Developing Humanoid Robots for Real-World Environments

Adrian Stoica[1], Michael Kuhlman[2], Chris Assad[1], Didier Keymeulen[1]

[1]*Jet Propulsion Laboratory, California Institute of Technology, USA*, adrian.stoica@jpl.nasa.gov

[2]*Rensselaer Polytechnic Institute, USA*

Abstract— **Humanoids are steadily improving in appearance and functionality demonstrated in controlled environments. To address the challenges of operation in the real-world, researchers have proposed the use of brain-inspired architectures for robot control, and the use of robot learning techniques that enable the robot to acquire and tune skills and behaviours. In the first part of the paper we introduce new concepts and results in these two areas. First, we present a cerebellum-inspired model that demonstrated efficiency in the sensory-motor control of anthropomorphic arms, and in gait control of dynamic walkers. Then, we present a set of new ideas related to robot learning, emphasizing the importance of developing teaching techniques that support learning. In the second part of the paper we propose the use in robotics of the iterative and incremental development methodologies, in the context of practical task-oriented applications. These methodologies promise to rapidly reach system-level integration, and to early identify system-level weaknesses to focus on. We apply this methodology in a task targeting the automated assembly of a modular structure using HOAP-2. We confirm this approach led to rapid development of a end-to-end capability, and offered guidance on which technologies to focus on for gradual improvement of a complete functional system. It is believed that providing Grand Challenge type milestones in practical task-oriented applications accelerates development. As a meaningful target in short-mid term we propose the '*IKEA Challenge*', aimed at the demonstration of autonomous assembly of various pieces of furniture, from the box, following included written/drawn instructions.**

I. Introduction

In the last few years a strong interest in humanoids research has led to important advances in appearance and functionality. Yet, with a few notable exceptions such as HRP-3 [1], the current humanoids are designed for laboratory environments rather than for industrial or other real-world settings. Their reliable and safe operation in real-world environments is deterred by a number of challenges that are still to be overcome. Some of them are illustrated in the following with two real world scenarios.

The first scenario relates to the use of future humanoids in the construction and assembly jobs. A simple example is the assembly of a piece of furniture from IKEA. Unpacking may require the careful use of a cutter, and extraction of possibly fragile objects (e.g. glass) from the box. Interpretation of visual and tactile information would inform about the progress of the action. Assuming all pieces can be arranged and unpacking complete, the most significant challenge remains the interpretation of the assembly instructions. This involves recognition of components in the diagrams, most likely after some manipulation that offers a view more consistent with the pictorial representation in the instructions. One can continue to elaborate on challenges in this scenario, which nevertheless deals with a simple, static, predictable world. Sensory-motor coordination and cognitive abilities beyond what is currently available are needed.

In a second, more complex scenario, consider humanoids involved in rescuing people trapped in buildings damaged by fire or earthquake. Accessing a building and carrying humans out of danger zones poses critical challenges to sensory-motor coordination and reasoning, requiring: agile and intelligent locomotion (for climbing ladders, walk on slippery angled roofs/ surfaces); dexterous manipulation (for use of tools possibly forcing an opening, pull-out and carry victim); and cognitive perception (e.g. recognize deformed objects). The cognitive challenges implicit in these tasks require handling unknown situations, managing dynamic environments, coordination with humans, etc. Reasoning must result in predicting consequences of actions (of own/others) or (in)action on environment. Planning and navigation in real-time in dynamic/complex environments is needed. An example of reasoning is 'That block is falling exactly on me, I'd better move to the area near the door, which appears safe'.

To address such challenges researchers have looked at the animal world for inspiration. In particular sensory-motor coordination and cognition have benefited from inspiration from the mammalian brain, while learning in biology has fueled the design of robot learning techniques. In the first part of the paper we propose contributions to these areas.

The development of real-world systems has benefited from recent iterative and incremental development methodologies [2] that proved fast and reliable progress toward the target capability. In the second part of the paper we propose the adoption of these methodologies to robotics, and we illustrate it through the benefits we experienced in the context of a practical task-oriented application.

Section II presents a cerebellum-inspired model that demonstrated efficiency in the sensory-motor control of anthropomorphic arms, and in gait control of dynamic walkers. Section III presents a set of new ideas related to robot learning, emphasizing the importance of developing teaching techniques that support learning. Section IV demonstrates the benefits of iterative and incremental development in the context of an assembly demo with HOAP.

II. Building Bio-Inspired Robot Brains

The first challenge is to provide efficient sensory-motor control for biped mobility, energy efficient trajectory and motion generation. Animals are very good at maneuvering and manipulating objects in unstructured complex environments, and typically possess dynamic, nonlinear and high degree-of-freedom (DOF) bodies that are intractable to conventional control methods. The Biomorphic Robotics design approach to more dexterous, agile robotics is to emulate the biomechanics and highly effective control algorithms found in the vertebrate nervous system. Important differences from conventional robotics include (1) exploiting the body's natural dynamics, (2) "springy" compliant actuation for efficient force control and energy recovery, (3) no precision sensors; instead, large arrays of fast and cheap "sloppy" sensors, for over-sensing for kinesthetics and proprioception, (4) no precision machining; instead, reliance on adaptive control to learn dynamics and adapt to changes over time, and (5) intelligent coordination of feedforward and feedback control strategies.

A. Cerebellum model

Agile control of high DOF, non-linear dynamic systems requires accurate dynamic state estimation (DSE), a fundamental component of a wide variety of sensor fusion, signal processing and control tasks in engineering. We take inspiration from the cerebellum, the "engine of agility" [3] in the vertebrate brain. The unique neural architecture of the cerebellum appears optimized for learning sensory-motor dynamics and predictive control of high DOF nonlinear systems. Although details of its function are not fully understood, the cerebellum is thought to perform DSE to achieve dexterous, coordinated and dynamically efficient motor output [4]. Our work at JPL has been to develop algorithms to capture the functionality of the cerebellum. We have begun to simulate, implement and deploy these algorithms for dynamic control of biomorphic robots [5,6].

We first developed a cerebellum-inspired neural network to perform DSE and predictive control. Controlling a dynamic system requires knowledge of the system's state and its response to new motor commands, a task that naturally lends itself to state space methods of solution. The cerebellum model can orchestrate dexterous, agile movements by learning: (1) to estimate and predict trajectories through state space, i.e., modeling the system dynamics, (2) decision boundaries around regions of state space in which to initiate actions to achieve desired goals, and (3) to modulate motor commands to redirect the trajectory as needed. The model estimates the current state by combining incoming sensor measurements and the implicit learned model of system dynamics, predicts the trajectory, and then can initiate actions in appropriate regions of the space.

B. Control of anthropomorphic arm

The cerebellum network was first demonstrated on a 2-link robot arm built with antagonistic pairs of McKibben air muscles (Fig. 1) [6]. McKibbens are fast and strong with muscle-like dynamics but are very difficult to control by conventional means. They can be used as variable spring constants to control compliance and mechanical response properties. The arm has a gripper end effector to hold and throw a tennis ball. Trajectory data was collected during multiple throwing trials and used to train the model offline. The data were projected onto 2-dimensional state space maps, from which the network learned to estimate state variables and decision boundaries. It successfully learned to trigger the grip release at the proper state for the ball to hit a target.

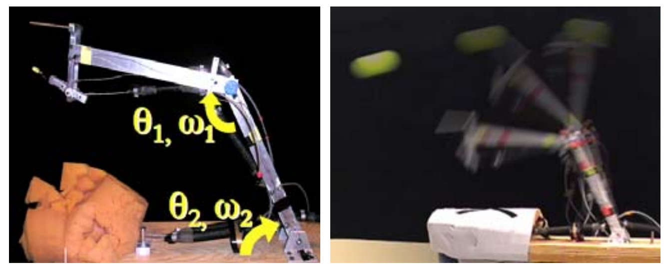

Fig. 1 Left: Biomorphic 2-link arm with state variables highlighted. Right: Three frames at 50 ms intervals near the time of release.

C. Toward the control of dynamic walkers

The cerebellar algorithm should be general enough to facilitate a wide range of dynamic robotic systems – because the model learns an implicit representation of the system dynamics, in principle it could be applied to any number of mechanical systems requiring DSE. In particular, this method should prove efficient for learning dynamic trajectories in legged walkers, including humanoid bipeds.

Progress in practical biped walking has been held back by hurdles in stability and power efficiency. Recently, a breakthrough was reported in human-like power efficiency based on "passive-dynamic" walking bipeds [7]. These are designed with efficient biomechanics to walk down a slight slope with no actuation or sensing, and then minimal actuation is added to walk over flat ground. The resulting dynamic walkers are at least an order of magnitude more power efficient than more conventional biped designs that are kinematically controlled for quasi-static stability (always maintain center of mass over a supporting foot). However, stability remains a serious challenge – the bipeds fall after several meters because there is no active feedback or control.

Our goal is to develop and demonstrate an "artificial vestibular system" for dynamic balance and stability in a walking biped platform, based on our model of cerebellum. The cerebellum network should learn to estimate relevant state variables from onboard sensors, predict trajectories, and then modulate actuation to improve stability. Preliminary results from a simplified 2-D biped simulation indicate that the cerebellum model can learn to predict instabilities during walking that can lead to falls (Fig. 2). The next step will be to learn actuator responses to correct the instabilities to avoid the fall. Questions of interest include: What is minimal DOF, i.e., determine a sufficient set of actuation/sensing to achieve dynamic balance? How to scale the control system to higher dimensions? How to add actuation/control for additional functionality; e.g., turning, start/stop, walking over rough terrain, in urban environment, carrying payload, etc.? Answering these questions will help enable the first dynamically stable fully autonomous walking biped.

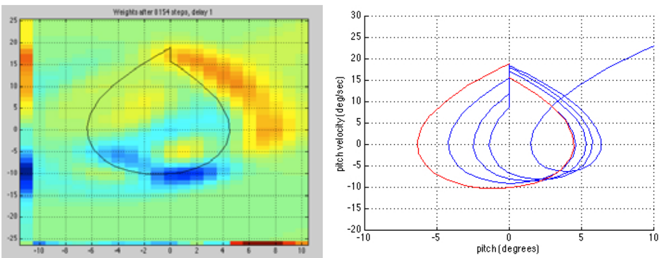

Fig. 2 Learned weights (left) and a 5-step walking trial (right), shown in the state space of pitch velocity vs. pitch angle.

III. Developing Capabilities by Learning

The importance of learning has long been recognized as paramount in addressing the challenges of the real world, which differs from the models with which the robot starts its operation. While unsupervised learning has merits, the experience in the human world indicates that learning under the control and supervision of a teacher is critical for cognitive development. The focus in robot learning has been on learning techniques; yet, as argued for example in [8], it is important to dedicate more effort on the development of methods and techniques that are used to teach the robots. This is especially relevant to humanoids. If indeed being taught by humans is the most important factor in robot skill acquisition, then this favors humanoids over other robotic forms, as optimal for being taught by humans, since humans best relate to them due to the resemblance in form.

A. Robot Fostering

We enumerate a set of principles for which more detail in given in [9], and which are considered to be important for development of cognitive capabilities; 2 and 4 are departures from the conventional view. These are:

1) The essence of endowing robots with intelligence is robot development (grounded, embodied, situated, gradual) - not robot programming. Development allows building of perceptions, schema, representations, and behaviors directly through interaction with the real world environment (a set of innate/pre-programmed capabilities is assumed).

2) The key to cognitive development is a focus on teaching techniques, at least equal to the effort as on robot learning techniques. This may include providing examples of gradual increase in difficulty (robot shaping), building of training sets, or helping the robot ("holding it by the hand") while learning.

3) Important techniques for fostering/teaching by a human or robot include imitation, demonstration, guidance (analogic teaching), and explanation. Imitation has received important attention for more than a decade [10], both for robots imitating humans and imitating other robots (acting as teachers) [10,11]. Still, one needs to dedicate more effort on the understanding of the task that is to be accomplished by the movement learned by imitation ('task-oriented application'). Demonstration provides a solution on how to solve a problem. Direct help from the human, guiding the movement, or supporting the robot, positioning it by hand, etc., greatly help the robot.

4) *Robot's ability to teach is a proof of cognitive learning.* The ability to teach is a validation that the essence of the task is grasped, that it is generalized and can be applied in a different context, that it is "conscious", meaning it has a flexible representation in context of self and outside world, and a rationale for why it is that way.

B. Developing teaching methodologies, enhancing learning

The following are new proposals for advances in automated teaching systems and automated learning systems:

1) Systems for automated teaching

- Development of a tutoring/training system that (semi/) automatically guides the humanoid cognitive development, monitors its progress, and chooses/implements the best training strategy
- Development of humanoid teaching methodologies, a teaching/training curriculum
- Techniques that allow humanoids to learn from instructional videos, movies and games
- Techniques for learning on-line (chat, internet games, etc, as teaching agents)
- A system that facilitates teaching by the robot, as a verification of cognitive learning (to enforce knowledge reformulation and abstraction)

2) Systems for automated learning

- *Dubitative systems.* Current systems 'believe' everything fed to them. 'Dubito ergo cogito, cogito ergo sum', expresses a specific analytic aspect of human cognition. It may worth adding such an aspect in cognitive robots.
- *Inquisitive systems.* Possibly in connection to a dubitative approach, it may be useful to design systems that persistently enquire for clarifications. This is related to means of determining an optimal learning strategy. Objectives may be to optimize cognitive improvement (knowledge, reasoning, etc), continuous model refinement, decisions on what to clarify/ask next.

IV. Incremental and Iterative Robot Development

Iterative and incremental development methodologies [2] have proven efficient in a number of areas, including software development. We propose their application to robotic development, in the context of practical task-oriented applications, using as target milestones the demonstrations of complete end-to-end integrated systems. We applied this methodology in the assembly application with HOAP-2, for which the first iteration is presented in the following. Future iterations will include JPL core technologies and brain-inspired architectures and learning discussed in prior sections.

A. Humanoid for assembly of a cubical structure

The objective of this effort was to develop robot capabilities for a practical task of autonomous assembly. We adopted an iterative and incremental development, focusing on a system-level, integrated system platform. The demonstration of a humanoid robot autonomously assembling a cubical frame also illustrates the potential of using humanoids for construction in a context where no other robot had yet succeeded: an end-to-end set of steps, in which the robot identifies a bar, walks toward it, picks it up, carries it to an assembly destination, and assembles the bars. A proof of feasibility demonstrated key component behaviors needed for this objective. A next level of iteration could include powerful core JPL robotic capabilities in vision, planning and navigation. Capabilities that the robot needed in order to perform the task autonomously include simple vision, walking, crouching (maintaining balance while walking/crouching with the rather long bar), grasping, eye-hand coordination, etc, and overall integration of these behaviours.

1) HOAP-2: We used the Humanoid Open Architecture Platform Second generation (HOAP-2) Fujitsu robot. The vision system consists of two CCD cameras, capable of capturing frames of 640 by 480 pixels. The body motions are provided through 25 servo actuators: 6 for each leg, 4 for each arm, 1 for each hand, 2 for the head, and 1 for its waist. There are 4 pressure sensors on the bottom of each foot, and an accelerometer and gyroscope inside the torso. Additional pressure sensors were mounted on the feet for balance.

2) *Command and Control:* We utilized a modular architecture to streamline code, maximizing computational efficiency and upgradability for future revisions. The *Command & Control Director* selects the action or sequence of actions to do next. It decomposes actions hierarchically into nested sub-actions. It uses a rule-based expert system to maintain a set of parallel state machines, and then manages real-time transitions based on events posted by the rest of the software system. In this way, it coordinates the decisions for the robot. A glue layer is used to instantiate and executes particular actions, resulting in a cohesive, integrated system.

3) *Speech* The humanoid uses speech recognition and synthesis to communicate with humans. HOAP is able to request assistance from the human to determine the best course of action to complete the task at hand. The human can also intervene on HOAP's progress and direct HOAP to do otherwise. Figure 3 illustrates example dialogue:

Human: "Pick up the bar."
Robot: *"Picking up the bar."*
Robot: *"I am unable to f nd the bar."*
Human: "Walk forward two inches."
Human: "Pick up the bar."
Robot: *"Picking up the bar."*
Robot: *"Bar is seventeen centimetres away."*
Robot: *"I now have the bar."*

Fig. 3 Example dialogue with HOAP

Speech generation was performed by integrating a third-generation Java-based synthesizer. This yielded relatively clean and understandable English, with a 95% comprehension rate. For speech recognition, the Microsoft English recognizer version 6.1 was used. To increase accuracy, a set of predef ned commands mapped generally used English phrases with task-specif c commands. In this way synonymous phrases such as "put the bar down," or "put it down" can be executed as the same command. Parameters can also be given and understood allowing input like "step forward three inches," or "move the bar two centimeters left." Using this system a recognition success rate of 99% was achieved.

4) *Visual perception:* The robot uses a world model, composed of models of the important objects in the surrounding environment, including rods, fastener joints, visual beacons, the robot's hands, and f xtures, along with their unique color patches. A model of the robot's current location is also kept. The robot uses the top-down vision to help disambiguate sensing from the bottom-up vision stack. Images are processed into color blobs for the left and right eyes. These are then converted into (x,y,z) 3D stereo readings, which are accurate to approximately 0.5 cm, and used to update object locations. The f nal result is a 3D model map of all the objects in the world around the robot, used for locomotion and manipulation during complex interactions between the robot and the world.

Firewire cameras with f sheye lenses were installed, integrated, and calibrated. These lenses are necessary because the physical constraints of the robot often prevent it from looking directly at its work area; the robot does not need to turn its head as much with f sheye lenses, reducing unnecessary movement. To accelerate computer vision techniques for properly identifying objects of interest, we added colored markers at key points of objects for proper identif cation, as seen in Figure 5. Blob f nding was implemented using the LTI vision routines library [13].

5) Walking

Two types of walking were designed: a predef ned sequence that gave ref exive walking, and precise computing for the position of each step. Reflexive walk was implemented using Zero Moment Point (ZMP) walking, with the center of gravity maintained over the robot's support structure at all times. We used a parametric walking scheme to define the size of each step, the height of each step, the angle the robot turns per step, and the position of the feet when the robot is standing. The parameters are adjustable, and were used to adapt to changes caused when the robot grabbed and carried a load; these can be further used for adaptation to optimal values for various environment/ context conditions.

The robot needs to move in various ways around the targeted object(s) in order to position itself in a suitable way for object handling and manipulation, including sidestepping, turning, and bending over without falling. The joint angle data for the robot as it performs a predefined walk was recorded and analyzed, then filtered in spectral domain. This was used to find rhythmic components of the walk cycle to generate a generic and stable walk pattern. From the recording of an unstable walk, data was filtered, attenuated and phase shifted in order to come up with a more stable walk. Once a basic walking pattern was achieved, it was used as a Central Pattern Generator, which controlled the default walk of the robot. When the robot is perturbed the walk can be modified (in terms of amplitude and phase of individual frequency components) to compensate for the disturbance. Foot feedback is used to make the foot more compliant.

A full closed-form leg solution supports precise motion of the body with respect to the feet and the objects around it. The robot can walk forward and backward and step sideways; it responds to commands to place itself within a millimeter or less. The precise walk imposed a tradeoff between speed of walking and accuracy of positioning – for 1 mm accuracy the operation was more than f ve times slower than normal. One can choose speed or accuracy depending on the particular type of movement required.

All link trajectories calculated by the walk module were combined into a single motion command to be executed by the pose interpolator. If instructed to move only a few centimeters the robot would lean that far, but if the robot was at the limit of its leaning it would take a step. If the robot is instructed to move farther it will take the appropriate number of steps, but scale the steps such that it can move an arbitrary distance and is not limited by a multiple of its number of strides.

6) Movement Actuation

One of HOAP-2's limiting factors is the relatively reduced number of degrees of freedom (DOF) – 25 total (about the same as a single human hand). Another, critical, limitation is the update speed of the motors that resulted in a 50Hz "jitter" stemmed from the inherent timing model of the Linux kernel. The interpolator from the Xenomai real time thread library and kernel patch was used to achieve acceptable timing resolutions that allow smooth operation of joints. The interpolator operated on the host PC, reducing the workload of the onboard microcontroller.

Closed-form inverse kinematic arm solutions were implemented for both left and right arms. This solution allows positioning the actuator anywhere in its workspace, as well as partial specif cation of the desired orientation. A grasp-planning module specif es the approach point and the location for picking up a desired object.

In order to simplify calculations, two out of the six DOF in the arms were sacrif ced, limiting mobility. Since the arm was going to be used for pickups and insertions, the positive Z axis (orientation out of the tip of the f ngertips) was the most vital for positioning. The arm uses the Z axis, plus the requested position, to constrain the elbow in space, by backing the hand's location out back to the elbow. Then, as long as a valid location is requested, this elbow position—along with the Z orientation of the forearm – determines the remaining three joints in the shoulder. The hand can spin about the Z axis, but at least it is known which way the f ngertips are pointing.

All this assumes that the determined elbow point is on the sphere of all possible elbow points that circles the shoulder point. This requires careful design. Since the upper arm has to point in some direction, the system takes the projection of the elbow point onto this sphere and uses that to determine shoulder azimuth and elevation, as well as bicep twist to get the forearm oriented properly.

In order to pick up a bar, or mate two parts together, maintaining correct orientation is an absolute requirement. However, specifying location and orientation reduces the workspace, since position and orientation are coupled in this system. This arm solution was found inadequate for the pool-cue motion required for inserting the bar onto the joint in a straight line. Another, specialized solution that was implemented only works with two of the joints to drive Y/Z position, and leaves end-effector orientation unspecified.

Grasp planning was initially calculated by assuming a horizontal bar relative to the robot. While the robot's arms are only 10 cm long per limb segment, the robot's chest is merely 6 cm deep. This results in a very small working envelope. It gets worse when one realizes that the hand's orientation is determined by the position of the elbow. Therefore, in order to grasp a horizontal bar, the hand and hence the elbow is forced to be in a horizontal plane relative to the bar, one of a set of planes that spins around the axis of the bar. Executing an effective grasp requires that the robot f rst moves its hand to a reachable withdrawal point, which is above the grasp point; and then moves from the withdrawal point down to the grasp point. The withdrawal point has to have the hand in the same orientation as the grasp, so that the hand does not twist much and knock the bar out of reach as the hand is going down.

B. Demonstration

Results from a demonstration of humanoid structure assembly in the laboratory environment with practical real world solution are detailed below and portrayed in Figure 4.

1) Positioning platform to insert bar.

When the robot has reached the proximity of the bar, it has to bend over and turn its head, in order to be able to see the assembly. Because the bar moves under the robot's hand, the robot has to use its vision to ensure that the bar is horizontal. Since the robot is leaning over, it has to subtract a factor for the tilted view of the world. The robot looks at the bar and moves its shoulder until the bar is horizontal.

2) Alignment of bar for insertion

The robot has to line the tip of the bar with the tip of the joint, in X and Z axes. There are not enough degrees of freedom in the hand to be able to do this, so the robot rotates its legs to align its orientation in the X and Z axes with the tip of the joint, controlled by visual servoing and verbal servoing.

3) Insertion

At this point, the robot has to make the insertion from its ready point. Doing so requires a precise and linear end effector Cartesian trajectory, which is difficult to implement with a low DOF system. A simple rotational movement does not suffice, especially for an 8 cm throw on the joint post. This is accomplished by a special arm solution that does a "pool cue" motion to keep the bar going in a straight line forward. To do this, one must sacrif ce hand orientation, which twists about the pinch axis of the rod. Also, the rod has to be aligned precisely, in only one position similar to bowling or shooting pool, so that the arm can move back and forth in Y while maintaining the same height in Z and the same position in X. When properly aligned, the robot was able to perform this insertion, in a smooth and controlled manner.

C. *Advantages of the iterative development*

The iterative approach led to a rapid development of an end-to-end capability, in this case the assembly task. It allowed understanding where the difficulties lied from the perspective of the entire development. It also acts as a visualization of a future capability. It provided a good illustration of the advantages of the iterative approach.

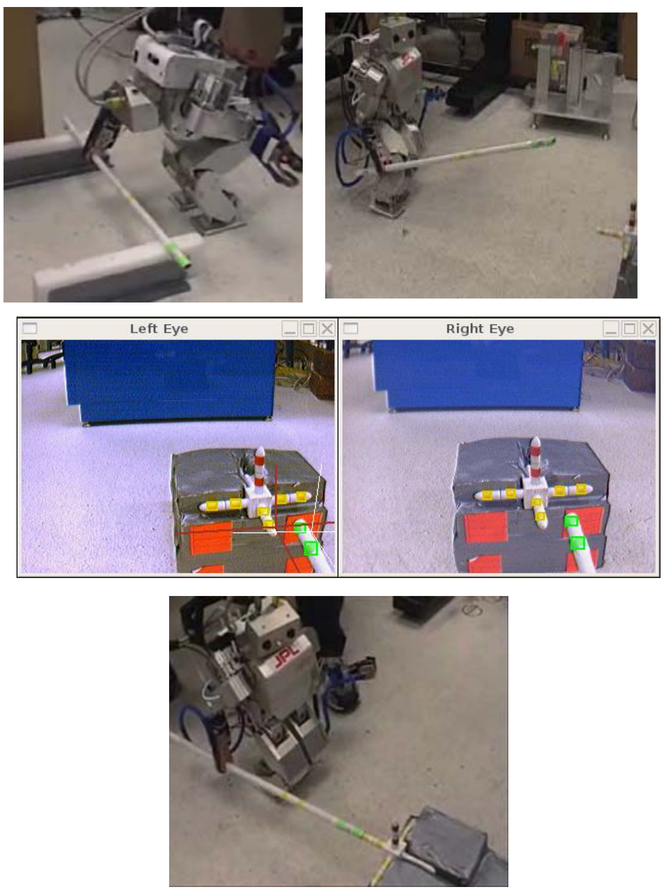

Fig. 4 Walkthrough of HOAP's assigned task

D. *Introducing the IKEA Challenges*

We believe that providing Grand Challenge type milestones in practical task-oriented applications accelerates development. Particularly relevant appear to be those that develop a technology that has most appeal for industry to transition. Competitions, such as the DARPA Grand Challenge, have a tremendous effect in motivating people and mobilizing resources. It is important to choose challenges that could allow the developed technologies to continuously transfer to industry. This maintains a reality check, and provides early/continuous return for technology investments.

As a challenge appropriate for the humanoids we propose the "IKEA Challenges". *The IKEA Challenges would aim the demonstration of a robot that can autonomously unpack an IKEA furniture package/box, identify the content of the box, and assemble the furniture from the pieces in the box, following the instructions found in the package.*

V. Conclusions

The paper presented contributions to brain-inspired architectures for sensory-motor coordination and learning. A cerebellum model was used to demonstrate control of anthropomorphic arms and gait with dynamic walkers. New concepts in teaching methodologies and learning systems were introduced. We proposed the application to robotics of incremental and iterative development and we illustrated it in the context of a modular assembly.

Acknowledgment

The work presented in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology for the National Aeronautics and Space Administration. The following personnel and students participated: John Myers, Aaron Schon, Jason Newton, Brent Allen, Alex Sanchez, Sal Trujillo, Sanjay Dastoor, Ling Xu. The work was funded by the JPL Advanced Concepts Program Office, managed by Dr. Neville Marzwell, NASA's CICT/ITSR Revolutionary Computing program, and by DARPA, under a program managed by Dr. Larry Jackel.

References

[1] K. Kaneko et al, "Humanoid Robot HRP-3," IEEE/RSJ Int. Conf. Intelligent Robots and Systems, Nice, France, Sept 2008, 2471-2478.

[2] C Larman and V. Basili, "Iterative and Incremental Development: A Brief History," Computer 36 (No. 6): pp 47–56, 2003

[3] I. Wickelgren, "The Cerebellum: The brain's engine of agility," *Science*, Vol. 281, pp. 1588-1590, 1998.

[4] M.G. Paulin, L.F. Hoffman, C. Assad, "A model of cerebellar computations for dynamical state estimation," *Autonomous Robots* 11:279-284, 2001.

[5] C. Assad, "An hypothesis for a novel learning mechanism in the cerebellar cortex," *Autonomous Robots* 11(3): 285-290, 2001.

[6] C. Assad, S. Trujillo, S. Dastoor, L. Xu, "Cerebellar dynamic state estimation for a biomorphic robot arm," IEEE Int Conf on Systems, Man and Cybernetics, Oct 2005, pp. 877-882.

[7] S. Collins, A. Ruina, R. Tedrake, M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science* 307:1082-1085, 2005.

[8] A. Stoica. "Robot Fostering Techniques for Sensory-Motor Development of Humanoid Robots" First IEEE-RAS Int Conf on Humanoid Robots, 2000, MIT.

[9] A. Stoica, et al "Humanoids for lunar and planetary surface operations", 5th IEEE-RAS Int Conf on Humanoid Robots, Tsukuba, Japan 2005

[10] A. Stoica, "Motion learning by robot apprentices: a fuzzy neural approach," PhD Thesis, Victoria Univ, Melbourne, Australia, 1995.

[11] A. Stoica, "Learning eye-arm coordination using neural and fuzzy neural techniques," Soft Computing in Human-Related Sciences, CRC Press, pp. 31-61, 1999.

[12] A. Stoica, "Robot Fostering Techniques for Sensory-Motor Development of Humanoid Robots," *J. Robotics and Autonomous Systems*, Special Issue on Humanoid Robots, **37**(2-3):127-143, 2001.

[13] LTI vision routines library [Online] available: http://ltilib.sourceforge.net/doc/homepage/index.shtml