International Journal of Humanoid Robotics
Vol. 1, No. 1 (2004) 65–93
© World Scientific Publishing Company

A PARALLEL DISTRIBUTED COGNITIVE CONTROL SYSTEM FOR A HUMANOID ROBOT

KAZUHIKO KAWAMURA, R. ALAN PETERS II, ROBERT E. BODENHEIMER, NILANJAN SARKAR, JUYI PARK, CHARLES A. CLIFTON, ALBERT W. SPRATLEY and KIMBERLY A. HAMBUCHEN

Center for Intelligent Systems, Vanderbilt University
VU Box 350131 Station B, Nashville, TN 37235-0131, USA

Received 15 August 2003
Accepted 26 November 2003

During the last decade, researchers at Vanderbilt have been developing a humanoid robot called the Intelligent Soft Arm Control (ISAC). This paper describes ISAC in terms of its software components and with respect to the design philosophy that has evolved over the course of its development. Central to the control system is a parallel, distributed software architecture, comprising a set of independent software objects or agents that execute as needed on standard PCs linked via Ethernet. Fundamental to the design philosophy is the direct physical interaction of the robot with people. Initially, this philosophy guided application development. Yet over time it became apparent that such interaction may be necessary for the acquisition of intelligent behaviors by an agent in a human-centered environment. Concurrent to that evolution was a shift from a programmer's high-level specification of action toward the robot's own motion acquisition of primitive behaviors through sensory-motor coordination (SMC) and task learning through cognitive control and working memory. Described is the parallel distributed cognitive control architecture and the advantages and limitations that have guided its development. Primary structures for sensing, memory, and cognition are described. Motion learning through teleoperation and fault diagnosis through system health monitoring are also covered. The generality of the control system is discussed in terms of its applicability to physically heterogeneous robots and multi-robot systems.

Keywords: Multi-agent based robot control architecture; cognitive control; biologically inspired memory structures; automatic motion generation; task learning; system health monitoring.

1. Introduction

The Center for Intelligent Systems (CIS) at Vanderbilt University was established in 1985 to advance the state of the art in intelligent systems through research and development of robots that interact with people. The current research emphasis is on humanoid robots, in particular task acquisition through the learning of behaviors and their sequences, sensory association and attention, cognitive control, and short- and long-term and working memory structures. Application-oriented research

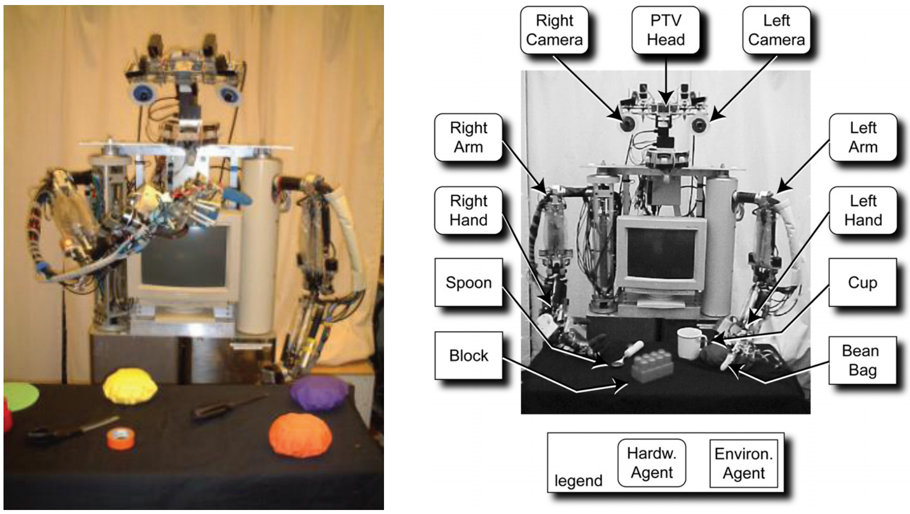


Fig. 1. ISAC humanoid robot and examples of atomic agents for ISAC.

includes collaborative teams of people and robots, heterogeneous robot teams, and tools and technologies for human-robot interface.[1]

The humanoid robot, ISAC (Intelligent Soft Arm Control) (Fig. 1) was designed initially to assist the physically disabled during meals.[2] Many of the problems solved for that purpose had broader applicability so the robot gradually became a more general-purpose test bed for human-robot interaction (HRI) research.[3] Such high-level interaction exposed many of the lower-level deficiencies in the architecture (and robot programming in general) which has led to more recent research in the sensory motor coordination basis for intelligent behavior[4] and biologically inspired memory structures. ISAC is equipped with pneumatic actuators called McKibben Artificial Muscles[5,6] and sensors including stereo CCD cameras, microphones, and infrared sensors (see appendix). Unlike humanoid research groups that emphasize human-like motion control (such as walk pattern generation), the Vanderbilt group's emphasis on HRI has led them to study the development of behavior learning and other cognitive aspects of the humanoid robot.

This paper describes the various components of the ISAC system starting with the parallel distributed, multi-agent software framework called Intelligent Machine Architecture (IMA).[7,8] IMA promotes code reuse and software integration. But it also has scalability problems which are discussed. This is followed by a description of the control system built on IMA and its primary constituents. Each of these is described in turn. That includes the low- and high-level software objects that the researchers call "agents" and the primary memory structures, short-term and long-term. Task learning through the acquisition of behavior sequences is described as is the combination of behaviors through interpolation. Another strategy for task learning through working memory is briefly touched upon. How ISAC will detect servo-motor level failures is the topic of the last section. Several experiments are also presented to illustrate how various components work together to complete tasks.

2. Intelligent Machine Architecture

A humanoid robot is an example of a machine that requires intelligent behavior to act with generality in its environment. Especially in interactions with humans, the robot must be able to adapt its behaviors to accomplish goals safely. As the complexity of interaction grows, so grows the complexity of the software necessary to process sensory information and to control action purposefully. The development and maintenance of complex or large-scale software systems can benefit from domain-specific guidelines that promote code reuse and integration. Intelligent Machine Architecture (IMA) was designed to provide such guidelines in the domain of robot control.[9,10] It is currently used to control ISAC and a set of mobile robots.[11,12]

IMA consists of a set of design criteria and software tools for Windows NT/2000 that supports the development of software objects that we call "agents." An *agent* is designed to encapsulate all aspects of a single element (logical or physical) of a robot control system. A single hardware component, computational task, or data set is represented by an agent if that resource is to be shared or if access to the resource requires arbitration. Agents communicate through message passing using DCOM, the Distributed Component Object Model service of Windows NT/2000. IMA facilitates coarse-grained parallel processing because of the loose coupling afforded by message passing and because DCOM allows software objects on separate computers to be treated as if they were local to each other. Each agent acts locally based on its internal state and provides a set of services to other agents through various relationships. The resulting asynchronous, parallel operation of decision-making agents simplifies the system model at a high level. IMA has sufficient generality to permit the simultaneous deployment of multiple control architectures. A behavior can be designed using any control strategy that most simplifies its implementation. For example, a simple pick and place operation may be most easily implemented using a standard Sense-Plan-Act approach, whereas visual saccade is more suited to subsumption, and object avoidance to motion schema.

There is a two-level hierarchy of IMA agents comprising *atomic agents* and *compound agents*. A compound agent contains or depends on other agents for its primary function. An atomic agent encapsulates a single resource; it neither subsumes nor references any other agent. (The term "atomic" is used in the sense of the Greek word *atomos*, literally "indivisible.") Each controllable hardware device or common data resource on the robot has an associated *hardware/resource* atomic agent. A dataset or computational procedure associated with a specific object in the robot's external environment has an associated *environment* atomic agent.

Within IMA, any existing agent can be accessed by any other agent. Their connectivity, as defined by message passing, is flat — without hierarchy. However, a virtual or logical hierarchy is implied by the structure of a compound agent. Various levels of abstraction form within the control system as needed but are not fixed. Two compound agents, the Self Agent and the Human Agent, are discussed in Sec. 3.

IMA works very well to promote software reuse and dynamic reconfiguration. However, the large systems built with it have experienced scalability problems on two fronts. First, as the system exceeds a certain level of complexity it is difficult for any programmer to predict the interactions that could occur between agents during actual operation. This level seems to be higher than for a direct, sequential program. But that level has been reached in the development of ISAC. The other scalability problem may or may not be a problem with IMA itself but may be an inevitable consequence of increasing complexity in a system based on message passing. The asynchronous nature of message passing over communications channels with finite bandwidth leads to system "lock-ups." These occur with a frequency that apparently depends on the number of agents in the system. It may be possible to minimize this problem through the use of system-self monitoring or through a process of automatic macro-formation. For example, the system could, through a statistical analysis, recognize the logical hierarchies of agents that form repeatedly within certain tasks or under certain environmental conditions. A structure so discerned could be used to "spin off" copies of the participating agents. These could be encapsulated into a macro, a compound agent that optimizes the execution and inter-process communications of the agents involved. For such an approach to be most useful, it should be automatic and subject to modification over time frames that encompass several executions of a macro.

The way in which IMA was used initially in the ISAC system caused problems with real-time control. It seems obvious in retrospect, but asynchronous message passing within a real-time control loop works only if the latency due to computation and communication is less than the sampling rate of the control loop. In a relatively simple task, this was not a problem. But it was as the system increased in complexity. The separate encapsulation of a multi-agent task (such as visual servoing) was found to be necessary if that task was but a small component of a larger control problem (such as responding to a person's speech while performing object recognition on-the-fly during a reach to grasp an object). The servo control loops for arm control were especially susceptible. Thus QNX, a real-time operating system, was employed to execute the arm-control agents directly on the computer that contained the arm hardware interface cards.[13]

To realize distributed object computing, agents in QNX are built using the ACE ORB, an Object Request Broker (ORB) for CORBA (Common ORB Architecture) that builds on the Adaptive Communications Environment (ACE). It is described by its developers as follows: "The ACE ORB (TAO) ... is our standards-based, CORBA middleware framework that allows clients to invoke operations on distributed objects without concern for object location, programming language, OS platform, communication protocols and interconnects, and hardware."[13] Within IMA on QNX, the agents are the TAO clients. Since most other IMA agents run on Windows platforms with DCOM (which performs the same function for windows as CORBA does for QNX) a DCOM-TAO bridge was built to permit communication between these two types of agents. Behavior generation described in Sec. 5 uses QNX-based robot arm teleoperation.

3. Cognitive Robot Architecture

IMA encapsulates the functions of hardware, low-level controllers, and basic sensory processing into independent, reusable units. This abstraction of details away from control loops, image operators, signal processing algorithms, and the like, enables programming to occur at the level of purposeful actions and environmental features. Actuators are supplanted by actions. Raw sensory data are replaced by features. For example, a programmer can instruct the robot to find the red object and pick it up, rather than grabbing images sequentially from a pair of cameras, transferring that data into an array where it can be segmented according to color, finding blobs in left and right images, computing disparity, then depth, filtering the 3D position estimate, computing the difference between the object location and the end-effecter position, computing a Jacobian then joint angles and velocities, sending those to a motion controller, waiting for the result, measuring the actual position, then grabbing a new pair of images and repeating until the arm is in position and the grasp can be planned and controlled. Each of the individual tasks is encapsulated by an atomic agent and these are subsumed by *find-colored-object*, *reach-to-point*, and *grasp-object* agents. The limits of such abstraction are the robot itself and the objects with which it interacts. Hence at the highest level, the ISAC control system comprises a Self Agent and a set of Object Agents, the most complex of which is the Human Agent.

Figure 2 depicts the key agents and the memory structure within the cognitive control architecture. Sensory processing agents write data to the Sensory EgoSphere (SES) which acts as a short-term memory (STM) and interface to the high-level agents (cf. Sec. 4.1). Object Agents, including the Human Agent, monitor the SES for information relevant to their tasks and write information on the SES that is relevant to the object. The Human Agent recognizes and keeps track of all information relevant to the robot's interactions with a person (cf. Sec. 3.2). The Self Agent subsumes a set of atomic and compound agents that actuate the robot and monitor its internal states. Long-term memory (LTM) stores motor skills in the form of procedural memory (PM) and in the future the knowledge base in the form of declarative memory (DM) (cf. Sec. 4.2). In order to continually bias processing throughout a given task, rapid and frequent access to stored information must be available. It is believed that this is accomplished by a working memory system in humans. Inspired by this, we are currently developing a working memory (WM) model for ISAC (cf. Sec. 4.3).

3.1. *Self Agent*

The Self Agent (SA) is responsible for ISAC's cognitive activities ranging from sensor signal monitoring to planning and cognitive control. Cognitive control[14] is

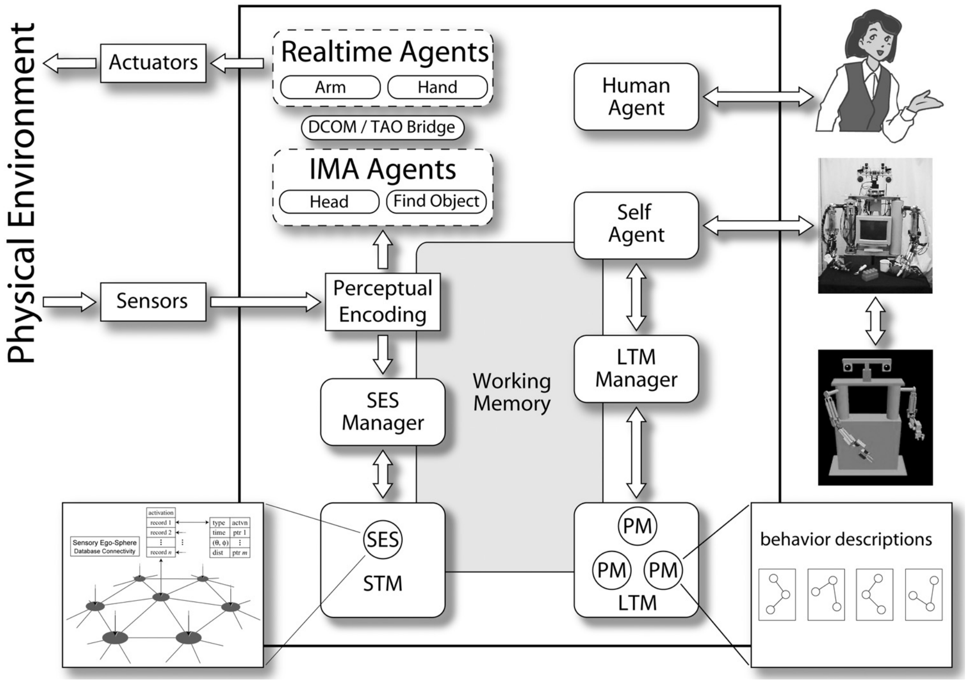


Fig. 2. Key IMA agents and the memory structure.

needed in tasks that require the active maintenance and updating of context representations and relations to guide the flow of information processing and bias actions.[15] Figure 3 is a schematic of the Self Agent and memory structure. The Description Agent provides the description of atomic agents available in the system in terms of what it can or cannot do and what is it doing. The Intention Agent selects the humanoid's actions based on its own state and on the perceived intentions of the person with whom the robot is currently interacting (supplied by the Human Agent). It also determines if this action would conflict with the robot's current activities. If so, the Intention Agent resolves the conflict as a function of the relative priorities of the actions. The Interaction Agent provides speech output to humans and the Pronoun Agent performs keyword matching during a task decomposition process.

The Central Executive Controller is being designed to interact with the Human Agent, the STM and the LTM to construct and invoke plans to perform various tasks. The goal of each generated plan is determined by input from the Intention Agent. Constructed plans are put into action by activating appropriate behaviors to form new procedural memory or motions, guided by the attention networks and sensory information received from the SES.

Another key feature of our cognitive robot might be called "self-reflection." Self reflection will allow the robot to reason its own abilities, cognitive processes, and

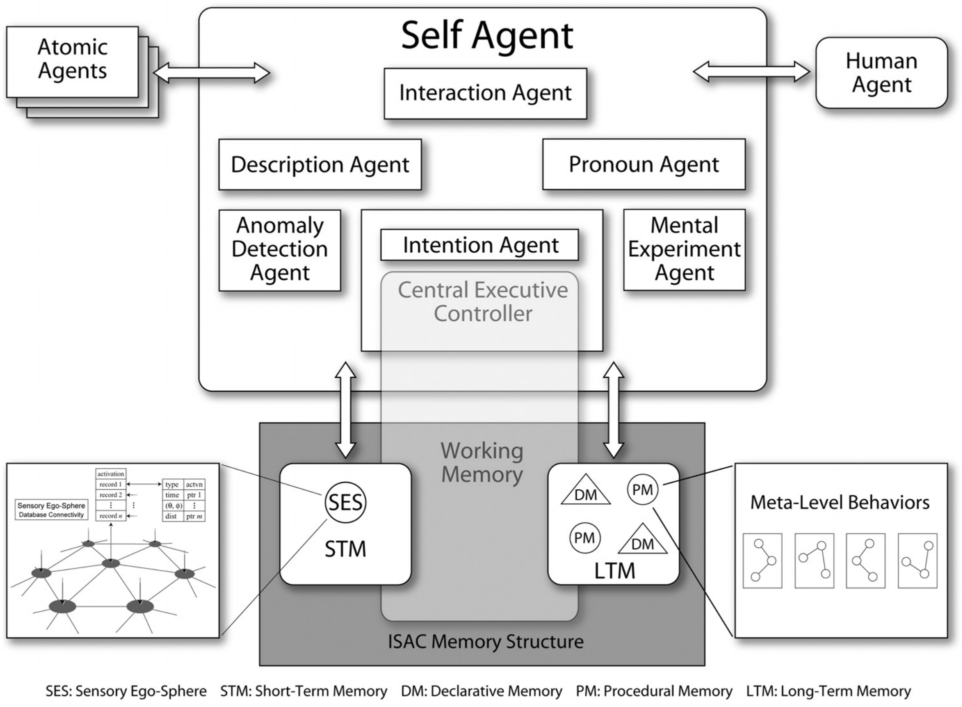


Fig. 3. Self agent and memory structure.

knowledge.[16] As part of an initial effort to incorporate self-reflective process into ISAC, we are incorporating two agents: the Anomaly Detection Agent (ADA) and the Mental Experimentation Agent (MEA) into the Self Agent. The ADA will monitor the inputs and outputs of the atomic agents in the system for fault detection (cf. Sec. 6). The MEA invokes the Central Executive Controller (CEC) to produce an action sequence appropriate for current task conditions and the current condition. When an impasse is raised and if the CEC fails to find an alternative action sequence, the MEA will conduct a search through the space of control parameters to accomplish the task in "simulated mode."

3.2. *Human Agent*

The Human Agent (HA) comprises a set of agents that detect and keep track of human features and estimate the intentions of a person within the current task context. It estimates the current state of people interacting with the robot based on observations and from explicit interactions (Fig. 4).[17] The HA receives input from various atomic agents that detects physical aspects of a human (e.g. the location and identity of a face). The HA receives procedural information about interactions from the SA, which employs a rule set for social interaction. The HA integrates the

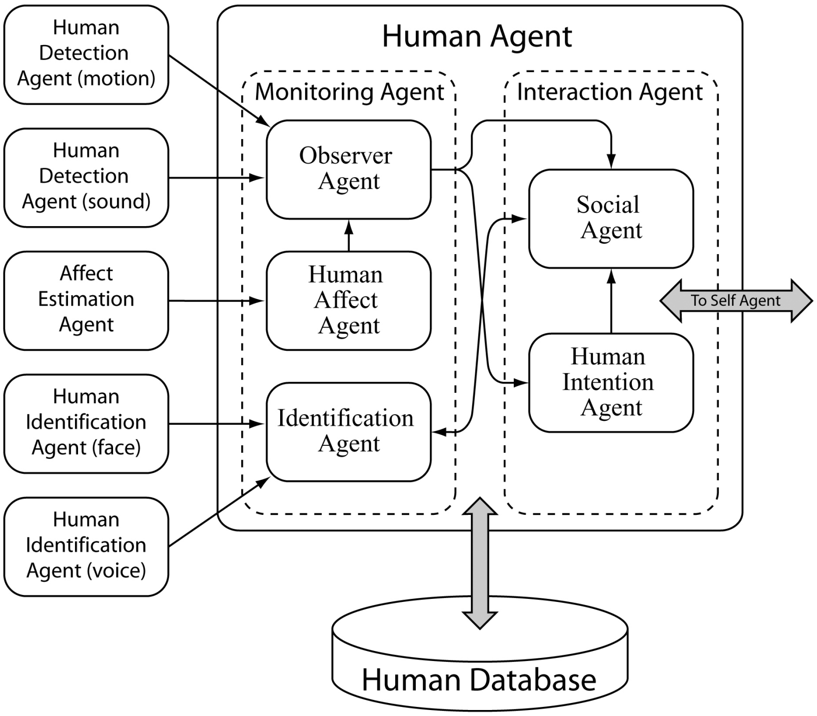


Fig. 4. Human agent and associated atomic agents.

physical and social information with certain inferred aspects of the cognitive states of interacting humans, such as a person's current intention.

The HA processes two types of human intentions. An *expressed intention* is derived from speech directed toward ISAC, e.g. greetings and requests from a human. *Inferred intentions* are derived through reasoning about the actions of a person. For example, if a person leaves the room, ISAC assumes it means that the person no longer intends to interact, therefore, it can reset its internal expectations.

The Human Agent's assessment of how to interact is passed on to the SA. The SA interprets the context of its own current state, e.g. current intention, status, tasks, etc. This processing guides ISAC in the selection of socially appropriate behaviors that lead towards the ultimate goal of completing tasks with (or for) humans.

Figure 5 shows the model of the levels of interaction engagement, represented by a numerical value, that we have developed as the basis for modeling social interaction. These levels progress the robot from a state of no interaction to an ultimate goal of completing a task with (or for) a person. Level 1, Solitude, corresponds to when ISAC does not detect anyone in the environment. In this situation, ISAC may choose actions to actively attract people with whom to interact. Level 2, Awareness of People, corresponds to a stage, often short time, when ISAC is aware of people around it, and has not interacted with them. Level 3, Acknowledgement, is the phase

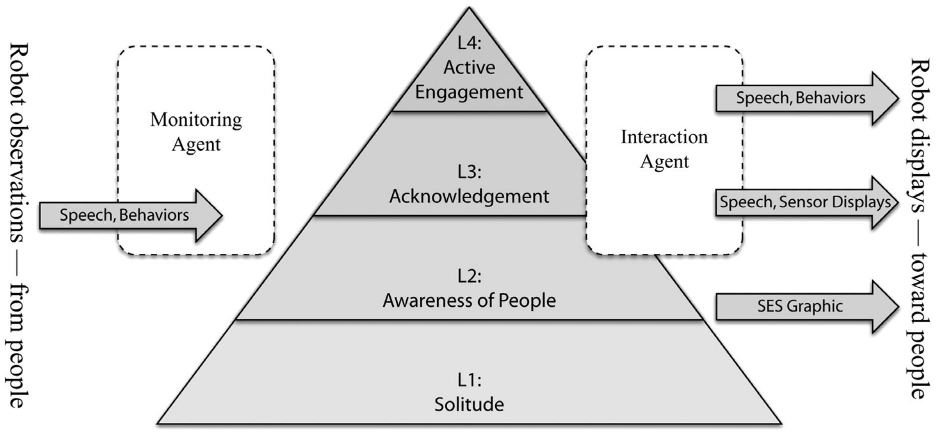


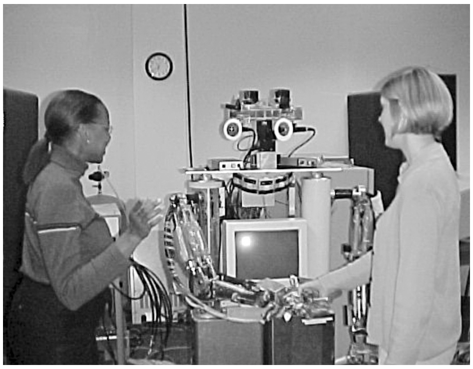
Fig. 5. Levels of interaction within the human agent.

when the robot actively acknowledges the presence of a person. This is performed if the person is approaching ISAC for the first time or if the person is interrupting an ongoing interaction. Level 4, Active Engagement, represents that stage of active interaction.

3.3. *Demonstration in situation-based acknowledgment*

In this demo, ISAC processes the intentions of the human, resolves them with its own intentions and abilities, and communicates to the person if there is a problem with the request. The scenario begins as a person approaches ISAC and gains its attention. The Human Agent determines that the person has an intention to interact with ISAC. If ISAC is unoccupied at the time, ISAC begins its interaction behaviors by turning toward the person and initiating a greeting and identification sequence. Once interaction is established, ISAC begins a social dialog. After greeting, ISAC may respond to a person's task request (an intention for ISAC to do something) if it is within ISAC's abilities. If a second person approaches ISAC and attempts to gain its attention, the Human Agent will notify the Self Agent that there is a new person with a pending intention [Fig. 6(a)]. The Self Agent must then resolve the current human intention with its own current intention. If the second human intention is not of sufficient priority to override ISAC's current task, then ISAC will then pause its current interaction, turn to the interrupter, and apologize for being busy [Fig. 6(b)]. ISAC can then return its attention to the first person and resume its previous interaction. There may also be a situation where the request of the interrupting person actually has higher priority than the task ISAC is currently performing. In this case, the Self Agent determines to switch to the new task after giving an explanation to the current person.

(a)



(b)

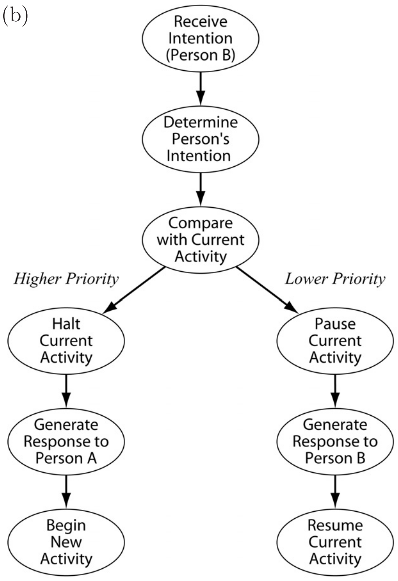

Receive Intention (Person B)
Determine Person's Intention
Compare with Current Activity
Higher Priority
Lower Priority
Halt Current Activity
Pause Current Activity
Generate Response to Person A
Generate Response to Person B
Begin New Activity
Resume Current Activity

Fig. 6. (a) ISAC responding to an interruption; (b) role of intention processing during an interruption.

4. ISAC Memory Structure

Memory in mammals is usually analyzed in two or more categories based on temporal duration. Most often used are the terms *short-term*, *long-term and working memories*. There is however neither agreement on the meaning of these nor consistency in their use. For example, Squire and Kendal[18] define short-term memory (STM) as comprising those memories that last anywhere from several minutes to several weeks that appear to be controlled, in part, by hippocampal function. Long-term memory (LTM) can persist indefinitely and appears to require the involvement of the frontal cortex. Procedural memory (PM), on the other hand, appears to be stored in the putamen.[19] Finally, working memory is used to describe how we juggle perceptions, memories and concepts.[19] For ISAC, we adapted short-term memories to be those that persist, at most, through a session of continuous operations, long-term memories to be those between sessions and working memories to be those needed for executing task-oriented operations and thus could be either short term or long term.

4.1. *Short term memory, coincidence detection, and attention*

ISAC's short-term memory resides in a Sensory EgoSphere (SES).[20] The SES operates asynchronously as a high-level agent in IMA. In addition to memory function, it provides multi-modal sensory association, and attentional processing. For a person interacting with the robot, the SES can be visualized as a spherical shell centered on the robot's base frame. Each point on the shell is a locally connected memory unit with an associated activation vector and a temporal decay. From an internal, computational point of view, the SES is a graph whose edges form a geodesic tessellation of a sphere. Each node of the graph connects to a database in addition to its neighbors. A SES manager agent interacts with other agents to write and read information to the SES.

On the sensory side of the interface, parallel, independent sensory processing modules (SPM) write data to the SES at points on the sphere. Directional sensors write data to the SES at the point in the direction of the data source. These include exteroceptive sensors, e.g. vision, SONAR, LIDAR, IR, and proprioceptive sensors, e.g. joint angles, force, and torque. Non-directional sensors, e.g. power level, write to an additional point included for such data. When an SPM writes to a point on the SES, the data is stored at the node closest to that point. The actual direction, distance (if known), and time (adjusted for the known latency of the SPM) are recorded and the value of an element in an associated activation vector is increased in a neighborhood of the point. (Although the data structure is discrete and of possibly lower resolution than some of the SPMs that write to it, full location resolution is maintained because that information is written along with the rest of the data. The geodesic discretization permits fast searches through the database, indexed by location.) The activation level decays with a time constant that is a function of the data type. Agents that use sensory data may read from the SES or may add activation to points of interest. Object agents can place descriptors on the SES or search for them there. That operation in itself makes the SES useful for people interacting with a robot as it provides an ego-centric representation of the robot's knowledge of the current environment.

As a short-term memory, the SES is useful for maintaining an inventory of objects in the robot's locale for subsequent manipulation or other action. When the robot recognizes an object, the location of a point of reference on the object (part of the object definition) and the object's pose are stored along with an identifier and time stamp at the closest SES node. The identifier is used as a tag by the SES for its search and recall routines. The time stamp can be used along with an activation decay constant to compute a probability that the object is at the recorded location after time has elapsed. As an object moves, its location is updated by the SES so that the robot always stores the object's position relative to the base frame. This position is likely to accrue errors if the robot is not actively tracking the object with its sensors. Therefore, the SES provides the starting location for a sensory search if the object is not found by the sensors at the recorded location upon later recall. The spatial layout of the SES keeps track of the spatial relationships between objects so that the robot can know "what is where" [Fig. 7(a)].

There are two primary hypotheses behind the definition and use of the SES: (i) often, a physical event in the environment will stimulate more than one of the robot's sensors, and (ii) changes in motion of the robot can precipitate a sensor event (a sudden detectable change in the signal, its derivatives, or its statistics). Thus, if two or more of the SPMs detect events at nearly the same time, and if directionally

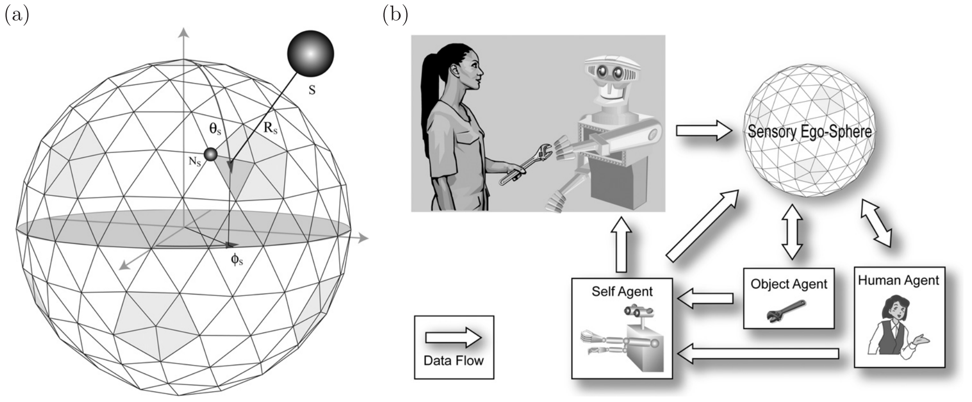


Fig. 7. (a) Sensory EgoSphere and (b) association with high-level agents.

sensitive modules report their events as having emanated from similar directions in space, then we presume that the robot has detected a real event. Moreover, if a change in motion is accompanied by the registration of events by more than one sensor we presume the events may be relevant. By including proprioceptive sensing and motor control sequences with the exteroceptive sensory streams that project to it, the SES makes spatio-temporal sensory-motor data associations. It does so without having to perform any comparative operations on the sensory signals [Fig. 7(b)].

Each node on the SES has either five or six neighbors connected by edges. To enable both coincidence detection and attentional processing, a radial basis function (RBF) is associated with each node. When the SES receives directional data, it adds activation to the node closest to the direction of the data. The RBF spreads the activation to all nodes that are a given number of edges away with an exponentially decreasing intensity. These nodal activation values can be used to direct the attention of the modules that read data from the SES and, thereby, the attention of the robot. The SES can be biased toward the selection of specific data by modulating the strength of activations assigned to SES nodes. This bias is useful for directing the robot's attention during tasks such as picking up objects, or during contextual circumstances such as working with people. The attention network balances the trade-off between contextually important data and unexpected yet salient data. It does this by combining the activation from the nodal RBFs (that represent the salience of events in the environment) with priority values (that represent desired data). The focus of attention is selected as the node that receives the highest combination of activation from both the RBFs and the priority values.

4.2. *Long-term memory*

Long-term memory (LTM) in the human brain stores information such as *motor skills* and *episodic experiences* for future retrieval. In our cognitive robot

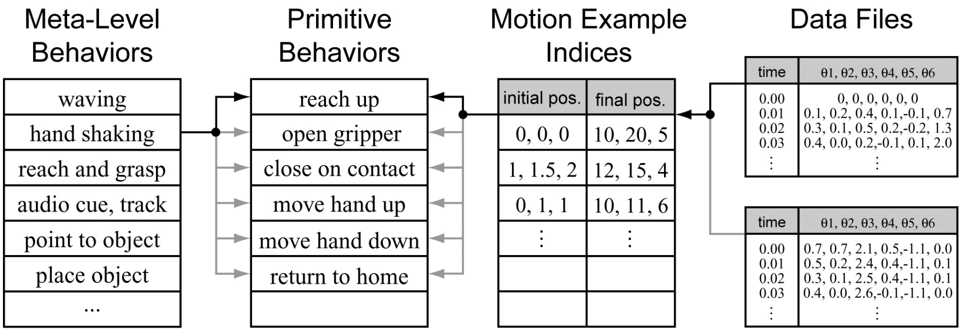


Fig. 8. Structure of long-term memory database.

architecture, LTM is a data structure which contains behaviors that will be combined to perform specific tasks. A motor skill unit of LTM is a Procedural Memory (PM). PMs are derived using the spatio-temporal Isomap method proposed by Jenkins and Mataríć.[21] A short description of how it was used to generate PMs is described in Sec. 5.1.

Behaviors are organized in LTM using pointers to motions stored in the form of trajectories, such as how to *reach to a point*. Motion skills for each behavior must be interpolated in order to be used in specific situations. The interpolation method we are using is the Verbs and Adverbs method developed in Ref. 22. This technique describes a motion (verb) in terms of its parameters (adverbs) which allows ISAC to generate a new movement based on the similarity of stored motions. Adverbs, or parameters, can be objective values which allow each set of motions to be interpolated in very different ways. Thus, ISAC will be able to show new movements based on a limited number of learned movements (cf. Sec. 5.2).

The current LTM structure consists of a set of joint-angle space trajectories indexed by the initial and final values of the motion stream as illustrated in the data files in Fig. 8. Shortcomings of this data representation are that a separate database must exist for every primitive motion or meta-level behavior and there is no mechanism for controlling the underlying motion parameters during interpolation. To avoid these limitations, a more logical description for each behavior is enhanced (Fig. 8). Each entry in the meta-level behavior table contains pointers to underlying primitive motions.

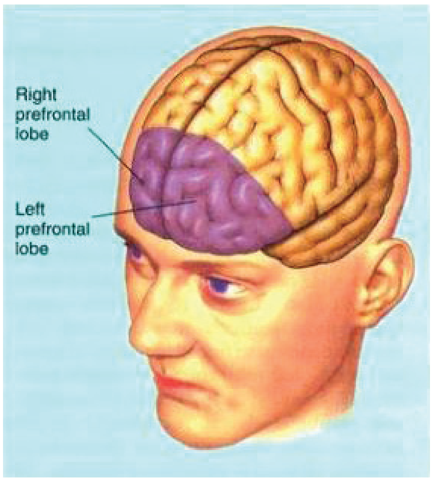
4.3. *Working memory*

Cognitive scientists have long gathered evidence for a variety of memory systems in the mammalian brain.[18] One dichotomy, introduced early in the history of psychology,[23] has been between short-term memory and long-term memory, with the former being more prone to rapid forgetting. More recently, separate short-term mechanisms have been found for visual-spatial information and speech-based information which led to a model of working memory based on three parts: the Central Executive, the Visual-spatial sketch pad and the Phonological loop holding acoustic and speech-based information.[19,24] While contemporary theories of working memory are diverse[25] researchers agree that working memory is a mechanism that protects a small number of informational "chunk" from interference and distraction and places them in a position to directly influence behavior.[26] This implies that the size of working memory should be *adaptive* to the task.

Neurological studies point to the prefrontal cortex region (PFC) as a likely candidate for supporting these functions performed by working memory[27,28] [Fig. 9(a)]. In the PFC, mental representations are protected from interference and/or frequently updated to bias processing in task completion. Irrelevant information must not be allowed to interfere with task execution. However, the system must be flexible enough to allow for the learning of new information. Working memory allows ISAC to focus its attention on the salient task features and adapt to learning new experience-based behaviors.

Inspired by this, we are currently developing an adaptive Working Memory (WM) system in our humanoid robot.[29] Of particular interest is the process by which the importance and sensitivity of certain information is modified based on the task success or failure. For effective task completion, the sensitivity to the overall system's inputs should be managed in such a way that both rapid updating and robust maintenance are facilitated.

A system utilizing a "gate" that can be opened to incorporate new data, or closed to limit the effect of distracting information would accomplish this. In humans, it is believed that gating is done by modulating dopamine (DA) levels. DA gating plays a role in reward based learning by allowing synaptic connection

(a)


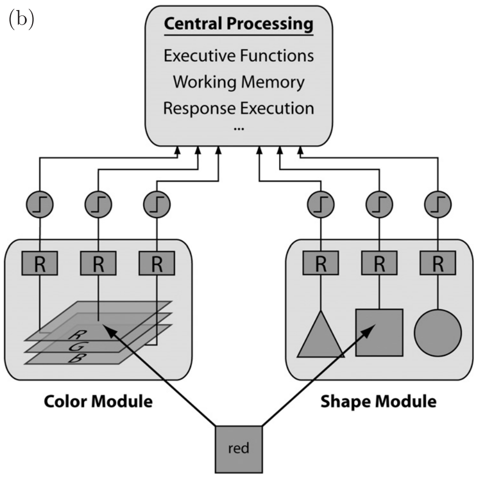
(b)


Fig. 9. (a) Prefrontal Cortex Region (PFC) region[31] and (b) schematic diagram of the visual DA model; mechanisms in this example detect shape or color to drive the desired visual response.[32]

strengths to be activated or deactivated when necessary.[30] By opening the gate (i.e. increasing DA levels), PFC representations are susceptible to change, and by closing the gate, the representations are protected from interference [Fig. 9(b)].

We can model the gating action in our system by applying a reinforcement learning algorithm, called Temporal Difference (TD) learning.[33] This uses the expectation of reward from a previous experience to determine the decision that will maximize reward for the current action. Over time, changes in the reward estimate will enhance future predictions and improve action selection. The structure the adaptive working memory and its integration into robots to provide the embodiment necessary for exploring the issue of task learning is the focus of the four-year NSF grant.[29]

5. Motion Generation and Behavior Derivation

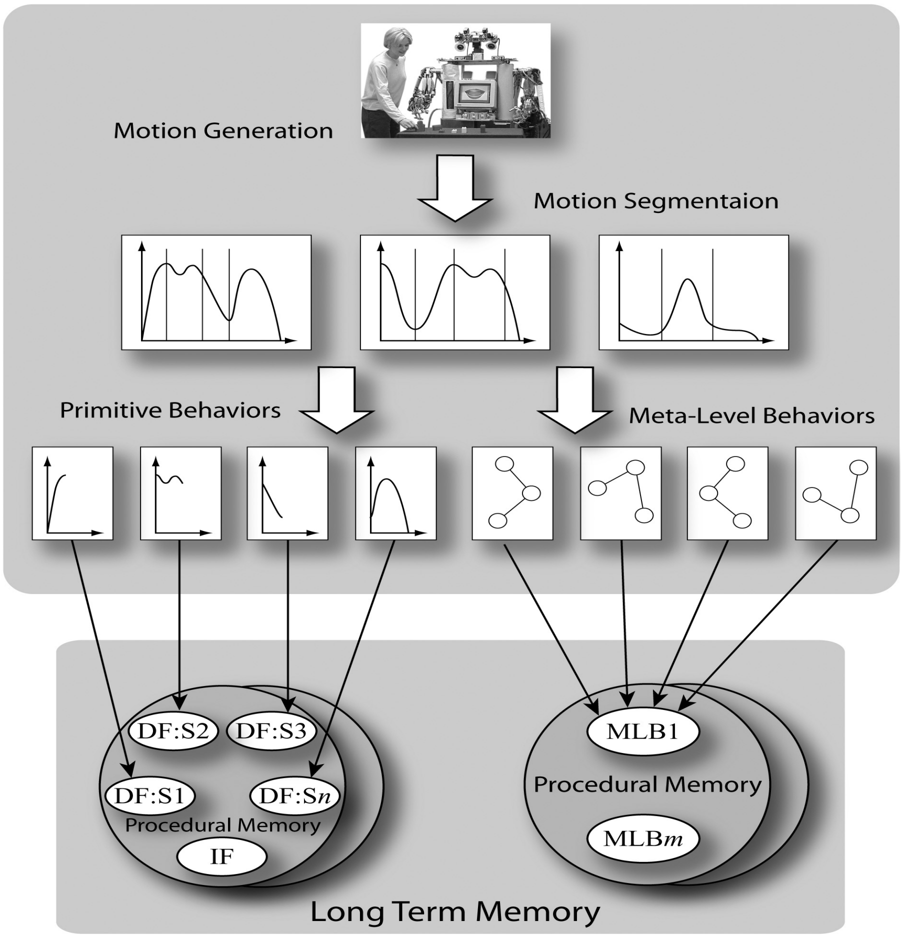
It is well known that sensory-motor coordination (SMC) can be used by a mobile robot to categorize features of its environment and to navigate within that environment.[4] We hypothesize that a similar approach could be used to generate motions and behaviors for a humanoid robot. Specifically, we are testing the following two approaches for automatic motion generation: the spatio-temporal Isomap developed by Jenkins and Matarić[21] and a multidimensional motion interpolation method called the Verbs and Adverbs developed by Rose, Cohen and Bodenheimer.[22]

5.1. *Spatio-temporal Isomap*

This motion derivation method consists of four main components. The derivation system takes as input a single continuous kinematic motion as a time-series of joint angle values. This motion is segmented into intervals based on some heuristic defining separating SMC events, with each segment assumed to be an atomic motion. The result from the derivation process is a behavior vocabulary consisting of primitive behaviors, which represent a family of kinematic motion across a span of variations, and meta-level behaviors, which represent sequential combinations of the primitives and index into them to produce action.

In our approach, the derived vocabulary is assumed to be an intrinsic substrate of basic robot skills. Consequently, this vocabulary is stored as long-term memory, more specifically as Procedural Memory (PM). Generally, PM is a memory unit for storing a skill and procedure, and is involved in tasks such as remembering how to reach to a point. As shown in the data file in Fig. 8, each primitive behavior is stored a set of trajectories in joint angle space with an indexing structure stored as a PM unit. This indexing structure stores the initial and final Cartesian coordinates for all arm trajectories in a primitive behavior.[34]

Motion data is collected from the teleoperation of ISAC and then segmented. The central idea in the derivation of behaviors from motion segments is to discover spatio-temporal structure in a motion stream as shown in Fig. 10.



DF:S*n* — Data File for Segment *n*
IF — Index File
MLB*m* — Meta-Level Behavior *m*

Fig. 10. Derivation of PM through human-guided motion stream.

5.2. *Verbs and adverbs*

We are incorporating recent developments from the field of computer animation into our behavioral control and motion generation. The particular technique we are using is called "Verbs and Adverbs"[21] which assumes that fundamental motion exemplars can be generated by given control strategies that have reasonable robustness properties. The technique then extends the range of these example motions into a continuous behavior by interpolating the robot's degrees of freedom in a user-defined *perceptual* space. The result is a set of behaviors (verbs) that are parameterized by user-defined, continuous controls (adverbs). The advantage of this technique is that example motions sparsely populate the control space.

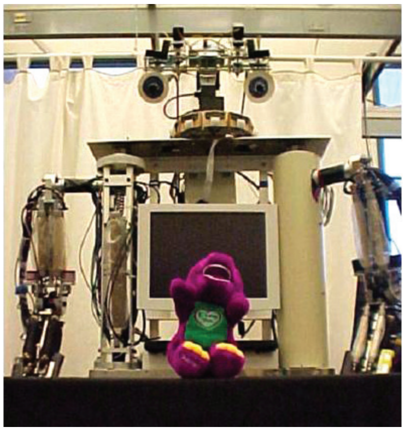
The technique consists of two components, dynamically time-warping the example motions and interpolating the underlying control signals. The motions are first manually segmented into segments bounded by shared points of structural significance, called key-times. Once these key-times have been identified, they are used to time warp the motions and their underlying controls so that all example motions occur on a canonical time-index. Also at this stage, the position in the perceptual space of the example motions is identified. New motions are now created by interpolating the controls used to generate the example motions using a weighted radial-basis function interpolation scheme augmented with a low-order polynomial. Radial basis functions are used because they are computationally efficient and the effect of any example motion is local (and determined by the width of the radial basis functions). The low-order polynomial finds the general trend of the data, and provides limited ability to extrapolate outside of the workspace defined by the example motions. Inverse time-warping is now applied to transform the synthetic motion from its canonical time-frame into normal time.

The constraints of this method are that example motions must be structurally similar and have the same number of key-times in each. Additionally, the example motions must be manually segmented and manually positioned in the perceptual space. This constraint could possibly be relaxed using the work of Jenkins *et al.*[21] but the manual techniques do provide the ability to have fine-grained control over the process. This fine-grained approach is important since the success of the method depends on the operating envelopes of the underlying control strategies being sufficiently robust that the radial basis function blending does not exceed their limits at any point in the perceptual space. Finally, the time-warping must be done with care. Since the underlying motion data is discrete, resampling the motion must be done so that the inverse time-warping produces meaningful control inputs to the robot.

To demonstrate this method on ISAC, a series of reaching and grasping motions were performed under manual control and recorded. Three trials of a reaching motion were recorded to three different collinear locations on a table. Each trial positioned the wrist degree of freedom in a different end position $0°, 90°$, and $180°$. Once the sample trajectories had been recorded, the motion streams were segmented using kinematic centroid, which calculates the Euclidean distance from the shoulder to the center of mass of the entire arm as it moves through the motion stream. The segments are identified as the critical points of the computed function. This technique has the additional advantage of serving as a low pass filter to minimize sensor noise. The example motions were then used in the interpolation method.

The method was tested by asking ISAC to reach out and grab a *Barney* doll [Fig. 11(a)]. The adverb values are the position and orientation of Barney. The verbs and adverbs algorithm produce a trajectory to the correct position and the

(a)



(b)

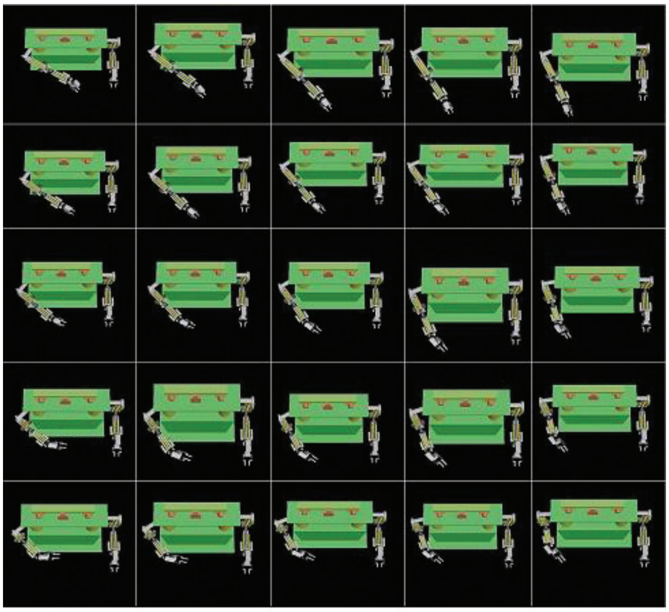


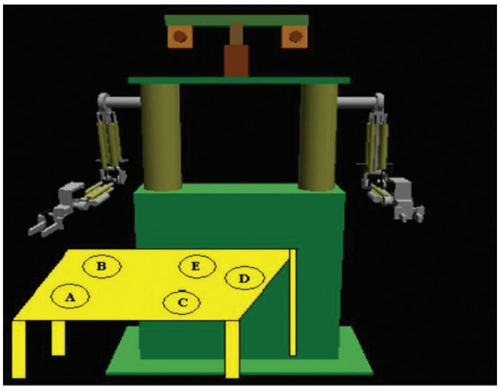
Fig. 11. (a) ISAC preparing to grasp *Barney* and (b) ISAC performing a reaching motion using the verbs and adverbs technique.

proper wrist orientation in order to grasp Barney from any position collinear with the original motions. This was found to be reasonably robust to deviations from collinearity [Fig. 11(b)].

5.3. *Human-guided motion generation and behavior derivation*

This experiment illustrates how the Self Agent, the STM and the LTM work together to generate a new behavior for ISAC under the following scenario: ISAC is told by

(a)



(b)

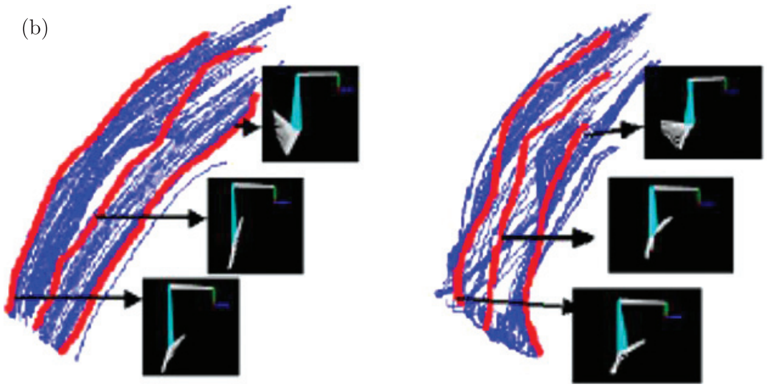


Fig. 12. (a) ISAC is taught to reach five points on a table and (b) results from interpolating selected action units. Each plot shows trajectories in Cartesian coordinates: (Left) reaching; (Right) returning to home.[34]

a human to reach to an object in an unlearned position on a table. First, ISAC is driven by a human to reach five random points (A, B, C, D, E) on a table [Fig. 12(a)], giving a motion stream consisting of five different reaching motions [Fig. 12(b)]. When applying the spatio-temporal Isomap to the motion stream, a meta-"reach-to" behavior is derived and stored as a PM in LTM.

The demonstration illustrates several levels of the cognitive robot architecture for directing attention to known and unknown objects, recalling generic "reach-to" behavior from LTM and using a stereo vision system and the spatial-temporal Isomap to execute the command. More specifically, upon receiving a speech cue and a finger-pointing gesture from the human, the robot's attention is directed to an object. The Human Finger Agent finds a pointed finger to fixate on the object (Fig. 13), and the SES returns the coordinates of the object. Based on this information, the Self Agent retrieves the motion data to execute the reaching motion as illustrated in Fig. 14.

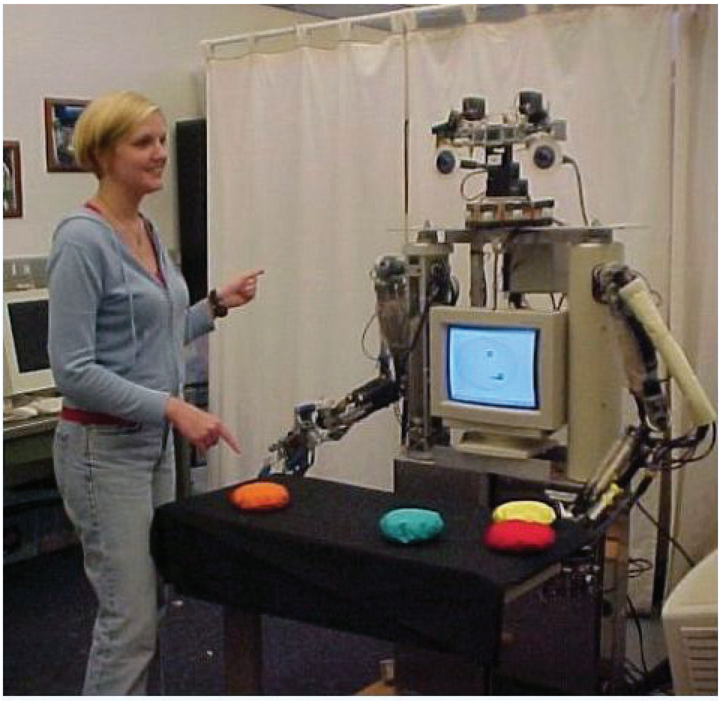


Fig. 13. Finger-pointing demo.

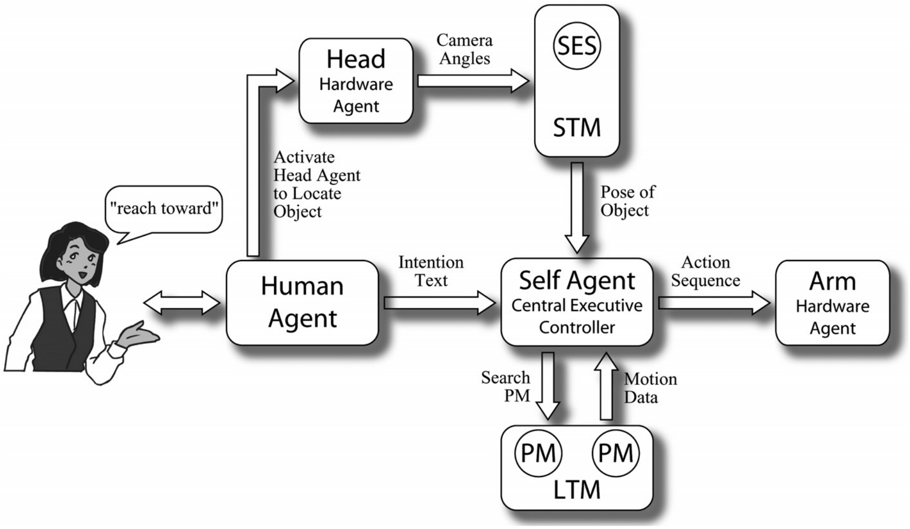


Fig. 14. Data flow for the human-guided motion generation and behavior derivation demo.

6. System Health Monitoring

Like most humanoid robots, ISAC is a complex mechatronic system comprised of many sensors and actuators. The overall task performance of such a system depends on the proper functioning of its components. Despite the best design and maintenance practices, it is unlikely that such a complex system will be immune to system faults. Any fault that may develop in the system components may adversely affect the humanoid. Therefore it is necessary to detect, isolate, and if possible accommodate these faults as soon as they develop. In order to perform the above-mentioned task, we designed an automated monitoring system for ISAC that is expected to provide a real-time status of its sensors and actuators, and an analysis of the possible task failures given a set of system faults. We call it the System Health Monitoring (SHM) module.

ISAC's SHM is designed in a hierarchical manner consisting of three different levels. The lowest level is the component fault detection. Task monitoring is the second level, which utilizes the knowledge of component fault detection to generate a possible task failure scenario given a set of component faults and/or human-robot communication. The highest level, called intention and decision monitoring, is expected to provide high-level information to the human about the humanoid's intention and the associated possible decisions (Fig. 15).

The fundamental component on which the SHM relies is its ability to monitor signals to detect faults. There are several methods that can be used to detect component faults. The main idea is to compare the estimated signal value with the real signal from sensors and actuators to detect faults when there is sufficient deviation (Fig. 16). Several estimator based fault detection techniques have been developed in

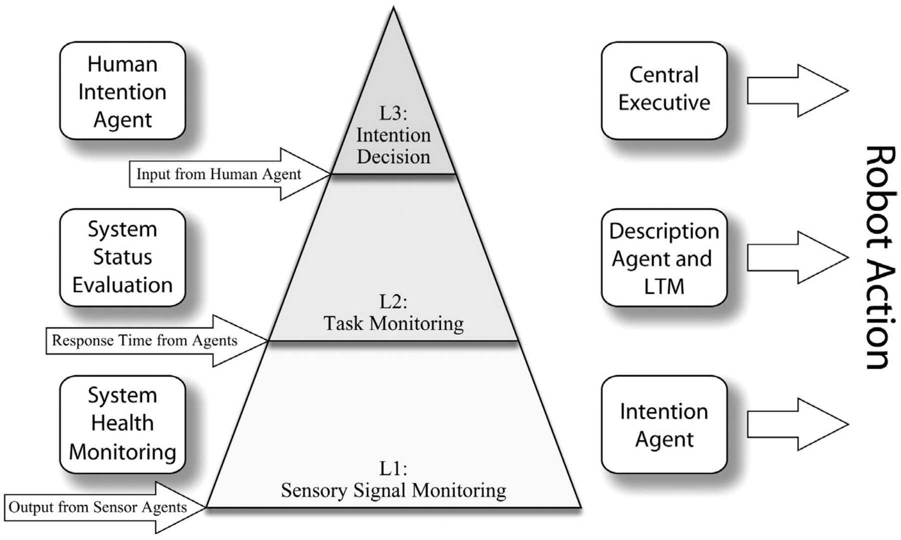
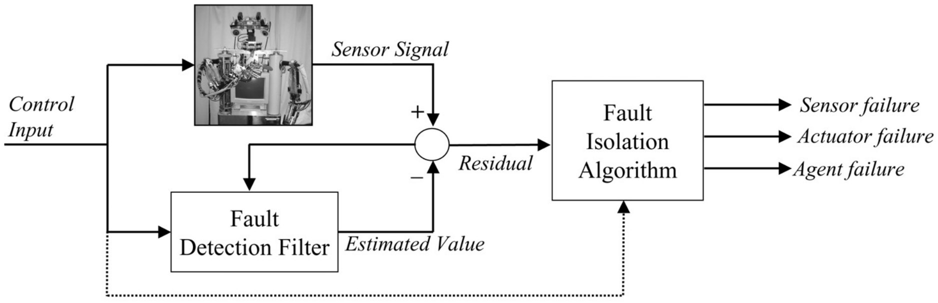
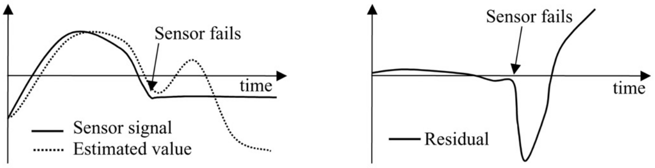


Fig. 15. Levels of monitoring and action in the self agent.



(a) Generic fault detection and identification scheme



(b) Typical plots of sensor signal and residual

Fig. 16. System block diagram and plots of sensor signals.

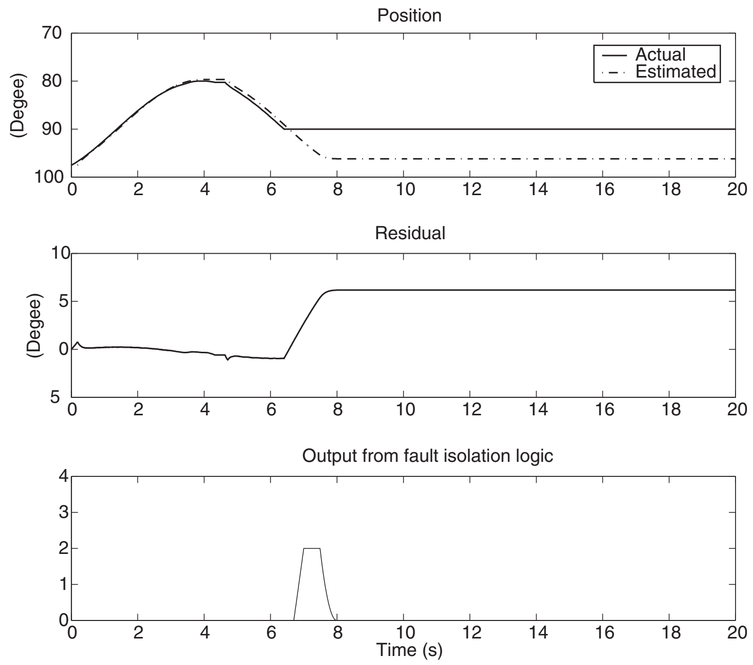
the literature. For example, Hanlon and Maybeck used a residual correlation Kalman filter bank to detect sensor fault in multiple mode of an aircraft.[35] Roumeliotis *et al.* employed a multiple model adaptive estimation (MMAE) technique to detect sensor faults in a mobile robot.[36] Both research groups detected sensor failure by comparing the sensed value with the estimated value from the bank of the Kalman filter. Scattolini and Cattane used Beard–Jones' fault detection filter to detect sensor faults in a large space structure.[37] We are using adaptive Kalman filtering and Beard–Jones techniques for fault detection in ISAC.

In order to identify the fault source, we are developing fuzzy rules based on observation of the error characteristics. We have considered faults in position sensors, actuators, during transmission and collision. As inputs to the fault isolation logic, we are using desired, estimated and measured joint position and measured joint torque. Simulation was conducted to verify the performance of the fault detection and isolation scheme to be used for task monitoring. Results are shown in Fig. 17.

7. Future Direction

Through our research collaboration with NASA-JSC's Robonaut group, USC, UMass and MIT, we plan to pursue a realization of cognitive robots through sensory motor coordination as the basis for generating intelligent behavior. As we gained experience with our cognitive robot architecture, it became

(a)



(b)

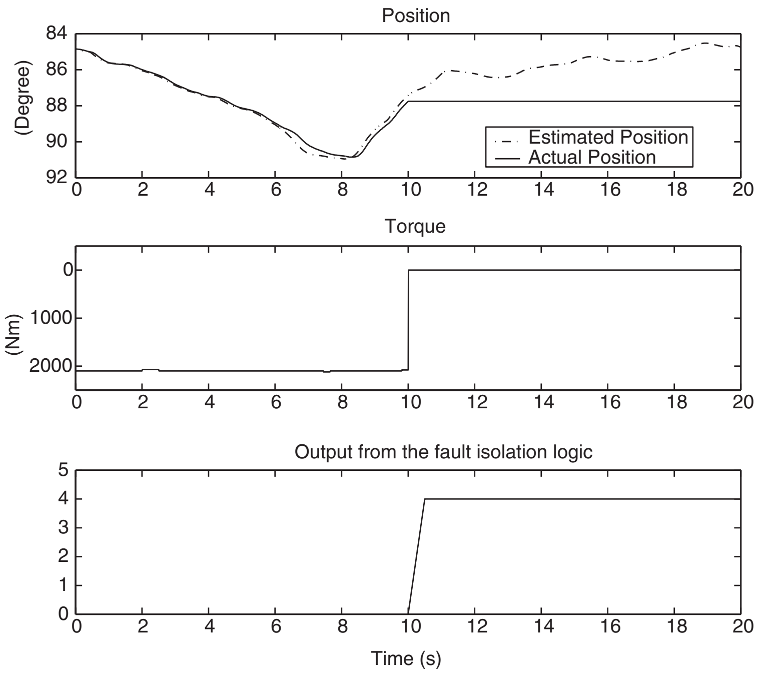


Fig. 17. Simulation results for fault detection and isolation.

clear that the initial memory structure involving short- and long-term memories may not be adequate as we increase tasks for ISAC both quantitatively (i.e. total number) and qualitatively (i.e. complexity). We continue to explore the realization of a third memory structure tentatively named an adaptive working memory, such as the lateral portion of human's prefrontal cortex (PFC) within our cognitive architecture. Our goal for such a PFC model is to allow ISAC to focus attention on the most relevant features of the current task without explicitly programming ISAC. This will finally complete our multiagent-based cognitive robot architecture development initiated more than a decade ago.

Acknowledgments

This research was partially supported under the following grants: DARPA MARS through SMDC (DASG60-99-1-0005), DARPA MARS2020 through NASA-JSC (NAG9-1446), DARPA-SMDC grant (DASG60-01-1-0001), and also through in-house sponsored research. We would like to acknowledge discussions with David Noelle and Kim Hambuchen in the CIS, with Chad Jenkins and Maja Matarić at the University of Southern California and with Rob Ambrose and Ken Alder at NASA-JSC that contributed to this work. The authors would also like to thank colleagues and students, past and present, in the Center for Intelligent Systems at Vanderbilt University for their contributions.

Appendix

Interaction Competencies for a Personal Robot	Hardware	Software
Observational		
Presence of Person: Infrared (IR)	Passive IR motion detector array	Digital I/O
Sound: Event Localization	Condenser Microphones	Matlab
Speech: Detection/Recognition	Handheld	Microsoft Speech Recognition Engine 4.0
Vision: Face and Gesture	Sony Color CCD Cameras	Visual C++ routines, some with Intel Libraries
Demonstrative/Responsive		
Speech: Synthesis	PC Speakers	AT&T Natural Voices Engine
Motor behaviors: Head	Directed Perception PTU-46-70	IMA wrapper for serial port
Motor behaviors: Arms	Pneumatic Muscles	Visual C++ routines and control

References

1. K. Kawamura, T. E. Rogers, K. A. Hambuchen and D. Erol, Towards a human-robot symbiotic system, in *Robotics and Computer Integrated Manufacturing* (2003), pp. 555–565.
2. K. Kawamura, S. Bagchi, M. Iskarous, M. Bishay and R. A. Peters II, Intelligent robotic systems in service of the disabled, *IEEE Trans. Rehabilitation Engineering* **3**(1), 14–21 (1995).
3. K. Kawamura, R. A. Peters II, D. M. Wilkes, W. A. Alford and T. E. Rogers, ISAC: Foundations in human-humanoid interaction, *IEEE Intelligent Systems and their Applications* **15**(4), 38–45 (2000).
4. R. Pfeifer and C. Scheier, Sensory motor coordination: The metaphor and beyond, *Robotics and Autonomous Systems* (*Special Issue on Practice and Future of Autonomous Systems*) **20**(2–4), 157–178 (1997).
5. H. F. Schulte Jr., The characteristics of the McKibben artificial muscle, in *The Application of External Power in Prosthesis and Orthodics*, National Academy of Sciences — National Research Council, Washington, DC, 1961.
6. J. Schröder, K. Kawamura, D. Erol and R. Dillman, Dynamic pneumatic actuator model for a model-based torque controller, in *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation* (July 2003), pp. 342–347.
7. R. T. Pack, D. M. Wilkes and K. Kawamura, A software architecture for integrated service robot development, in *Proc. IEEE Systems, Man and Cybernetics* (1997), pp. 3774–3779.
8. R. Olivares, Intelligent machine architecture 1.0: Introduction and system overview, *Intelligent Robotics Laboratory*, CIS Technical Report, Vanderbilt University, May 2003.
9. K. Kawamura, R. T. Pack, M. Bishay and M. Iskarous, Design philosophy for service robots, in *Robotics and Autonomous Systems*, *International Workshop on Biorobotics: Human-Robot Symbiosis*, eds. K. Kawamura and T. E. Davis (Elsevier, 1986), pp. 109–116.
10. R. T. Pack, IMA: The intelligent machine architecture, Ph.D. Dissertation, Vanderbilt University, Nashville, TN, May 1998.
11. A. B. Koku, Egocentric navigation and its applications, Ph.D. Dissertation, Vanderbilt University, Nashville, TN, May 2003.
12. K. Kawamura, A. B. Koku, D. M. Wilkes, R. A. Peters II and A. Sekmen, Toward egocentric navigation, *Int. J. Robotics and Automation* **17**(4), 135–145 (2002).
13. D. C. Schmidt, D. L. Levine and S. Mungee, The design and performance of real-time object request brokers, *Computer Communications* **21**(4), 294–324 (1998) (http://www.cs.wustl.edu/∼schmidt/TAO-intro.html).
14. T. S. Braver and D. M. Barch, A theory of cognitive control, aging cognition, and neuromodulation, *Neuroscience and Behavioral Reviews* **26**, 809–817 (2002).
15. K. Kawamura, C. A. Clifton, K. A. Hambuchen and P. Ratanaswasd, MultiAgent-based cognitive robot architecture and its realization: Towards body-mind integration, submitted to *IEEE Int. Conf. Robotics and Automation*, New Orleans, 26 April–1 May 2004.
16. K. Kawamura, D. C. Noelle, K. A. Hambuchen, T. E. Rogers and E. Turkay, A multi-agent approach to self-reflection for cognitive robotics, *Int. Conf. Advanced Robotics*, Coimbra, Portugal, 30 June–3 July 2003, pp. 568–575.
17. T. E. Rogers, The human agent: A model for human-robot interaction, Ph.D. Dissertation, Vanderbilt University, Nashville, TN, August 2003.

18. L. R. Squire and E. R. Kandel, Memory: From minds to molecules, *Scientific American Library* **69** (HPHLP, New York, NY, 1999).
19. R. Carter, *Mapping the Mind* (University of California Press, 2000).
20. R. A. Peters II, K. A. Hambuchen, K. Kawamura and D. M. Wilkes, The sensory egosphere as a short-term memory for humanoids, in *Proc. 2nd IEEE-RAS Int. Conf. Humanoid Robots* (2001), pp. 451–459.
21. O. C. Jenkins and M. J. Matarić, Automated derivation of behavior vocabularies for autonomous humanoid motion, in *Proc. 2nd Int. Joint Conf. Autonomous Agents and Multiagent Systems*, Melbourne, Australia (July 2003), pp. 225–232.
22. C. Rose, M. F. Cohen and B. Bodenheimer, Verbs and adverbs: Multidimensional motion interpolation, in *IEEE Computer Graphics and Applications* **18**(5) (September–October 1998), pp. 32–40.
23. W. James, *The Principles of Psychology* (Holt, Rinehart and Winston, New York, 1890).
24. A. D. Baddeley, *Working Memory* (Clarendon Press, Oxford, 1986).
25. A. Miyake and P. Shah, Models of working memory: Mechanisms of active maintenance and executive control (Cambridge University Press, Cambridge, UK, 1999).
26. P. S. Goldman-Rakic, Circuitry of the prefrontal cortex and the regulation of behavior by representational knowledge, in *Handbook of Physiology*, eds. F. Plum and V. Mountcastle (American Physiological Society, Bethesda, MD, 1987), pp. 373–417.
27. J. D. Cohen, S. D. Forman, T. S. Braver, B. J. Casey, D. Servan-Schreiber and D. C. Noll, Activation of prefrontal cortex in a nonspatial working memory task with functional MRI, *Human Brain Mapping* **1**, 293–304 (1994).
28. E. E. Smith and J. Jonides, Working memory: A view from neuroimaging, *Cognitive Psychology* **33**, 5–42 (1997).
29. M. Wilkes, D. C. Noelle, J. M. Keller, K. Kawamura and M. Skubic, ITR: A biologically inspired adaptive working memory system for efficient robot control and learning, *NSF 02-168, Information Technology Research (ITR) Program Solicitation*, Awarded December 2003.
30. T. S. Braver and D. M. Barch, A theory of cognitive control, aging cognition, and neuromodulation, *Neuroscience and Behavioral Reviews* **26**, 809–817 (2002).
31. www.driesen.com/prefrontal_cortex.htm.
32. A. Cohen and U. Feintuch, The dimensional-action system: A distinct visual system, *Common Mechanisms in Perception and Action: Attention and Performance XIX*, eds. W. Prinz and B. Hommel (Oxford University Press, New York, 2002).
33. R. S. Sutton, Learning to predict by the method of temporal differences, *Machine Learning* **3**, 9–44 (1988).
34. D. Erol, J. Park, E. Turkay, K. Kawamura, O. C. Jenkins and M. J. Matarić, Motion generation for humanoid robots with automatically derived behaviors, in *Proc. IEEE Systems, Man and Cybernetics* (October 2003), pp. 1816–1821.
35. R. D. Hanlon and P. S. Maybeck, Multiple-model adaptive estimation using a residual correlation Kalman filter bank, in *IEEE Trans. Aerospace and Electronics Systems* **36**(2) (April 2000), pp. 393–406.
36. S. I. Roumeliotis, G. S. Sukhatme and G. A. Bekey, Sensor fault detection and identification in a mobile robot, in *Proc. of IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 13–17 October 1998, pp. 1383–1388.
37. R. Scattolini and N. Cattane, Detection of sensor faults in a large flexible structure, *IEE Proc. Control Theory Application* **146**(5) (September 1999), pp. 383–388.

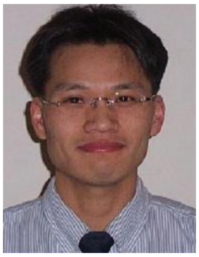
Kazuhiko Kawamura received his M.S. from the University of California, Berkeley and Ph.D. in EE from the University of Michigan, Ann Arbor. He was a Lecturer with the University of Michigan, Dearborn, and a Systems Planner at Battelle Columbus Laboratories, Columbus, OH. Currently, he is a Professor of Electrical and Computer Engineering and of Management of Technology and Director of the Center for Intelligent Systems (CIS), Vanderbilt University, Nashville, TN.

Dr Kawamura has published over 150 research papers, a book, and book chapters in the fields of intelligent systems, intelligent robotics, human-robot interfaces, behavior learning, computational intelligence and cognitive robotics. He directs research projects at the CIS in humanoid robots, computational intelligence, multiagent-based robot control, automatic behavior generation, and brain-machine interfaces. Dr Kawamura is a Fellow of IEEE. He was Founding Chair of the Technical Committee on Service Robots for the IEEE Robotics and Automation Society and was General Chair of the IEEE International Conference on Systems, Man and Cybernetics in 2000 in Nashville, TN. Dr Kawamura is a member of the editorial board of *Applied Intelligence*, *International Journal of Humanoid Robotics*, and *International Journal of Human-Friendly Welfare Robotic Systems.*

Alan Peters received his M.S. and Ph.D. in EE from the University of Arizona, where he was a Fellow of the American Electronic Association. He is currently an Associate Professor of Electrical Engineering and an Assistant Director of the CIS, Vanderbilt University, Nashville, TN. Dr Peters' current research involves intelligent robotics, computer vision and image analysis in three different areas: sensory-guided robotics, mathematical morphology, and electromagnetic scattering. Dr Peters is working on the integration of these functions through robot learning of sensory-motor coordination. The goal of this work is to enable a robot to learn through its own experiences and, ultimately, interact naturally with human beings.

Bobby Bodenheimer received his M.S. from the University of Tennessee and Ph.D. in EE from the California Institute of Technology. He is an Assistant Professor in Computer Science, Vanderbilt University, Nashville, TN. He was a visiting researcher at Microsoft Research and a postdoctoral fellow at the Georgia Institute of Technology. Dr Bodenheimer's research focuses on most areas of computer graphics and computer animation, in particular human figure animation. His research focuses on improving the quality and robustness of animations, emphasizing human figures. He received the NSF Faculty Early CAREER Development Award for computer animation research.

Nilanjan Sarkar received his M.E. in ME from the Indian Institute of Science and Ph.D. in ME and Applied Mechanics from the University of Pennsylvania. He was a postdoctoral fellow in the CS, ME and Psychology departments at Queen's University, Kingston, ONT. Dr Sarkar was a faculty member at the University of Hawaii. He is currently an Assistant Professor in the Department of Mechanical Engineering, Vanderbilt University, Nashville, TN. Dr Sarkar has been active in dynamics, control and robotics research. He has more than 50 publications in archival journals and conference proceedings. He is an Associate Editor for the *IEEE Transactions on Robotics and Automation* and the Vice Chair of the Robotics Panel of the ASME Dynamic Systems and Control Division. He is a member of the ASME and the IEEE.



Juyi Park received his M.S. in precision engineering and Ph.D. in ME from Korea Advanced Institute of Science and Technology (KAIST), Daejon, Korea. He was a postdoctoral fellow in the University of Texas at San Antonio from 2001 to 2002. Since 2002, he has been a research associate at the Department of Electrical Engineering and Computer Science in Vanderbilt University, Nashville, TN. Dr Park's research interests are suppression of residual vibration in flexible structure, robots for disabled people and real time robot control.

Charles Clifton received his B.S. degree in Computer and Electrical Engineering from the University of Florida in 2000. He is currently a Master's student and Research Assistant in the Center for Intelligent Systems. His research interests include the design of a working memory system for ISAC and embedded control systems.

Albert Spratley received his B.S. degree in Computer Engineering from Auburn University in Alabama. He is currently a Master's Student and Research Assistant in the Center for Intelligent Systems, Vanderbilt University, Nashville, TN. From 1999 to 2002 he served as an Engineer Trainee with Dynetics, Inc. in Huntsville, AL., where he was responsible for Simulation and Testing of various radar systems. His research interests include machine learning and control theory. Mr Spratley is a member of IEEE and a founding member of the AUVSI (Huntsville) Student Chapter at Vanderbilt.