ is a scalar potential function (e.g., a *navigation* function [5]) defined to satisfy properties that guarantee asymptotic stability.
- **Co-Articulation:** Multi-objective control actions that support co-articulated behavior are constructed by combining control primitives in a prioritized manner. Concurrency is achieved by projecting subordinate/inferior actions into the nullspace of superior actions, and is denoted $c_{inf} \triangleleft c_{sup}$. This prioritized mapping assures that inferior control inputs do not destructively interfere with superior objectives and can be extended to n-fold concurrency relations.
- **State Estimation:** The error dynamics $(\phi, \dot{\phi})$ created when a controller interacts with the task domain supports a natural discrete abstraction of the underlying continuous state space [4]. One simple discrete state definition based on *quiescence events* and controller relevance was proposed in [3]. Quiescence events occur when a controller reaches an attractor state in its potential. A collection of n distinct primitive control actions, therefore, define a discrete, robot-centric state/action space from which programs can be assembled.
- **Behavioral Programming:** Sensorimotor programs are learned in the control basis framework given the state and action spaces $\mathcal{S}$ and $\mathcal{A}$ defined by the set $\{\Omega_\phi, \Omega_\sigma, \Omega_\tau\}$ and a reward function $\mathcal{R}$. Formulating the learning problem as a Markov Decision Process (MDP), allows a learning agent to estimate the value, $\Phi(s, a)$, of taking an action a in a state s using reinforcement learning (RL) techniques [6]. Representing behavior in terms of a value function provides a natural hierarchical representation for control basis programs where attractor states of the value function capture quiescence events in the policy. As a result, the state of a program can be captured using the same state representation as above, even though that program may have its own complex transition dynamics.

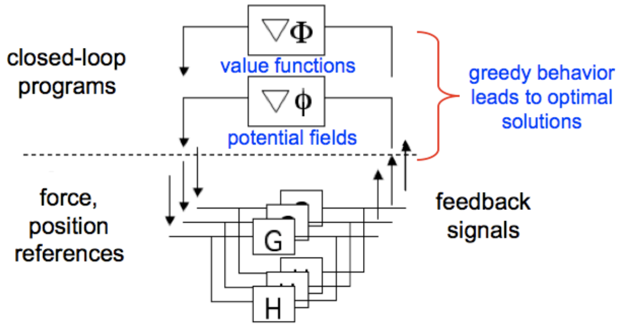

Fig. 2. This diagram shows our hierarchical control architecture. Control actions greedily descend local-minima-free potential fields, sending reference signals to low-level feedback loops that guarantee stable performance at the hardware level. Programs written on top of these control applications descend value functions that guarantee adaptive-optimal performance.

III. INTRINSICALLY MOTIVATED AFFORDANCES

In [3], we provide an intrinsic reward function called the *multi-modal imperative* that provides a measure of *value* to control basis programs. It rewards control actions that afford controllable interaction with the environment—measured through a controller's dynamics—and allows robots to *autonomously* acquire rich behavioral knowledge. the multi-modal imperative provides reward for the following criteria:

- **Stable Control Response:** Reward occurs if the robot can respond stably to feedback it receives from its environment. We capture such stability in terms of controller *quiescence*.
- **Stimuli Regulation:** Reward occurs only for quiescence events on controllers that reduce input errors from *direct* sensory feedback signals. In other words, those that achieve a degree of *stimuli regulation* between the robot and its environment.
- **Deep Knowledge Construction:** Reward increases as the robot discovers rich areas of stable control events (e.g., a manipulatable object). To achieve this criteria, we define a memory structure called a *catalog*, $\mathcal{C}$, that records collections of rewarding control affordances that occur together with some regularity.

In the next sections, we briefly describe how this single MMI has been used to address three different aspects of sensorimotor learning in the same unifying framework.

A. Hierarchical Manipulation

The multi-modal imperative has successfully been used to teach the bimanual robot Dexter (Figure 1) the following manipulation behaviors—many of which employ other of these same behaviors hierarchically—through a series of learning stages:

- SEARCHTRACK: This program allows the robot to posture itself in configurations where various stimuli tend to occur in the environment and then track that stimuli. It has been applied successfully to allow Dexter to saccade to and track various visual cues (such as highly saturated regions of interest), or to move its fingers into contact with objects it can grab.
- REACHGRAB: This program allows the robot to find objects in its workspace, to reach out to them, and to grab them.
- VISUALINSPECT: This program allows Dexter to pick up objects and move them to a place where its stereo cameras are well conditioned to observe those objects with high acuity.
- HANDTRANSFER: This allows Dexter to transfer grabbed objects between its two hands.
- PICKANDPLACE: This allows Dexter to pick up objects and place them in other locations, while controlling the interaction forces between the object and goal. It provides the general structure multi-object interactions such as stacking, inserting, etc.

The learning experiments for some of these behaviors are reported in [3] and forthcoming publications.

B. Transfer and Generalization

The multi-modal imperative also provides a means to structure the generalization of control programs to novel situations [2]. Generalization is possible because the sensory and effector resources, Ω_σ and Ω_τ, in the control basis adhere to strict typing constraints such that only certain sensors and certain effectors may be combined with a certain objective functions.

Generalization is achieved by factoring learned policies into *declarative* and *procedural* parts. The declarative component maintains a policy of the typed objective functions that will achieve reward, while procedural component provides information pertaining to which sensory and motor resources should be "attached" to those objective functions based on the run-time context.

The result is that programs learned in one situation can bootstrap learning in a different, but related context. [2] demonstrated how the program REACHGRAB learned using Dexter's left hand, was generalized to afford situations where the robot could employ right-handed are bimanual grasps on objects based on reference inputs to the robot's control actions pertaining to the size, location, and velocity of the objects to be acquired.

C. Affordance-Based Memory

Finally, the single MMI intrinsic reward function has been used to govern the development of Dexter to produce behavior that is typically achieved in the literature only through separate intrinsic reward functions for *habituation*, *novelty*, and *surprise*. A complete description of this work will be provided in a forthcoming publication.

To achieve this, Dexter builds statistical distributions about the real-valued inputs to rewarding control actions that engage objects, capturing which situations do and do not lead to reward. These distributions are captured in the catalog $\mathcal{C}$, and have their own dynamics. Models that have high variance encourage the robot to engage objects more often; as models become more predictable, the robot will habituate on those objects. If the predictability later changes, the robot will be "surprised" and re-engage the objects.

REFERENCES

[1] J. Gibson. The theory of affordances. In *Perceiving, acting and knowing: toward an ecological psychology*, pages 67–82, Hillsdale, NJ, 1977. Lawrence Erlbaum Associates Publishers.

[2] S. Hart, S. Sen, and R. Grupen. Generalization and transfer in robot control. In *8th International Conference on Epigenetic Robotics (Epirob08)*, 2008.

[3] S. Hart, S. Sen, and R. Grupen. Intrinsically motivated hierarchical manipulation. In *Proceedings of the 2008 IEEE Conference on Robots and Automation (ICRA)*, Pasadena, California, 2008.

[4] M. Huber and R. Grupen. Learning to coordinate controllers - reinforcement learning on a control basis. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Nagoya, JP, August 1997. IJCAI.

[5] D.E. Koditschek and E. Rimon. Robot navigation functions on manifolds with boundary. *Advances in Applied Mathematics*, 11(4):412–442, 1990.

[6] R. Sutton and A. Barto. *Reinforcement Learning*. MIT Press, Cambridge, Massachusetts, 1998.