Bringing up robot: Fundamental mechanisms for creating a self-motivated, self-organizing architecture

Douglas Blank & Deepak Kumar
Department of Math & Computer Science
Bryn Mawr College
Bryn Mawr, PA 19010
{dblank,dkumar}@brynmawr.edu

Lisa Meeden
Computer Science
Swarthmore College
Swarthmore, PA 19081
meeden@cs.swarthmore.edu

Abstract

A significant pitfall exists in task-oriented robot design—an inherent anthropomorphic bias. Traditional research in the design of robots has attempted to get robots to do the tasks a human can, and to do them in the way a human would. The pitfall in this approach is that our human conceptualization of how to solve a particular task is firmly grounded in our unique sensory experiences and motor skills. A robot that is equipped with very different sensor and motor capabilities cannot easily share our conceptualizations. Instead, it is more fruitful to eliminate the anthropomorphic bias by adopting a developmental approach. This paper proposes a multi-level, cascaded discovery and control architecture for developmental robotics. The key components of this developmental architecture are mechanisms for abstraction and anticipation in the context of a model of self-motivation.

1 Introduction

Most intelligent robotics control systems begin with the goal of creating a robot to carry out human-issued tasks. These tasks vary in difficulty but must, by their very nature, involve abstract concepts. For example, typical tasks might be: go to a specific location, identify an object, or pick up something. Attempting to directly achieve the goal of carrying out human commands creates basic assumptions about fundamental architectural design of a robot. We call this philosophy *task-oriented design*.

Inside the task-oriented design paradigm, there are two competing methodologies: top-down, and bottom-up. Top-down designers apply computational algorithms that can be carried out on the robots so as to accomplish a given task. The range of computational models employed varies in robotics: dead reckoning (e.g., using internal measures of space), sensor fusion, behavior fusion, and symbolic AI driven models.

Bottom-up designers again usually take the task to be performed by the robot as a prespecified assumption. However, the control architecture of the robot is designed in a bottom-up fashion. Examples include: subsumption architectures, supervised learning schemes, and evolutionary computation.

We believe that a significant pitfall exists in both the top-down and bottom-up task-oriented robot design methodologies: *inherent anthropomorphic bias.* This bias refers to the design of prespecified robot tasks: traditional research in the design of intelligent robots has attempted to get robots to do the tasks a human can, and do it in a human-centered manner. Historically, this methodology started out by imitating the physical actions of a child playing with blocks. A task was decomposed into a *planning* problem, and then, with a robot equipped with an arm and a gripper, the robot was asked to manipulate specific blocks. The inherent anthropomorphic bias existed by design since the issue was to explore models of intelligent behavior. The pitfall in this approach is that the symbolic modeling of behavior is anthropomorphized based on the capabilities of a human body and human concepts. Both capabilities may be inappropriate assumptions for the physical body and experiences of the robot.

Furthermore, even if we could build a robot with a human-like body and senses, it is not clear that we can jump straight away to the abstract task at hand. Many control issues need to be solved in order to have a robotic system carry out even the simplest of these tasks. After a half-century of continued research, the artificial intelligence and robotics communities have not yet developed any type of general purpose intelligent system.

More recently, a new approach called *developmental robotics* is being applied to the design of robot behaviors. In this approach, an artifact under the control of an intrinsic developmental algorithm discovers capabilities through autonomous real-time interactions with its environment using its own sensors and effectors. That is, given a physical robot or an artifact, behaviors (as well as mental capabilities) are *grown* using a developmental algorithm. The kinds of behaviors and mental capabilities are not explicitly specified. The focus is mainly on the *intrinsic developmental algorithm* and the computational models that allow an artifact to grow.

A developmental approach to robotics is partly an attempt to eliminate the inherent anthropomorphic bias. By exploring the nature of development, the robot is essentially freed from the task of achieving a specific goal. As long as the intrinsic developmental algorithm demonstrates growing behavior there is no need to prespecify any particular task for the robot to perform. Indeed, it is the goal of developmental robotics to explore the range of tasks that can be learned (or grown) by a robot, given a specific developmental algorithm and a control architecture. This paper outlines our approach to a developmental robotics program.

2 Overview

The ultimate goal of our developmental robotics program is to design a control architecture that could be installed within a robot so that when that robot is turned on for the first time it initiates an ongoing, autonomous developmental process. This process should be unsupervised, unscheduled, and taskless and the architecture should work equally well on any robot platform—from a fixed robot arm, to a wheeled robot, to a legged robot.

The intrinsic developmental process we are currently exploring contains two essential mechanisms—*abstraction* and *anticipation*. In a realistic, dynamic environment, a robot is flooded with a constant stream of perceptual information. In order to use this information effectively for determining actions, a robot must have the ability to make abstractions so as to focus its attention on the most relevant features of the environment. Then, based on these abstractions, a robot must be able to anticipate how the environment will change over time, so as to go beyond simple reflexive behavior to purposeful behavior.

The developmental process is employed in a hierarchical, bootstrapping manner, so as to result in the discovery of a range of increasingly sophisticated behaviors. That is, starting with a basic, built-in innate behavior, the robot exercises its sensors and motors, uses the mechanisms for abstraction and anticipation and discovers simple reflex behavior. A control scheme employs these discoveries to relieve the robot from the innate behavior. This constitutes the first stage of the bootstrapping process.

The same intrinsic developmental algorithm can be employed recursively in subsequent stages, using the knowledge discovered in previous stages. For example, the second stage abstracts *sequences* of behaviors and corresponding perceptual views. These behavior sequences, termed *protoplans* [6], can lead the robot through a series of views in the environment thus resulting in 'interesting' places to visit. We will call these places *protogoals*. Here, the *proto* prefix implies a distinction between standard notions of plans and goals from the developmental ones used here. Once again, a control scheme, not unlike the one used in the earlier stage, employs these discoveries and relieves the robot from the reflex behavior learned earlier. The same developmental process can be cascaded beyond this stage to result in discovery of actual goals and plans.

The control scheme that is responsible for driving the robot at each stage uses the discovered abstractions and anticipations. Additionally, at each stage, the proposed architecture incorporates a model of motivation. At the lowermost level, this model indicates to the system how 'comfortable' it is in the given environment. If it is too comfortable, it becomes bored, and takes measures to move the robot into more interesting areas. Conversely, if the environment is overly chaotic, it becomes over-excited and shuts down the sensations. These anthropomorphic terms will be described below in more technical terms.

Ultimately, the robot will grow up enough to start exhibiting purposeful behaviors. For example, a robot could form a goal of getting to some place, and then be able to plan its behavior to go to it. By its very nature, goal-directed behavior is decomposed using regression mechanisms, as is traditionally done in most research on *AI Planning*. However, the 'planning' performed in such a developmental system is related less to a search than it is to a model of stimulus-response.

To summarize, we are proposing a multi-level, cascaded discovery and control architecture to explore developmental robotics. Each level of the architecture uses an instantiation of the intrinsic developmental algorithm and the control scheme. The key components of the developmental algorithm are the processes of abstraction and anticipation in the context of a model of motivation. In what follows next, we elaborate more on the details of this proposal.

3 The Intrinsic Developmental Algorithm

As in nature, the control architecture is not a completely blank slate, but contains a simple reflexive model for producing behavior, as well as the infrastructure necessary for adaptation. Over time, through self-motivated interactions with the environment, the robot acquires the knowledge necessary to exist in the environment in a purposeful way. The robot learns, not only about its environment, but also about its own perceptual and motor capabilities via this process. This approach to development proceeds hierarchically.

Each level of the hierarchy combines two essential mechanisms—abstraction and anticipation. Our implementation of these two mechanisms is based on two types of neural network models: self-organizing maps and simple recurrent networks, respectively.

3.1 Discovering abstractions

Every robot is endowed with a suite of sensors and effectors. In the process of exploring its environment, data obtained from a robot's sensors and effectors represents the robot's experiences in the environment. As an essential step in the growing process, the robot has to discover abstractions from such data. Even in the case of simple mobile robots, sensorimotor data tends to be very high dimensional. Nonparametric clustering algorithms work well for abstracting high dimensional data. We are using it as our main algorithm for discovering abstractions.

Self-organizing maps (SOMs) were pioneered by Kohonen in the 1980's and 1990's[4]. Briefly, a SOM is a mapping of a typically high-dimensional input vector to a particular cell in a low-dimensional matrix. The matrix is topologically arranged in a unsupervised manner such that very similar input vectors map to the same cell, and slightly similar input vectors map to nearby cells.

Specifically, similarity is computed by comparing an input vector with a model vector associated with each cell. The model vector that is closest (as determined by the smallest sum of squared differences to the input vector) is designated as the winner. The model vector of the winner and the model vectors of the cells in its neighborhood are updated to more closely match the given input vector.

The SOM idea is quite simple and effective. Any information that can be turned into a vector of numeric values can be self-organized into such a map. The idea has been applied in a wide variety of problems ranging from creating maps for classifying the World Wide Web to analyzing financial data. Resulting SOMs are useful for two related reasons: their ability to automatically find abstractions, and their ability to help visualize complex data [4].

3.2 Discovering anticipations

Once a robot has discovered some abstractions, it is important to try and use them to anticipate what would happen next. This can help a robot predict its future and can also be used to help take over control from a lower level. Anticipation is a temporal activity and thus requires a time-sensitive computational mechanism.

The simple recurrent network (SRN) was created by Elman in the late 1980's [2]. To understand its significance, one must realize that there are two main classes of artificial neural networks: those that are feed-forward (all activation flows in one direction) and those that are recurrent (activation is allowed to flow forward and backward). In order to deal with time-dependent tasks, a feed-forward network usually requires a fixed window of the past inputs, while a recurrent network can take the current input alone and build up a contextual memory of the past inputs. Elman's SRN has the simplicity of a feed-forward network with the power of a recurrent network. Like SOMs, SRNs are also simple and effective. They have been applied to analyzing many types of sequential processing, including language and music.

3.3 The Control Scheme

Figure 1 depicts the hierarchical nature of the proposed control architecture. Level 0, which is built-in, contains a set of simple reflexes for controlling the robot. Each subsequent level combines an abstraction mechanism and an anticipation mechanism to adapt to the environment based on experience. The anticipation mechanism has a feedback loop to illustrate its time dependent nature. The outputs from all levels are integrated in a subsumption fashion, with higher levels having priority over lower levels.

Input to the first level of the hierarchy comes directly from the robot's sensors and motors. The abstraction mechanism at this level begins to extract basic perceptual and motor features observed during the robot's initial reflexive movements. The anticipation mechanism observes the abstractions being made and begins to recognize repeated multi-step sequences of features through time, chunking them into new, more compact representations.

The next level of the hierarchy takes these newly created chunked representations as input. Using the same abstraction mechanism, this level begins to make

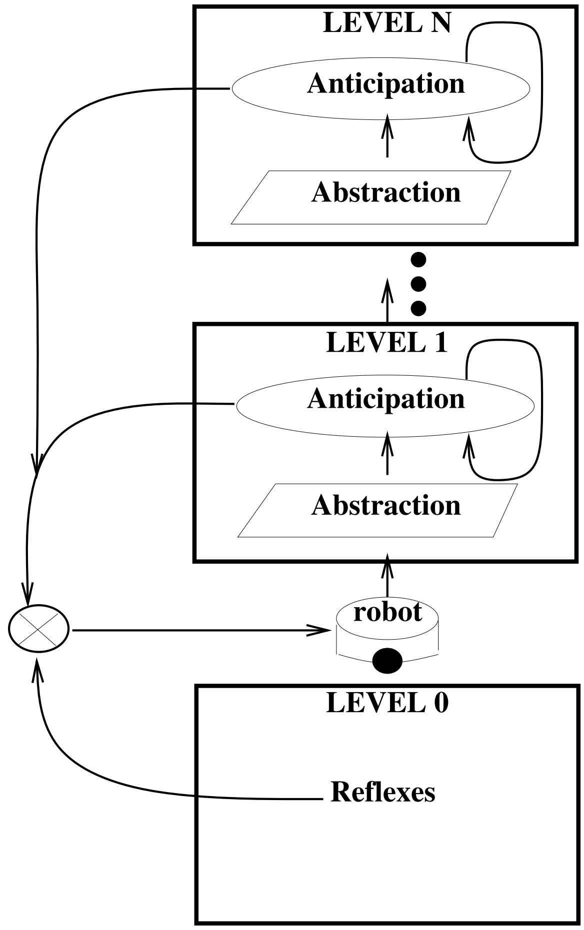

Figure 1: A general developmental robotics architecture

abstractions about these chunked sequences. Using the same anticipation mechanism, this level begins to recognize sequences of sub-sequences from the previous level, chunking them again and sending them on to a further level. In this way, each level of the hierarchy processes the input at a longer time scale than the previous level.

This hierarchical development is driven by internal motivations rather than external goals. One central motivation is to avoid boredom while not straying into chaos, or in other words maintaining a balance between exploitation and exploration. The anticipation mechanism provides a good measure of where the developing system falls along this continuum. When the anticipation mechanism of one level is able to accurately predict the behavior of the previous level, it is time for that level to subsume control of the robot and for exploration to begin at the next level.

3.4 Model of Motivation

Our goal is to find a developmental equivalent to a co-evolutionary competitive arms race. We imagine a mechanism in the system such that it would get "bored" in environments that it can easily predict, but retreat from environments that seem chaotic. However, the area between predictability and randomness (the so-called "edge of chaos") is suspected to be a prime area for learning (see, for example, [5]). The exact nature of such a mechanism has yet to be explored.

We believe that a model of self-motivation could drive the system to continued development. Such internally driven artificial systems are rare. One such mechanism is the competitive arms race of co-evolutionary systems (see, for example, [3]). The basic idea of the competitive arms race is that two populations are pitted against one another, and gradually one-up each other in a spiraling increase of fitness. This works if the two populations begin about equal, and remain relatively even throughout the race.

4 Toward Purposeful Behavior

We have described a proposal for implementing the intrinsic algorithm as a robot developmental control architecture. However, even if the above-described project worked as planned, we are left with a robotics system that explores the world with no real purpose. Of course, reaching such a point would be interesting in its own right, even though the resulting behaviors developed may not be useful to us. Finally, we need to incorporate the human-bias, returning to the task-oriented, goal-based objectives.

The missing functionality is the ability for the system, given a goal, to carry out the steps necessary to get there. The solution to this problem, we believe, is to build in a foundational goal-based mechanism, and incrementally bootstrap up toward full-scale goals.

The level at which purposeful behaviors can be incorporated in our architecture is an open question and remains to be explored. However, we have evidence that protoplans can be used to achieve protogoals [1, 6]. The more interesting issue here is that of *goal creation*. That is, how does a robot commit itself to achieving a specific goal? We expect the motivational component of the control scheme to play a central role in this.

5 Conclusion

> The question 'How does a thing become conscious?' could be put more advantageously thus: 'How does a thing become pre-conscious?'. And the answer would be: 'By coming into connexion with the verbal images that correspond to it'. From Sigmund Freud's *The Ego and the Id* (1927)

We believe that Freud was correct in emphasizing the gradual, interactive nature of the development of consciousness and intelligence. We imagine that if Freud had been a roboticist, he may have answered his own question thus: 'By self-organizing the senses and concepts that correspond to it.'

In this paper, we have argued that position. We have argued that there is a significant pitfall in the traditional task-oriented robot design—an inherent anthropomorphic bias. Because a robot is equipped with very different sensor and motor capabilities, it cannot easily share our conceptualizations.

Furthermore, we have argued to eliminate the anthropomorphic bias by adopting a developmental approach. We proposed a multi-level, cascaded discovery and control architecture for developmental robotics. We have argued that key components of this developmental architecture are mechanisms for abstraction and anticipation in the context of a model of self-motivation.

References

[1] Douglas Blank, Deepak Kumar, and Lisa Meeden. A developmental approach to intelligence. In Sumali J. Conlon, editor, *Proceedings of the Thirteenth Annual Midwest Artificial Intelligence and Cognitive Science Society Conference*, 2002.

[2] J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[3] H. Juillé and J. B. Pollack. Dynamics of co-evolutionary learning. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, pages 526–534. MIT Press, 1996.

[4] T. Kohonen. *Self-Organizing Maps.* Springer, Third Edition edition, 2001.

[5] Christopher Langton. Computation at the edge of chaos: Phase transitions and emergent computation. In Stephanie Forest, editor, *Emergent Computation*, pages 12–37. MIT Press, 1991.

[6] L. Meeden. *Towards planning: Incremental investigations into adaptive robot control*. PhD thesis, Indiana University, 1994.