System Integration with Working Memory Management for Robotic Behavior Learning

Stephen Gordon
Center For Intelligent Systems
Vanderbilt University
Nashville TN, 37235-0131
stephen.m.gordon@vanderbilt.edu

Joe Hall
Center For Intelligent Systems
Vanderbilt University
Nashville TN, 37235-0131
joe.hall@vanderbilt.edu

Abstract – As a robot learns behaviors and task execution, several systems must be in place to allow the robot to store what has been learned as well as to recall learned information. We believe having a long-term memory is essential for our robot to be able to learn tasks and behaviors over time. We also believe that it is important for the robot to have some means of representing the immediate information in the environment. This goes beyond just a world representation and incorporates using short-term memory to track the environment. We have developed a robot that possesses such short-term and long-term memory systems. This robot has been given perceptive abilities with which it can populate its short-term memory and has been taught motion generation as a portion of its long-term memory population. Finally, through the use of a working memory system, we plan to show that our robot can "focus" on pertinent task-related information from each of its memory systems in order to learn how to successfully execute tasks.

Key Terms – short-term memory, long-term memory, working memory, behavior learning

I. INTRODUCTION

Behavior-based robotics became popular in the mid-1980's. Brooks introduced the subsumption architecture in [1] with three key ideas of his work being to:

1. Mimic evolution – Incrementally add layers of behavior and complexity to a system. Behaviors can operate with or without the presence of prior behaviors. Interaction of these multi-layered behaviors produces a new emergent behavior guiding the system's interaction with the environment
2. Tightly couple perception and action in each added layer
3. Minimize interaction between layers

This paradigm worked well for simple mobile robots, however, as the field of humanoid robotics began to emerge Brooks [2] later suggested "in order to act like a human, an artificial creature with human form, needs a vastly richer set of abilities". Because humans are the highest form of intelligence, it is only natural that intelligent robots begin to assume "qualities", so to speak, found in humans.

Some of the qualities we believe that intelligent robots should possess are the ability to: learn, record learned information, represent the world around, "focus" on task related information, and perform tasks or self-generate motion. This covers the relative ideas presented in this paper. The term "focus" in this context refers to the ability to select from the world and from learned information what is important for a given task.

An important reason for giving an intelligent robot the means to represent the surrounding world and for choosing behaviors to interact with this world is that this gives the robot the ability to break from simple reactive architectures to more deliberative ones. As the field of computer vision advances, the robot's representation of the world will similarly advance to incorporate the latest advances and understandings of human perceptive abilities. As understanding of the cognitive abilities and mechanisms in intelligent beings advances, the basic world representations and behavioral interactions with that world can be built upon or modified. What most researchers are focusing on currently is the basic system: the robot's understanding or perceived understanding, of the world, the ability to learn how to interact, to record information, and the ability to focus on relevant chunks of information about the world.

In order to control the robot in the manner discussed in this paper, it is necessary to develop an appropriate control hierarchy or architecture. Albus [3] discussed certain key components necessary for the creation of an intelligent system. Various similar architectures have been present since robotic research began, such as the sense-act, sense-plan-act, or hybrid architectures. Albus, however, proposed the necessity of having a world model, short-term, and long-term memories, and task planners. An excellent example of a system that implements this type of approach is demonstrated by the Animate Agent Architecture of [4]. In this system a world model representing the robot's understanding of the real world is maintained as well as a recorded library of reactive plans and plans steps known as the RAP library [4]. This system demonstrates one coupling of a world model, long-term memory, and task-based decision-making.

With the success of such architectures, researchers have begun combining their efforts with those of cognitive psychologists and neuroscientists. Work by Precott, et al in [5] discusses the implementation of a system for robot control that draws inspiration from the model of the basal ganglia in humans for action selection.

It is this type of work that we find particularly interesting: the combination of engineering design theory and cognitive models for the construction of an intelligent system. Work by Baddeley [6] proposes the idea of a working memory system. This idea is expanded by Noelle and Phillips [7] and adapted for robotics applications. It is this idea of combining a model for working memory developed by cognitive neuroscientist with a world model and database for long-term memory storage that we feel can aid in the creation of an intelligent system. Such architecture implementations have already begun in [8].

However the structure for robotic intelligence is created, one unifying theme in most robotic research today is learning. How can a robot learn and how much should the robot have to learn. The next section will examine these questions.

II. ROBOTIC LEARNING

A. How much should an intelligent robot have to learn?

Certainly it is possible to program several behaviors into a robot in *a priori* fashion. Even general behavioral schemata can be pre-programmed into a robot allowing that robot a diverse range of motion execution. How much pre-programmed initial knowledge and ability should the robot possess? Obviously, there must be some initial knowledge and ability – one can not simply turn on a system and expect it to begin learning – if nothing else, the system must be pre-programmed with the ability to learn.

Work by Schultz and Grefenstette [9] suggest that before programming a robot in any manner, the designer must decide where to draw the line between knowing and learning. Optimally, the robot should be pre-programmed with as much information as the designer can easily program into the system. However, when the cost of endowing a robot with a set of knowledge is outweighed by the difficulty of programming such knowledge then, from a purely engineering point-of-view, it becomes cost effective to allow the robot to learn any further knowledge, abilities, or behaviors it might need by interacting with the environment.

B. What is the best way in which an intelligent robot can learn?

In general there are two ways in which a robot can learn a behavior: learn by example and learn by trial and error. One method for learning by example is through imitation – where a robot attempts to mimic some aspect of a demonstrator's behavior. This technique is implemented by Billard in [10]. This type of learning is beginning to find biological inspiration in the presence of what is called mirror neurons. Mirror neurons are neurons that "fire" when either an action is performed or when that action is observed being performed by another individual. Mirror neurons and imitation learning is discussed at length in [11].

Robots can also learn by being told what to do – having an operator command the robot to perform certain actions. One of the easiest ways to "tell" the robot to do something is through teleoperation. If the robot is appropriately intelligent enough, the robot can generalize behaviors learned through this technique. Work in [12] demonstrates the practical effectiveness of this technique in which the robot is taught, through teleoperation, how to perform a variety of reaching tasks. The robot generalizes the tasks and uses this information to perform similar tasks.

Having the robot learn by trial and error is a somewhat slower process but definitely has advantages. For example, learning through trial and error gives the robot the ability to make it's own generalizations about the environment and the robot's own motion capabilities. In this manner, capabilities can still emerge from the system in unexpected but possibly fortuitous ways. Learning through trial and error requires the robot to have some understanding of what it can do, such as moving it's arm joints, opening and closing it's gripper, etc. and also requires that the robot have some metric for evaluating choices. One of the best ways for this type of learning to be implemented is through reinforcement learning. An excellent example of this is demonstrated in Grupen's work [13] in which a robot begins learning behavioral schema through exploration and trial and error.

Finally, one of the most important, though often times least discussed tools for robot learning is the memory structure. Obviously there is no need to learn if the learned material cannot be recalled for later use. Animals and humans possess a variety of memory structures such as short-term memory structure for day-to-day items, long-term memory structure for storing learned information over long periods of time, and some form of working memory structure useful for maintaining task-related information.

This paper focuses on implementing simple vision learning, behavioral motion learning, and integration of short-term, long-term, and working memory structures to learn how to perform a task in a given situation. It is the summation of all of these parts that will enable our humanoid robot ISAC to begin linking commands with the appropriate responses. ISAC (Intelligent Soft Arm Control) is an on-going humanoid robot (Figure 1) project at the Center for Intelligent Systems at Vanderbilt University.

III. ISAC

A. Hardware

ISAC's hardware, a perceptual and actuator system, is very applicable for this experiment. The perceptual system consists of two cameras providing stereo-vision. Each camera is located atop of an independent pan-tilt unit. The actuator system consists of a 6 DOF right arm. Rubbertuator air muscles power the right arm. These air muscles give the arm the ability to move through space freely, though with the security that if an object appears in its path, the arm will softly collide with the object rather than collide dangerously. ISAC also has two pneumatic hands. Each hand is equipped with four fingers with touch sensors at the tips. There are two proximity sensors located on the palm of each hand.

In addition to this, ISAC possesses the various memory components previously mentioned. ISAC possesses a short-term memory (STM) known as the Sensory EgoSphere (SES) that is used to store percepts that ISAC has detected

in his environment [14] over time. The SES is an extension of Albus' egosphere [3] developed for ISAC by Peters and Kawamura [14] and is represented as a geodesic dome (Figure 2) surrounding ISAC that is indexed by azimuth and elevation angles only. The SES is maintained as a simple MySQL database containing appropriate fields to which data is entered. As ISAC's perceptual processes identify a percept, that percept is placed on the SES at the corresponding location. In this way, the SES represents the world as it appears to ISAC but the SES is also used to recall where objects were previously located over short time periods. This acts as a filter essentially reducing a very complex world to a reasonably small set of interactive objects, the locations of those objects, and the immediate past locations of those objects. If an operator gives ISAC a command that command is stored on the SES. If ISAC recognizes an object visually that object is stored on the SES. If ISAC's arm is in the process of executing a motion, the current position of the arm is stored on the SES.

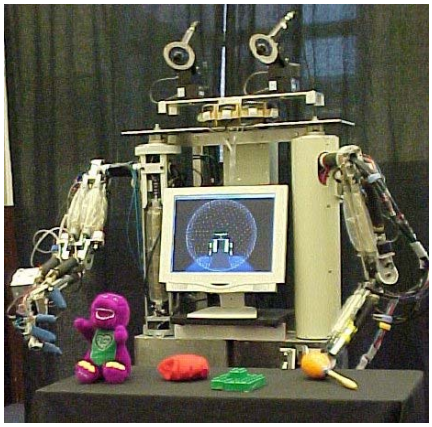

Figure 1: Air muscles enable ISAC's arms to "give" if they collide with an obstacle

ISAC possesses a long-term memory (LTM) that is comprised of two components: a *semantic memory* system that stores descriptive information about objects or percepts and a *procedural memory* system that stores information relating to behaviors and behavior execution. ISAC's LTM is implemented as a MySQL database. The *semantic memory* for a particular percept would store, for example, the type of percept, the known descriptions of the percept, the algorithm used to identify the percept, and the parameters required by that algorithm. The *procedural memory* stores behaviors such as: *reach*, *handshake*, *wave*, *touch*, *grab*, etc. and the parameters for executing those behaviors. For the time being, the LTM is only comprised of items ISAC has learned or been trained to know.

B. Vision Training

ISAC is capable of recognizing different objects in the environment. Using stereo-vision, ISAC is able to determine both the azimuth and elevation of detected objects and the radial distance to the objects. Using a very simplistic HSV detection algorithm, ISAC can be trained to identify simple objects such as bean bags, small toys, cola cans, etc. During training ISAC records the pertinent detected information about the object and the descriptive information provided by the trainer in the LTM. In this manner ISAC is able to, upon necessity, retrieve learned perceptual information, pass the information to the appropriate algorithms, and search for/detect the objects. The objects that ISAC will be trained to recognize for this experiment are: bean bags (both red and blue), green lego toy, and a stuffed Barney doll.

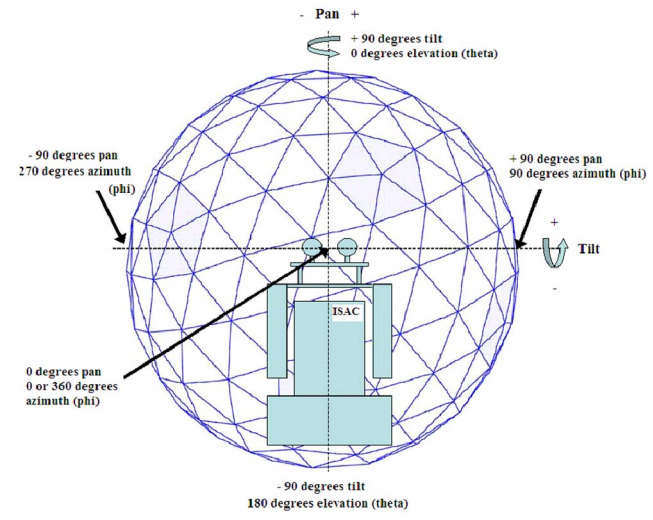

Figure 2: A diagram of the SES indexed by azimuth & elevation angles and used to store sensory information [1 15]

C. Behavior Training

ISAC is also capable of generating it's own motion using a technique known as *verbs and adverbs* [16]. This technique developed by Cohen, Rose, and Bodenheimer [16] and developed for ISAC by Spratley [17] takes motion exemplars, known as *verbs*, and interpolates new motions based on *adverb* values that relate quantitatively to a particular space of the motion. For example, ISAC can be taught (through teleoperation) the behavior *Handshake*. The basic handshake motion would be referred to as the verb, while the adverbs could be: the azimuth angle of the handshake, the frequency of the handshake (the number of times the hand goes up and down), or the speed of the handshake. When recording the motion exemplars, the recorder should attempt to record motions near the edge of each of the adverbs respective space, thereby limiting the motion generation to interpolation rather than extrapolation. The behavior motions ISAC has exemplars of for this experiment are: *handshake*, *reach*, and *wave*.

D. Working Memory

The working memory system (WMS) developed by Noelle and Phillips [7] is designed to emulate a biological working memory system. The working memory system manages data structures called "chunks". These chunks can handle or relate to any type of data. Because of this versatility, chunks are able to store different items such as percepts, behaviors, or other task-related information. The

number of chunks the WMS can hold is specified prior to any experiment.

The WMS uses a set of feature vectors to assist the memory management. This is because the WMS utilizes a neural network for management of the chunks. The values in the feature vector have information about both the state of the system and the chunks in the system. These values are translations of ISAC's current state and of the current chunks.

The working memory's neural network uses temporal-difference learning to learn the appropriate chunks to load given the current state of the system. The WMS is implemented via a working memory toolkit (WMtk) [7] that interacts with ISAC's SES and LTM databases for chunk selection.

E. Control Architecture

The architecture shown in Figure 3 was implemented for this experiment. This architecture centers around the WMS. It is this system that enables ISAC to make the appropriate connections between tasks commands and focus.

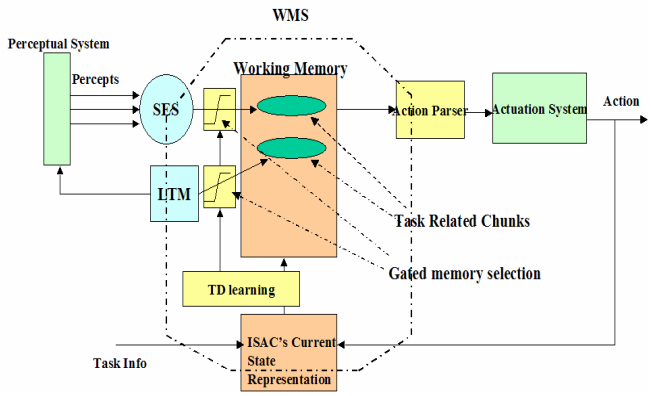

Figure 3: Experimental Control Architecture

This system implements the perceptual system, the motion generation (actuation) system, SES, and LTM systems. Figure 3 shows the SES and LTM as virtual components of the WMS because the WMS can only be comprised of information found in either the SES or LTM. Additionally, part of this WMS is the system's state representation that is used to monitor the task-related chunks kept in working memory. TD learning allows the system, over time, to load the chunks for which it expects the most reward given the current state. Finally, there is an action parser that identifies which chunk is a behavioral chunk as well as which chunk is a perceptual chunk and sends the appropriate command (such as "perform action X on percept Y") to the actuation system.

IV. EXPERIMENT

As previously discussed, this experiment involves the integration of a variety of components; the steps for this experiment are as follows:

1. Prior to this experiment, ISAC is given certain initial knowledge:
   a. ISAC's perceptual system is trained to recognize specific objects (colored bean bags). This information is stored in the semantic memory section of ISAC's LTM.
   b. Using *verbs and adverbs* [16, 17] ISAC, is taught a small set of motion behaviors including how to *reach*. This information is stored in the procedural memory section of ISAC's LTM.
2. ISAC is presented with two bean bags placed on a table as shown in Figure 4:

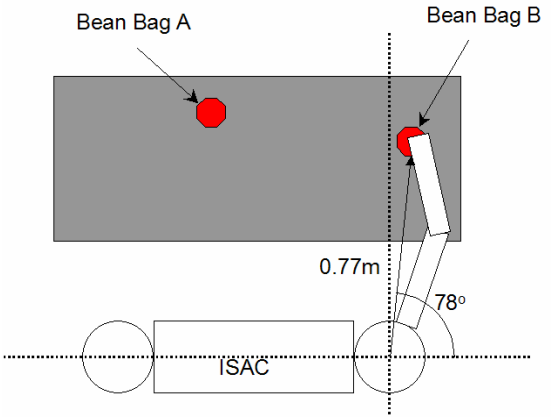

Figure 4: Sample configuration for reaching

3. ISAC is given the command "*reach* to the bean bag". The specific bean bag that ISAC is to reach to is be specified.
4. ISAC's perceptual system recognizes the bean bag objects and post the information to the SES.
5. ISAC attempts to load the relevant "chunks" into it's working memory system. Two chunks are required by the WMS: one chunk to specify the appropriate behavior (*reach*, *handshake*, or *wave*) and another chunk to specify the percept to act upon.
6. The reward rule in the WMtk gives reward based upon the completion of the action.
7. Over time ISAC learns which of the two bean bags is the most appropriate and that the *reach* behavior best accomplishes the task. The *handshake* behavior should, however, develop as the next best choice for reaching to the bean bag.
8. Once ISAC has demonstrated that it has learned the most appropriate chunks to load into the WMS, the bean bags are rearranged (Figure 5), and ISAC is again given the command "reach to the bean bag".

If this experiment is successful, when the bean bags are rearranged ISAC should not necessarily *reach* to the same bean bag as before but rather should choose the bean bag percept that is the most appropriate. The correct behavior should still be chosen. That percept and behavior will be used to execute the action.

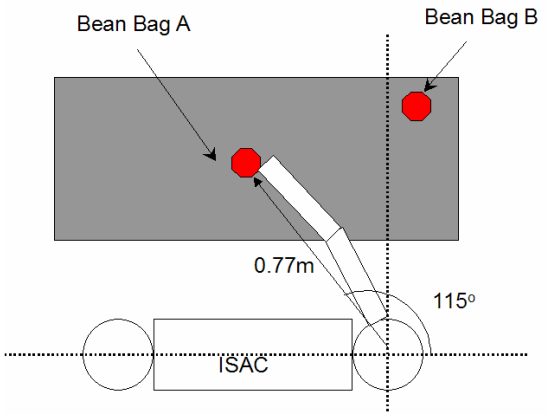

Figure 5: Second sample configuration

V. PERFORMANCE EVALUATION

For each of the trained behaviors, ISAC interpolates the behavior in order to act upon a desired object. In other words, ISAC can *reach* to an object, *handshake* in the direction of an object, or *wave* at an object.

The reward rule used for this experiment was based on three criteria:

1. What was the degree of success for the behavior the WMS chose to load?
2. How well did the object chosen by the WMS meet the task criteria? e.g., for the task "reach to bean bag, as long as ISAC focused on a bean bag, reward was given.
3. How well was ISAC able to act on the object? e.g., in this experiment, could ISAC reach the bean bag?

In order to measure reward criterion #3, the reward was given based on the inverse proportion of the distance from ISAC's hand to the object. Reward criteria #1 and #2 gave a discrete positive valued reward if the system chose appropriately. No preference (i.e., reward of 0) was the result if the system did not choose correctly. The values for the overall reward typically fell in the range of 0-400. Since, it was desired to give negative reward to the system when it did not act appropriately, a negative weighting factor of –200 was added to the final reward to "tilt" the low values into the negative range.

Note that using these reward criteria, it is possible to incorrectly reward the system for performing the task in a less than optimal manner. For example, if the system were to *handshake* or *wave* the appropriate bean bag and if this action happened to bring the hand very close to the bean bag, then the system would receive a positive reward. Over time, if this happens enough then the system would learn that *handshake* or *wave* for some reason better accomplishes the *reach* task.

Initial trials were performed in simulation to speed-up initial testing of the system. The simulation removed the time-bottleneck of generating and performing motions. If the simulation attempted to act on an object within ISAC's workspace, it was assumed that ISAC was able to reach to the object (reward criterion #3).

The action taken was determined by what the WMS currently believed was the best choice. In other words, the action that the WMS believed would yield the greatest reward. This system also contained an exploration percentage, specified as a part of initial knowledge that determined the percentage of trials that the WMS chose a new or different action. This enabled the WMS to always continue learning and exploring.

During the trials, the simulation was not allowed to choose the same action more than twice. This constraint enabled a much quicker simulation time. Once the system had explored the space of actions, the system was restarted with the learned information and given the task "*reach* to the bean bag". For each arrangement (Figure 4 and Figure 5) the system chose appropriately to reach towards the correct bean bag, i.e. the nearest one. Table 1 shows the contents of ISAC's SES and LTM during the training portion of the simulation.

SES	LTM
1. Bean bag: location = (Figure 5), type = A	1. *reach*
2. Bean bag: location = (Figure 4), type = B	2. *handshake*
	3. *wave*

Table 1: Simulation memory contents during training

Given the task, the WMS was allowed to choose two "chunks" from the short-term and long-term memory systems to accomplish the task. However, the WMS was not restricted to choosing exactly one object and one behavior. If the working memory chose to focus on two objects or two behaviors, then respectively, a behavior or object was chosen at random to ensure that an action was still performed. The reasoning behind this was so that the system did not learn to simply choose combinations that lead to no reward, a situation that could be preferred if the WMS was consistently getting negative reward for its choices. Table 2 shows the contents of the WMS in these trials.

	Working Memory Contents			
	Trial 1	Trial 2	Trial 3	Trial 4
Chunk 1:	bean bag: A	bean bag: B	*wave*	*handshake*
Chunk 2:	*reach*	bean bag: A	bean bag: B	bean bag: A
Random:	NA	*handshake*	NA	NA
Reward:	203.4	-20.5	-197.7	2.3

Table 2: Working memory contents during simulation training

To evaluate system performance, a third task was developed. For this task ISAC was again given a command to "*reach* to the bean bag", however the *reach* behavior was deleted from the LTM limiting the behavior choices to *handshake* and *wave*. The WMS had to choose the *next best* behavior. For each of the arrangements shown previoiusly (Figures 4 and 5), the WMS chose to perform the *handshake* behavior. This behavior was chosen because it allowed the arm to get closest (reward criterion #3) to the bean bag (reward criterion #2) and thus best accomplished the task.

After the initial training, ISAC was allowed to perform the generated motions. Two new objects (a green Lego toy, and a purple Barney doll) were added to the table at random positions. ISAC's vision system was trained (Step 1) to

recognize each new object and recorded the type of object as well as some simple descriptive information (color=green, purple; toy type=Barney, Lego). ISAC was given tasks (Step 3) such as "*reach* to the bean bag" or "*reach* to the toy". Each of these tasks did not specify to which bean bag or toy ISAC was to *reach*. ISAC recognized the objects (Step 4). The WMS focused on "chunks" of information from the SES and LTM in order to accomplish the task (Step 5). ISC was allowed to explore the space of possible actions receiving reward each time (Steps 6 and 7). After this was accomplished, the objects were rearranged in a variety of different positions (Step 8) and ISAC was given a command. The results (set of 20 commands) were that ISAC successfully performed the correct action on the nearest (easiest to reach) requested object.

VI. CONCLUSIONS

What is so special about this system? Obviously for the small set of possible behaviors and percepts the appropriate response could have been more easily attained by hard-coding the responses into a table [9]. However, that was not our intention. Our goal is to develop an intelligent robot that utilizes various structures to learn how to perform in its environment. The SES and LTM systems are excellent tools for developing our intelligent robot. Utilizing the LTM gives our robot a wide capacity for learning. The WMS then should enable our robot to "focus" its abilities during task execution.

The experiments discussed in this paper are clearly simplified ones. Numerous more behaviors need to be taught to ISAC and different reward rules need to be specified to capture different types of behavior. These reward rules should be fed into the system for different types of tasks.

As this work progress, over time ISAC will learn how to perform specific tasks by learning what to percepts and behaviors to focus on. This information will be stored in the LTM enabling ISAC to accumulate ability. This paper represents our work in starting this process in an assembled system.

ACKNOWLEDGEMENTS

This work was supported in part under NSF grant EIA0325641, "ITR: A Biologically Inspired Adaptive Working Memory System for Efficient Robot Control and Learning". This work was conducted under the tutelage and guidance of Dr. Kazahiko Kawamura.

REFERENCES

[1] Brooks, R. A., "A Robust Layered Control System for a Mobile Robot", *IEEE J. Rob, Autom. 2,* 14-23, 1986.

[2] Brooks, R.A., "Behavior-Based Humanoid Robotics", *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, IROS, 1-8, 1996.

[3] Albus, J.A., "Outline for a Theory of Intelligence", *IEEE Trans. on Systems, Man, and Cybernetics*, 21(3), 473-509, 1991.

[4] Firby, R.J., "Modularity Issues in Reactive Planning", In Proc. *3rd Int'l Conf. on AI Planning Systems*, B. Drabble (ed.), 78-85, Menlo Park, CA: AAAI Press,

[5] Precott, T.J., K. Gurney, F. Montes-Gonzalez, M. Humphries, and P. Redgrave, "The Robot Basal Ganglia: Action selection by an embedded model of the basal ganglia." In L. F. B. Nicholson and R. Faulls (Eds.) *Basal Ganglia VII*. NY: Plenum Press, 2002.

[6] Baddeley, A., "Working Memory", *Oxford Psychology Series, 11,* Oxford: Clarendon Press, 1986.

[7] Phillips, J. L. and D. Noelle, "A Biologically Inspired Working Memory Framework for Robots", *Proc. of the 27th Annual Conf. of the Cognitive Science Society*, 1750-1755, 2005.

[8] Ratanaswasd, P., W. Dodd, K. Kawamura, and D. Noelle, "Modular Behavior Control for a Cognitive Robot", *12th Int'l Conf. on Advanced Robotics*, 2005.

[9] Grefenstette, J. and A. Schultz, "An Evolutionary Approach to Learning in Robots", *Machine Learning Workshop on Robot Learning*, 1994.

[10] Calinon, S. and A. Billard, "Learning of Gestures by Imitation in a Humanoid Robot", A. Revel and J. Nevel (eds.) *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions,* Cambridge: Cambridge University Press, 2004.

[11] Bekey, G. *Autonomous Robots,* MIT Press, 2005.

[12] Erol, D., J. Park, E. Turkay, K. Kawamura, O.C. Jenkins, and M. Mataric, "Motion Generation for Humanoid Robots with Automatically Derived Behaviors", *Proc. IEEE Int'l Conf. SMC*, 1816-1821, 2003.

[13] Grupen, R.A. and M. Huber, "A Framework for the Development of Robot Behavior", *AAAI Spring Symposium Series: Developmental Robotics*, 2005.

[14] Peters II, R.A., K.A. Hambuchen, K. Kawamura, and D.M. Wilkes, "The Sensory Ego-Sphere as a Short-term Memory for Humanoids", *Proc of the IEEE-RAS Int'l Conf. on Humanoid Robotics*, 451-459, 2001.

[15] Achim, K., *Image Mapping and Visual Attention on a Sensory Ego-Sphere,* Master's Thesis, Nashville: Vanderbilt University, August 2005.

[16] Rose, C., M.F. Cohen, and B. Bodenheimer, "Verbs and Adverbs: Multidimensional motion interpolation". *IEEE Computer Graphics & Applications*, 18(5), 32-40, 1998.

[17] Kawamura, K., R.A. Peters II, R. Bodenheimer, N. Sarkar, J. Park, A. Spratley, and K.A. Hambuchen, "Multiagent-based Cognitive Robot Architecture and its Realization", *Int'l Jo. of Humanoid Robotics*, 1(1), 65-93, 2004.