Proceedings, IEEE Systems Man and Cybernetics (SMC 2003), pp 1816-1822, October, 2003, Washington, D.C.

Motion Generation for Humanoid Robots with Automatically Derived Behaviors

D. Erol, J. Park, E. Turkay, K. Kawamura
Center for Intelligent Systems
Vanderbilt University
Nashville, TN

O. C. Jenkins and M. J. Matarić
Center for Robotics and Embedded Systems
University of Southern California
Los Angeles, CA

Abstract –*In this paper, we present a method for motion generation from automatically derived behaviors for a humanoid robot. Behaviors are derived automatically by using the underlying spatio-temporal structure in motion. The derived behaviors are stored in a robot's long-term (or procedural) memory. New motions are generated from the derived ones with a search mechanism. In our approach, vision, speech recognition, short-term memory and decision-making operate in parallel with long-term memory in a unique architecture. This organization is intended for autonomous robot control and learning.*

Keywords: humanoid robot, behavior-based control, motion learning, long-term memory, Spatio-temporal Isomap

1 Introduction

Robots will eventually become a ubiquitous part of our daily life, similar to technologies for communication, automobiles, and transportation. In order to be ubiquitous, robots must function reasonably and autonomously under a variety of conditions, while adapting to environmental changes and continually pursuing their goals. Providing an autonomous control system for a humanoid robot is a complicated problem. For example, ISAC (Intelligent Soft-Arm Control), a humanoid robot developed at the Cognitive Robotics Laboratory of Vanderbilt University requires complex control in high dimensional space (Figure 1). Classical planning can be used to autonomously perform tasks for ISAC in a general and flexible framework. However, as tasks and environments for ISAC become more complex, classical planning-based control requires increasing computation that quickly becomes intractable for real-time control.

To consider the alternatives to classical control, we briefly review the taxonomy of control provided by Matarić [11,12]. A key goal of control is efficient action in real-time [13]. The planning paradigm has a *sense-plan-act* organization for control. A robot provides control through sense-plan-act cycle, consisting of: 1) modeling its status and the external world utilizing sensing, 2) deliberating to compute a goal-achieving plan based on the world model, and 3) acting to execute the computed plan.

Sense-plan-act can be thought of as "thinking hard" to provide control. In contrast, *reactive systems* take a "don't think, act" approach to control and use no deliberation; they react to the current situation based on a set of predefined rules. Reactive systems are a very fast means of providing control, but lack the flexibility for general autonomy. In an attempt to remove dependence on time, *hybrid systems* (or three-layer systems) take a "think and act separately" perspective by combining a high-level planner with a set of reactive modules. These systems plan at an extended time-scale with deliberation while acting in real-time using reaction. The caveat with hybrid systems, however, is that an interface layer is needed to integrate long-term plans with immediate situations.

Behavior-based control is an alternative to hybrid systems that "think the way they act" [12]. Behavior-based control systems consist of a collection of behaviors, processes that provide control commands for a given situation, that interact through a structured set of connections. Deliberation capability results from the interactions across behaviors, rather than in a single centralized module, allowing a robot to think and act at simultaneously in a dynamic world.

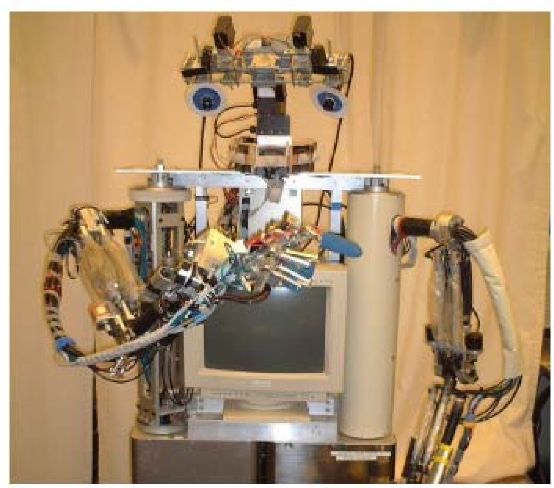

Figure 1. ISAC, a humanoid upper torso, equipped with two six-DOF McKibben actuated arms, anthropomorphic end-effectors called PneuHands, a four-DOF stereovision head, voice recognition, and a monitor for graphical display.

A significant issue in using behavior-based control is selecting and implementing a useful collection of behaviors. Once behaviors are constructed, the behavior-based architecture provides a means to structure them for autonomous control. The construction of behaviors, however, can be non-trivial due to challenges such as scalability and suitability to unanticipated situations. Recently, Jenkins and Matarić have proposed a method for *automatically deriving skill-level robot behaviors* from captured real-world human performances [8]. Deriving behaviors from humans, through motion capture or teleoperation, is attractive because it can leverage a human's underlying movement structure for activities the human considers to be important, consciously or otherwise. Additionally, Jenkins and Matarić can automatically derive "vocabularies", similar to [16], that allow for intuitive manual refinement and can be constructed into nonlinear dynamical systems for control.

In this paper, we propose a method for motion generation using derived skill-level behaviors within the Intelligent Machine Architecture (IMA) for autonomous control. IMA is a *task-level* control architecture containing various subcomponents (or agents) that handle the perception, control, and proprioceptive aspects of the robot [14]. Other agents within an IMA system sense the robot's current situation and continually provide task-level directives the index into a vocabulary of skills to achieve the robot's current objectives.

This paper is organized as follows. Section 2 presents the multi-agent based architecture developed for ISAC. In Section 3, we present the system for deriving the behaviors. Section 4 describes the role of behaviors in the procedural memory. Section 5 explains the generation of new motions. Section 6 presents the results and evaluation for reaching behaviors. Sections 7 and 8 discuss related work and present the conclusions of the described work.

2 Multi-Agent Based Architecture

The IMA is an agent-based software system that permits robot control through collections of cooperating software agents (Figure 2). In a general sense, IMA is an architecture for concurrently executing IMA agents on separate machines that perform as a group through extensive inter-agent communication. Atomic IMA agents are autonomous entities that have one or more execution threads. Typically, an atomic agent cannot perform useful activity independently. Collections of IMA agents interact to complete tasks by providing capabilities, such as sensing the external environment. For robot control, various types of IMA agents exist, including: hardware agents for accessing sensors and actuators, environment agents for abstracting object and environment interactions. Within IMA, the robot itself is abstracted as a *self-agent (SA)*, and the state of external entities, such as people, is abstracted in the form of human agents [9].

The self-agent uses a memory database structure, consisting of short-term and long-term structures, to determine the appropriate situational control commands for the robot. The short-term memory (STM) uses the Sensory EgoSphere (SES) to represent short-term external events in the environment [15]. The SES is a data structure that provides a short-term-memory to store events, such as the state of external human agents. The long-term memory (LTM) contains information about procedures considered intrinsic to the robot. The self-agent uses both the short and long term memory structures to provide control. The STM is used by the self-agent to provide an estimate of the current external state for determining appropriate task-level intentions for the robot. Based on these intentions, the self-agent uses procedures in LTM to provide control commands to accomplish the robot's intentions. In this paper, we propose a self-agent that uses derived behaviors as procedures to produce motion control for achieving robot objectives. Within the self-agent, a *central executive controller (CEC)* uses the derived behaviors in LTM.

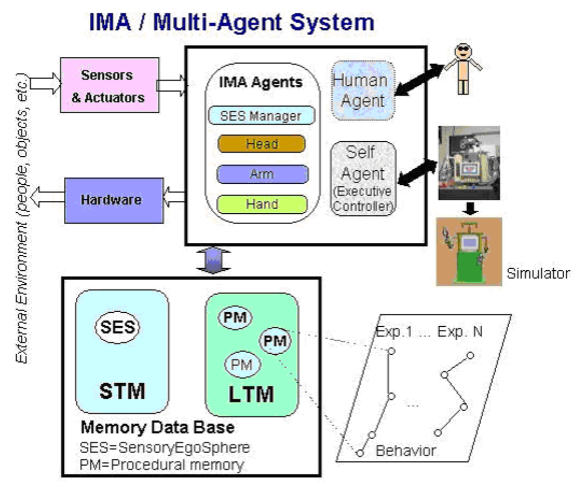

Figure 2. Component view of the IMA multi-agent-based system

3 Automated Behavior Derivation

To utilize behavior-based control properly, a substrate of behaviors is needed that can express the desired range of the robot's capabilities. In deriving such skill capabilities for a robot, we assume that captured human motion is structured by underlying behaviors and that the performed activities are representative of the robot's desired capabilities. We further assume that each underlying behavior produces motion with a common spatial signature and is typically performed in sequence with common preceding and subsequent behaviors. Given these assumptions, we can automatically derive a *behavior vocabulary* using a spatio-temporal extension [8] of Isomap [18], an nonlinear spectral dimension reduction technique. Such derived behavior vocabularies are structurally similar to *Verbs and Adverbs* vocabularies

[16] in that each behavior is defined by a set of exemplar motions that are generalized through interpolation.

The behavior derivation method (described in depth in [8]) 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 events, with each segment assumed to be an atomic motion. Several methods exist for segmenting time-series data. We use Kinematic Centroid Segmentation [8], which treats each limb as a pendulum and greedily seeks "swing" boundaries. Segmentation with time normalization produces an ordered set of data points in a D*N dimensional space, where D is the number of DOFs and N is the number of frames in each motion segment. Spatio-temporal Isomap works by finding *common temporal neighbors (CTN)*, pairs of points with small spatial distances whose sequential neighbors also have small spatial distances. Through transitivity, CTNs for *CTN connected components* that result in easily distinguishable clusters in the produced embedding. Furthermore, the number of clusters is found automatically using no *a priori* cardinality estimate. Each cluster, called a *primitive feature group*, is a set of exemplars with a common spatio-temporal theme that is indicative of some underlying behavior. Interpolation is used within a cluster's set of exemplars to sample new motions from the underlying *primitive behavior*. By densely sampling cluster exemplars, each primitive behavior is uncovered as a set of motion trajectories that form a low-dimensional flow field manifold in joint angle space. Additionally, further embedding/clustering iterations can be applied to successive embeddings for clustering higher-level *meta-level behaviors* as sequential transition models of primitive behaviors. 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 motion [8].

4 Role of Behaviors in Procedural Memory

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) (Figure 3). 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 [3].

Each behavior in the vocabulary is stored as a PM unit. Each primitive behavior is stored as 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.

The Central Executive Controller (CEC) uses the PM index to search the desired initial and final position of motion. It uses the PM units to translate robot goals into control commands for accomplishing the goal by searching for PM units suitable for accomplishing goals and then uses the indexing structure of the PM units to produce the desired motion. For example, if the robot has the goal of *"reach-to XYZ"*, the CEC can determine that PM *"reach"* at coordinates *"XYZ"* will accomplish this goal. In a sense, the goal *"reach-to XYZ"* spawns the intention to *"reach, XYZ"*, and this intention directly specifies which action to take.

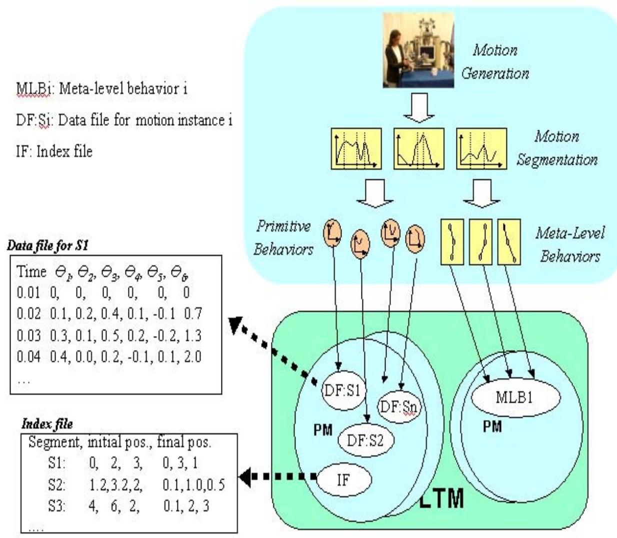

Figure 3. Structure of the LTM database

5 Generating Motion from Stored Behaviors

ISAC can react to the environmental changes autonomously by generating desired motions from the stored behaviors. The behaviors stored in the PM are managed by a central planner, which performs searching across behaviors and within each behavior. Generating new motions from the generic behaviors involves a planning algorithm and an interpolation method.

The search mechanism in the CEC receives an estimate of the current external state of the environment from the Sensory EgoSphere (SES) to determine appropriate task-level intentions for the robot. Based on these, CEC uses two-step tasks to search the LTM to provide control commands to accomplish the robot's intentions (Figure 4). As mentioned above, PM units store primitive behavior motion data as a dense sampling of motion trajectories representing the span of variations of a behavior. In the case of a match, the motion trajectory is sent to the CEC. If there is no match, a new motion is interpolated using the local neighbor motions from the primitive behavior database. Shepard's interpolation [17] is used. The generated motion data are sent back to the planner. The actions are then sequenced and sent to the hardware controller(s) to accomplish the robot's intentions.

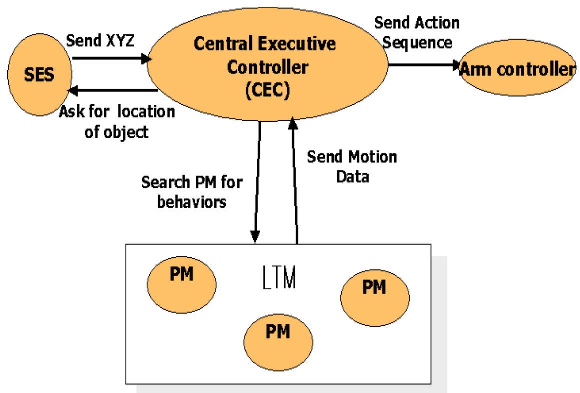

Figure 4. Structure of the search mechanism for generating new motions from stored behaviors.

6 Validation

6.1 Experiment

ISAC is driven by a human operator to reach to 5 random points (A, B, C, D, E) on a table, a limited working space, as shown in Figure 5.

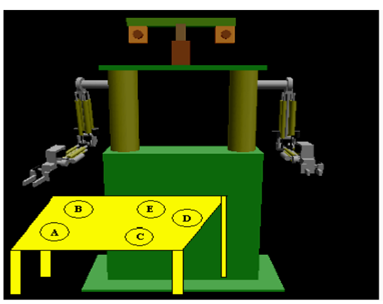

Figure 5: ISAC is taught to reach 5 points on a table

The motion data stream consisted of 1241 frames, which contained only one reaching behavior. The stream was segmented using the Kinematic Centroid Method [8] resulting in 10 segments. Each segment was normalized to 100 frames. The parameters for the K-nearest neighborhood, common-temporal (c_{ctn}) and adjacent temporal neighborhood (c_{atn}) constants for spatio-temporal Isomap were selected (*). The structure was embedded to 3 dimensions. The segments that were similar came closer after applying spatio-temporal Isomap (Figure 6; each segment in the embedding is numbered). The primitive feature groups were produced through clustering. The closer segments were clustered in the embedded space within 0.1 of the diagonal of the embedding bounding box to derive the primitive behaviors [4]. The clustering resulted in two derived primitive behaviors: reaching and returning to home positions.

(*) K –nearest neighbors = 3, c_{ctn} =100 and c_{atn} = 1.

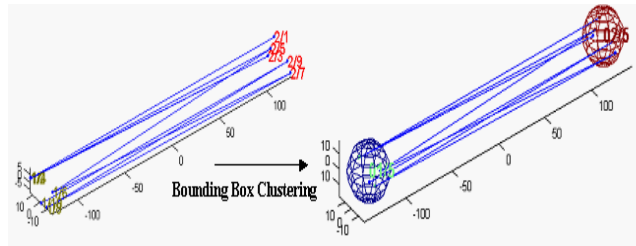

Figure 6: Embedding and clustering of first motion stream using spatio-temporal Isomap. Lines show temporally adjacent segments.

100 random coordinates with the bounding box method were interpolated to obtain new joint space trajectories for each primitive behavior [17]. The reaching behavior and return to home behavior are interpolated (Figure 7).

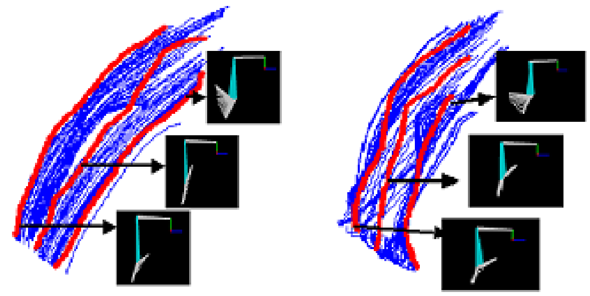

Figure 7: Results from interpolating selected action units. Each plot shows trajectories in Cartesian coordinates. (Left) reaching, (Right) returning to home.

Spatio-temporal Isomap was applied a second time (**). Sweep-and-prune clustering was then used in the embedded space with a 0.1 threshold value to derive the meta-level behaviors (Figure 8). The segments from the same primitive feature groups became proximal after the first application of Spatio-temporal Isomap. After the second application of spatio-temporal, the primitives typically performed in sequence were collapsed (Figure 9). Thus segments were collapsed from the corresponding primitive feature groups in the sequence into the higher-level behaviors. The meta-level embedding resulted in the same number of clusters as in the primitive embedding. This convergence of the embeddings indicates there is no higher-level structure in the motion greater than the primitive level. Thus, we consider the vocabulary has one meta-level behavior that encompasses the two primitive behaviors.

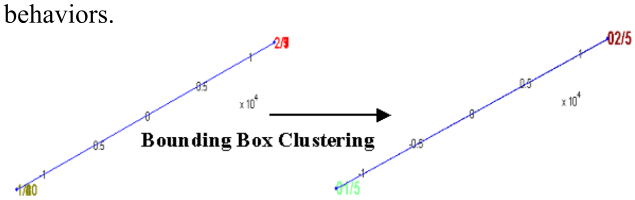

Figure 8: Embedding and clustering of second application of Spatio-temporal Isomap. Proximal motion segments are grouped to form behaviors.

(**) K-nearest neighbors 4, c_{ctn}=100, c_{atn}= 4.

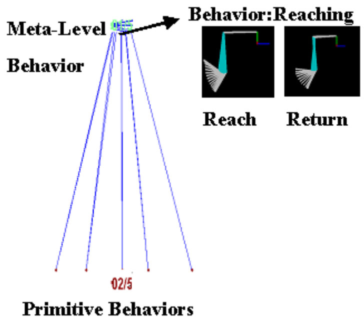

Figure 9: The transitions between the segments that derive meta-level behavior for each reaching motion. Lines indicate transitions between actions.

6.2 Application

A human using a finger to point can produce a demonstration for ISAC, who attends to the objects in its workspace and generates reaching motions to those objects. The application begins with a speech cue from the human, which directs the robot's attention to an object. To direct ISAC's attention to unknown position of the object, the human tells ISAC to find the new the location of the object such as “Reach-to”. The Human Agent sends this intention text to the Self Agent (SA), and activates the Human Finger Agent (HFA) inside the Human Agent and parses the name of the object. The HFA finds a pointed finger to fixate on the object. Next, the Head Agent is activated to find the pointed finger place and camera angles information is forwarded to the Sensory EgoSphere, which returns the coordinates of the object. Based on these intentions, the CEC uses procedures in LTM to retrieve the motion data to accomplish the robot's reaching intention. The desired motion data sequence is sent back to the CEC, and then sent to the Arm Agent to perform the reaching task.

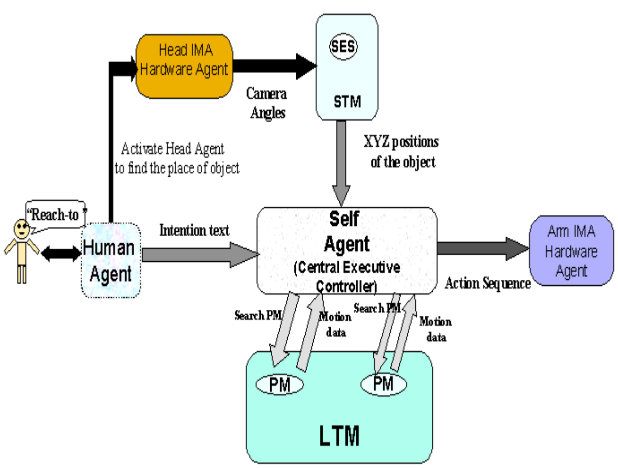

Figure 10: Schematic representation of the system communication for the demonstration of Reaching the New Point

7 Related Work

Autonomous control of humanoid robots has been a topic of significant research. Preprogramming and tele-operation remain common methods for most applications, such as lift trucks driven by robotic operators [5]. However, these approaches are quite tedious for complex tasks and environments. Huber and Grupen [6] have presented hybrid architecture for autonomous control given a manually constructed *control basis*. Consequently, their approach to control is dependent on the quality of the design and construction of the control basis, whereas the automated derivation we used [8] leverages the structure of humans.

Other methods for automatically deriving behaviors from human motion are not always suitable for autonomous control. Bregler [1] presented automatically developing groups of behaviors in the form of *movemes* from image sequences of a moving human. Complex motion can be described by sequencing the movemes generated for each limb, but indexing the movemes for coordinated motion generation is not obvious. Li et al. [19] and Kovar et al. [10] presented work for building linear dynamical systems and directed graphs from motion capture. For generating motion for control, however, these systems require a constraint optimization to be applied on structures that may not be parsimonious.

Human motion has been used as a basis for controlling humanoid robots. Ijspeert et al. [7] presented an approach for learning non-linear dynamical systems with attractor properties from motion capture. Their approach is useful for perturbation-robust humanoid control, but is restricted to deriving a single class of motion.

Brooks and Stein [2] developed an integrated physical system including vision, sound input and output, and skillful manipulation, which are all controlled by a continuously operating parallel communication. The goal was to enable the resulting system to learn to "think" by building on its bodily experiences to accomplish progressively more abstract tasks [2]. ISAC’s motion learning system is similar. In our work, vision, speech recognition, short-term memory, self-agent and behaviors stored in long-term are all operating parallel in a unique architecture in order to control the humanoid robot autonomously.

8 Conclusion

We have described an approach for generating new motions from derived behaviors. We were able to derive the behavior vocabularies based on the Spatio-temporal Isomap method. We stored these behaviors in a robot’s long-term memory, and used a search mechanism to generate autonomous control based on robot's perceived goals. We demonstrated the usefulness of our approach with respect to generating new motions for new environment changes. Our approach works together with speech and vision recognition, short-term memory and a self-agent in a unique architecture. Experiments were performed for a simple task of reaching to specific points. However, we believe that this approach will be more useful for more complicated tasks such as grasping various objects in various poses, manipulating screwdrivers, etc.

Acknowledgments

This collaborative research has been supported under a DARPA MARS2020 grant through NASA-JSC (NAG-1446), DARPA MARS Program grant DABT63-99-1-0015 and ONR MURI grant N00014-01-1-0890. We would like to acknowledge discussions with David Noelle and Richard Alan Peters that contributed to this work.

References

[1] C. Bregler, "Learning and Recognizing Human Dynamics in Video Sequences", IEEE Conference on Computer Vision and Pattern Recognition, pp. 568-574, 1997.

[2] R. A. Brooks, L. A. Stein, "Building Brains for Bodies", Autonomous Robotics, Vol 1,No.1, pp. 7-25, November 1994.

[3] R. Carter, *Mapping the Mind*, University of California Press, 2000.

[4] J. D. Cohen, M. C. Lin, D. Manocha, and M. K. Ponamgi, "ICOLLIDE:An interactive and exact collision detection system for large-scale environments", Proc. ACM Interactive 3D Graphics Conference, pp 189–196, 1995.

[5] H. Hasunuma, M. Kobayashi, H. Moriyama, T. Itoko, Y. Yanagihara, T. Ueno, K. Ohya, K. Yokoi,"A Tele-operated Humanoid Robot Drives a Lift Truck", Proc. International Conference on Robotics & Automation, Washington DC, May 2002.

[6] M. Huber R. A. Grupen, "A Hybrid Architecture for Learning Robot Control Tasks", Robotics Today, Vol. 13, No.4, RI/SME, 2000.

[7] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Trajectory formation for imitation with nonlinear dynamical systems", Proc. International Conference on Intelligent Robots and Systems, pp. 752–757, 2001.

[8] O. C. Jenkins and M. J. Matarić, "Automated Derivation of Behavior Vocabularies for Autonomous Humanoid Motion", Proc. 2nd International Joint Conference on Autonomous Agents and Multiagent Systems, Melbourne, Australia, 2003.

[9] K. Kawamura, D. C. Noelle, K. A. Hambuchen, T. E. Rogers and E. Turkay, "A Multi-Agent Approach to Self-Reflection for Cognitive Robotics", International Conference on Advanced Robotics, Coimbra, Portugal, June 30- July 3 2003.

[10] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs", Proc. 29th Annual Conference on Computer graphics and interactive techniques", pp. 473–482, ACM Press, 2002.

[11] M. J. Matarić, "Behavior-Based Control: Examples from Navigation, Learning, and Group Behavior", Journal of Experimental and Theoretical Artificial Intelligence, special issue on Software Architectures for Physical Agents, 9(2-3), H. Hexmoor, I. Horswill, and D. Kortenkamp, eds., 1997, 323-336.

[12] M. J. Matarić., "Behavior-Based Robotics", MIT Encyclopedia of Cognitive Sciences, Robert A. Wilson and Frank C. Keil, eds., MIT Press, April 1999, 74-77.

[13] R. R. Murphy, "Introduction to AI Robotics", Massachusetts Institute of Technology, 2000.

[14] R. T. Pack, D. M. Wilkes, G. Biswas, and K. Kawamura, "Intelligent Machine Architecture for Object-Based System Integration", Proc. International Conference on Advanced Intelligent Mechatronics, Waseda University, Japan, June 1997.

[15] R. A. Peters II, K. A. Hambuchen, K. Kawamura and D. M. Wilkes, "The Sensory EgoSphere as a Short-Term Memory for Humanoids", Proc. 2nd International Conference on Humanoid Robots, pp 451-459, 2001.

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

[17] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data", Proc. of the ACM National Conference, pp 517–524,ACM Press, 1968

[18] J. B. Tenenbaum, V. de Silva, and J. C. Langford," A global geometric framework for nonlinear dimensionality reduction", Science, 290(5500):2319–2323, 2000.

[19] L. Yan, "Motion Textures: A Two-Level Statistical Model", Siggraph 2002, 21-26 July 2002, San Antonio.