Perceptual-Motor Sequence Learning Via Human-Robot Interaction

Jean-David Boucher and Peter Ford Dominey

CNRS, Institut des Sciences Cognitives, 67 Bd Pinel, 69675 France
{jdboucher, dominey}@isc.cnrs.fr
http://www.isc.cnrs.fr/dom/dommenu-en.htm

Abstract. The current research provides results from three experiments on the ability of a mobile robot to acquire new behaviors based on the integration of guidance from a human user and its own internal representation of the resulting perceptual and motor events. The robot learns to associate perceptual state changes with the conditional initiation and cessation of primitive motor behaviors. After several training trials, the system learns to ignore irrelevant perceptual factors, resulting in a robust representation of complex behaviors that require conditional execution based on dynamically changing perceptual states. Three experiments demonstrate the robustness of this approach in learning composite perceptual-motor behavioral sequences of varying complexity.

1 Introduction

The current research explores mechanisms that allow autonomous systems to acquire complex composed behaviors through a combination of interaction with the sensory-motor environment, and a human teacher. We have previously developed a system that allows the user to use spoken language to teach the AIBO ERS[1] robot the association between a name and a single behavior in the robot's repertoire (Dominey et al. 2005 [3]). More recently, we have extended this so that the system can associate a sequence of commands with a new name in a macro-like capability.

The limitations of this approach result from the fact that all of the motor events in the sequence are self contained events whose terminations are not directly linked to perceptual states of the system. We can thus teach the robot to walk to the ball and stop, but if we then test the system with different initial conditions the system will mechanistically reproduce the exact motor sequence, and thus fail to generalize to the new conditions.

Nicolescu & Mataric (2001, 2003) [5] [6] developed a method for accommodating these problems with a formalized representation of the relations between pre-conditions and post-conditions of different behaviors. They demonstrate how pre- and post-condition relations between the successive behaviors can be extracted, generalized over multiple training trials, and finally used by the robot

[1] http://www.sony.net/Products/aibo

S. Nolfi et al. (Eds.): SAB 2006, LNAI 4095, pp. 224–235, 2006.

to autonomously execute the acquired behavior. A similar approach to learning generalized behavioral sequences by a bimanual humanoid robot has been developed by Zöllner et al. (2004) [10]. Billard and Mataric have also demonstrated how related approaches can be used for teaching through demonstration and imitation (2001) [1]. Kaplan et al. (2002) [4] used a progressive shaping approach, and Saunders et al. (2006) [7] use a related method in which behavior components are successively developed in a hierarchy of increasing complexity.

The current research builds upon these approaches in several important ways. First we enrich the set of sensory and motor primitives, that are available to be used in defining new behaviors (defined in Tab. 1). Second, we enrich the human-robot interaction domain via spoken language and thus allow for guiding the training demonstrations with spoken language commands, as well as naming multiple newly acquired behaviors in an ever increasing repertoire. Third, we ensure real-time processing for both the parsing of the continuous valued sensor readings into discrete parameterized form, as well as the generalization of the most recent history record with the previously generalized sequence. This ensures that the demonstration, test, correction cycle takes place in a smooth manner with no off-line processing required.

Before going into the technical details we provide a simple example scenario with AIBO entertainment robot (Sony) that is our platform for these studies. In this case the user will teach the robot a form of collision avoidance through demonstration.

1. The user initiates the learning by commanding the robot with a spoken command "turn around" that does not correspond to a primitive command nor to a previously learned command.
2. The robot thus has no knowledge of what to do, and awaits further instructions.
3. The user commands the robot to "march forward" and the robot starts walking.
4. The user sees that the robot is approaching a wall, and tells the robot to stop.
5. He then tells the robot to turn right. Behind and to the right of the robot is the red ball.
6. When the robot has turned away from the wall and is facing the ball the user tells it to stop turning, and then tells it that the learned behavior demonstration is over.

Now let us consider the demonstration in terms of the commands that were issued by the user, and executed by the robot, and the preconditions that could subsequently be used to trigger these commands. The robot was commanded to "turn around." Because it had no representation for this action, it awaited further commands. The robot was then commanded to walk. Before it collided with the wall the robot was commanded to stop walking. It was then commanded to turn right, and to stop when it was in front of the red ball. Now consider the perceptual conditions that preceded each of these commands, which could be used in a future automatic execution phase to successively trigger the successive commands. The pertinent precondition to start walking was that the command to "turn around" had been issued. The pertinent precondition to stop walking was the detection of an obstacle in the "near" range by the distance sensor in the robots face. The pertinent preconditions for subsequently turning right are that the robot is near something, and that it has stopped walking.

The goal then is for the system to encode the temporal sequence of all relations (which include user commands, exteroception and proprioception values) in a demonstration run, and then to determine what are the pertinent preconditions for each commanded action relation. Likewise, it may be the case that perceptual relations were observed during the demonstration that were not pertinent to the behavior that the human intended to teach the robot. The system must thus also be able to identify such "distractor" perceptions that occurred in a demonstration, and to eliminate these relations from the generalized representation of the behavioral sequence. The following sections will define the system architecture and its functioning that meet these requirements.

2 Perceptual Motor Learning Architecture - PML

2.1 Platform

The robot platform that we employ is the Sony AIBO ERS7, running the URBI[2] operating and control system. URBI provides a systematic access to both the entire set of onboard sensors (including vision of the red ball and other objects, sensitivity to presses of the several buttons on the robot's back and head, joint angles, position-orientation sensors etc.), and to movement commands for walking, turning, backing up etc.

A central aspect of the PML system (Fig. 1) is that there is a single coherent temporal representation of all perceptual and motor relations. That is, both the commands issued by the user, as well as the sensory values from the propioceptors (e.g. selected joint angles) and exteroceptors (vision, distance sensors) are to be represented in a single temporal sequence of Boolean values. This transformation of continuous sensor values from a world model into a symbolic representation of logical predicates in a situation model corresponds to a form of conceptualization (Siskind 2001 [8], Steels & Baillie 2003 [9], Dominey & Boucher 2005 [2]).

Concretely, during the experiments, the "situation modeler" sends messages to the model whenever the robot detects a new event, i.e. a sensorimotor relation change (RC) (fig. 1.B). These Boolean relation change RC values (see Tab. 1) are communicated in an XML format. Based on the received RCs the model generates the world vectors (WVs) that are the vectors of all relation values. The WVs and RCs are stored in a chronological History that can be compared with a sensori-motor memory. For greater clarity, the History and its subsequences can be represented as directed acyclic graphs (Sect. 2.2) or as chronongrams (Sect. 2.3). During the first learning trial of a complex behavior, the user guides

[2] http://www.urbiforge.com

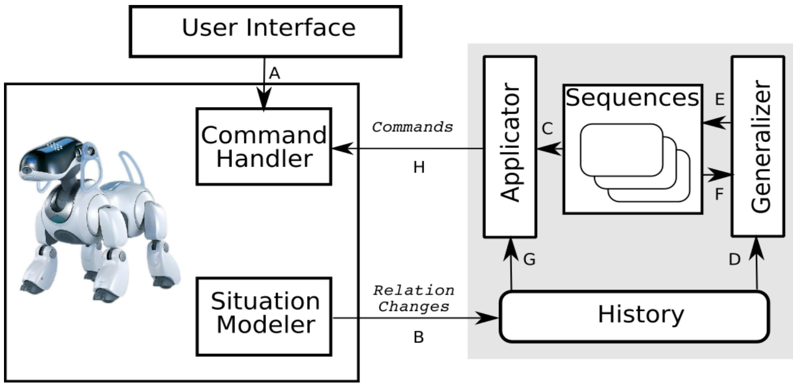

Fig. 1. System Architecture: The User Interface sends action command messages to the robot (A). The primitive actions are of the type `command`, and are implemented in the robot. Actions to be learned are of the type `complex` and are represented as sequences. The Situation Modeler sends RCs to the History (B) whenever the robot observes any change in a relation. When the behavior to be learned is completed (complex relation deactivated) the Generalizer extracts the last sequence from the History (D) and compares it with the corresponding sequence in the repertoire (F). The resulting generalized sequence replaces the previous one in the repertoire (E). When the user executes a learned behavior (complex relation activated) the Applicator executes (G). It retrieves the corresponding sequence in the repertoire (C) and sends the corresponding commands (H) if their preconditions are met.

Table 1. Actionnal and perceptual relations

Relation type	**Meaning**	
Action: When the relation is true, the command or the complex is going on. E.g. in Fig. 9, the robot is walking WV(3-4), is tracking WV(2-5) and is approaching WV(1-6).	**Command**	
	March(robot,front)	Walking forward
	March(robot,back)	Walking backward
	Rotate(robot,left)	Turning left
	Rotate(robot,right)	Turning right
	Track(robot,ball)	Tracking ball with head
	Complex	
	Approach(robot,ball)	Approaching the ball
	Align_right(robot,ball)	Aligning to the ball
	Turn_around(robot)	Turning around an obstacle
Perception: When the relation is true, the robot is perceiving. E.g. in Fig. 9, the robot is near something WV(4-6) and always see the ball.	**Exteroception**	
	See(robot,ball)	The robot is seeing the ball
	Near(robot,thing)	The robot is close to an object
	Proprioception	
	Neck(robot,left)	neck/head turned left ($\pm$ 10°)
	Neck(robot,right)	neck/head turned right ($\pm$ 10°)
	Neck(robot,center)	neck/head turned center ($\pm$ 10°)
	Touch(robot,shoulders)	shoulder button pressed

the robot through each step (fig. 1.A). When the complex command is issued, the Applicator is activated (fig. 1.G) but has no sequence to apply (fig. 1.C). Once the first learning trial is completed, the Generalizer extracts the sequence (fig. 1.D) from the History to include it in the sequence repertoire (fig. 1.E). For the next learning trial, the Applicator now finds the corresponding sequence in the repertoire (fig. 1.C) and attempts to apply it. Thus, while reading the sequence, when the preconditions for an action are met, the Applicator issues the appropriate command (fig. 1.H). After the second learning trial, the Generalizer extracts the just executed sequence from the History and compares it with the corresponding sequence in the repertoire (fig. 1.E). Sect. 2.2 and Sect. 2.3 respectively explain the roles of the Applicator and the Generalizer.

2.2 Generalization

The Generalizer performs generalization on a complex behavior that has just been executed and the existing generalized sequence for that behavior (respectively the sequence extracted from the History Fig. 1.D and the existing sequence

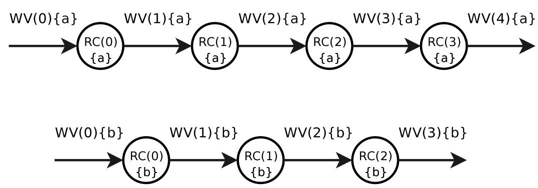

Fig. 2. Two complex behavior learning trials. The first trial yields 4 RC nodes and 5 WV arcs, `RC(0-3){a}` and `WV(0-4){a}` respectively. The second trial is identified by {b}.

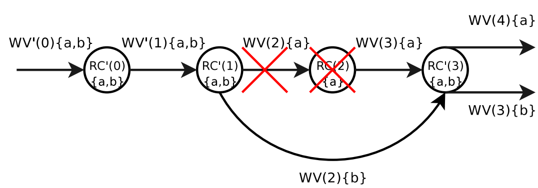

Fig. 3. WV-RC Fusion & deletion: `RC'(0){a,b}=RC(0){a}=RC(0){b}` `RC'(1){a,b}=RC(1){a}=RC(1){b}` `RC'(2){a,b}=RC(2){a}=RC(2){b}` `RC'(3){a,b}=RC(3){a}=RC(2){b}` `WV'(0){a,b}=WV(0){a}=WV(0){b}` `WV'(1){a,b}=WV(1){a}=WV(1){b}`

in the repertoire Fig. 1.F). Fig. 2 represents these as graphs. Generalization serves to determine which are the pertinent components in this command sequence pair. For this we apply two operations. The first is to merge identical WVs and RCs and remove superfluous loops (Fig. 3). The second operation determines the sufficient conditions for sending a command or receiving a perception (Fig. 4 and Fig. 5). This requires comaring the WVs preceding a given RC and identifying relations whose values can indifferently be true or false.

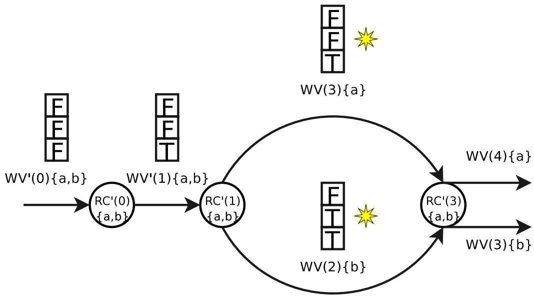

Fig. 4. Relation variability: The rectangles of 3 slots correspond to three arbitrary relations making up a WV. F and T correspond to False and True relations, respectively. We see that `WV(3){a}` and `WV(2){b}` lead to the same `RC'(3){a,b}`.

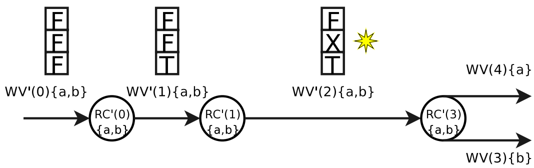

Fig. 5. Vectorial generalization: The new generalized vector`WV'(2){a,b}` is equal to `vect_gene(WV(3){a},WV(2){b})`. If the value of a given relation in the WV can be either True or False, this is designated by an X.

2.3 Application

After an initial nave learning (where the user leads the robot through each stage) the user can then ask the robot to perform the learned behavior. This process of re-execution of a learned sequence (stored in the generalized sequence repertoire) is called "application". During generalized sequence application, if the robot is

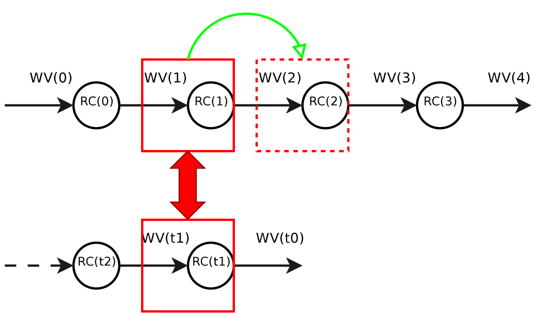

Fig. 6. Perceptive RC reception: If the current `(WV(1), RC(1))` pair of the generalized sequence matches with the last `(WV(t1), RC(t1))` pair of the History then the reading index is incremented. Concretely, the robot waits to be in the same state (`WV(1)` matches `(WV(t1)`) and to perceive the same state change (`RC(1) = RC(t1)`) in order to proceed to the next step.

in identical conditions (same WVs) then it can execute the next action in the sequence (sending a command Fig. 8). However, if not, the robot waits for these conditions (receipt of an appropriate RC Fig. 6) and thus remains static. The user can thus force the execution by explicitly commanding the robot if necessary (reception of an action RC Fig. 7).

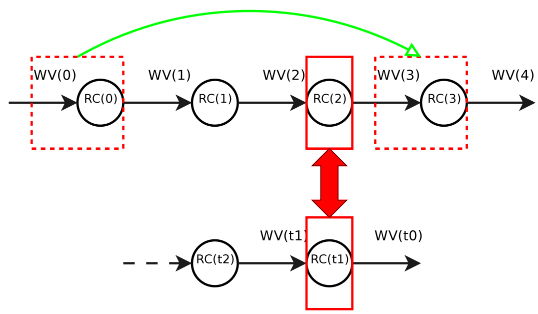

Fig. 7. Command RC reception: The reading head is positioned on the pair `(WV(0), RC(0))` though the received command `RC(t1)` matches the next command in the sequence `RC(2)`. The user has thus forced the robot. To continue the sequence execution correctly, the reading head is repositioned after the command matching that in the History.

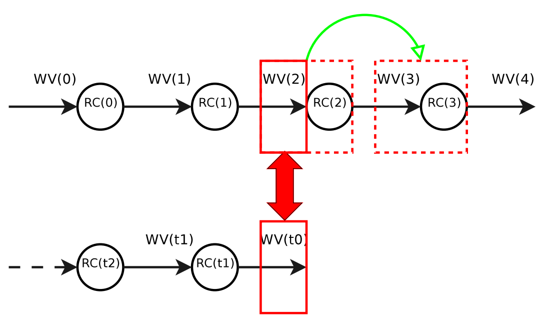

Fig. 8. Command RC emission: If the last History vector (`WV(t0)`) matches the current sequence vector (`WV(2)`) and `RC(2)` is a command, then the reading head is advanced. This incrementing occurs, of course, after the command is sent to the robot by the model.

3 Experimental Results

We can now present generalization results from three experiments that test the ability of the system to learn generalized representation of three distinct complex sensory-motor behaviors. In all cases, the human instructs the robot the first run through, and then the robot begins to generalize and attempts to execute the generalized sequence, with the intervention of the user when necessary.

3.1 "Approach" Behavior

The first experiment tests the ability of the system to learn to terminate an ongoing motor behavior based on a change in a perceptual state that occurs as a function of the execution of that behavior. This is a basic function of the generalized learning capability that allows the system to acquire behaviors that depend on generalized sensory-motor correlations, rather than on exact identical initial conditions.

For the behavior in question, the user places the red ball in front of the AIBO. The user then commands the AIBO to perform the complex action `approach`. At this time there is no Generalized Sequence stored for this behavior and so the AIBO does nothing. The user then commands the AIBO to visually track or look at the ball, and then to start walking. When the AIBO gets close to the ball, the user commands the AIBO to stop walking.

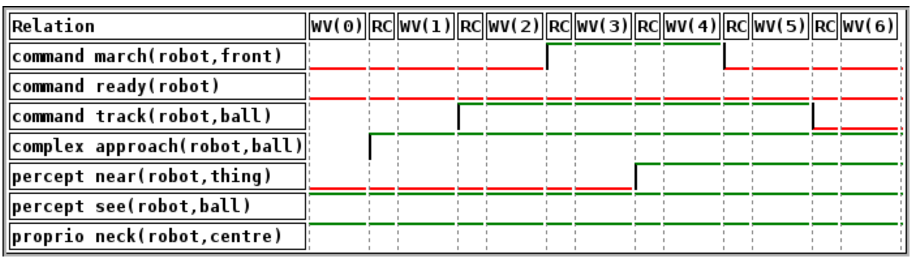

Fig. 9. Chronogram of World Vectors and Relation Change states generated after two learning trials of the complex behavior `approach(robot, ball)`. Note that distances along the horizontal time axis do not correspond to actual durations, though temporal order is accurately indicated. The user issues the `approach` command. This establishes the necessary preconditions to automatically initiate tracking the ball, which then establishes the preconditions to initiate the `march` command. The robot proceeds in this manner until the perceptual relation `near` changes to true. This establishes the preconditions for issuing the command to stop marching, and to stop tracking. At this point the learned behavior has successfully been performed in a fully autonomous mode by the system.

The crucial point is that there is a direct relation between the initial distance from the AIBO to the ball, and the distance that the AIBO walks before the user commands it to stop. The desired generalization is that whatever this distance is, on future trials, the AIBO will learn when to stop. Inspection of Fig. 9 reveals that prior to the command to stop walking (top trace `command march(robot, front)` goes to False or zero) the value of the perceptual relation `near(robot, thing)` has changed to true. That is, the proximity sensor value has reached a threshold indicating that the robot is now in physical proximity with an object. This physical proximity detection - which occurred because the robot walked sufficiently close to the ball independent of how far it was initially - can thus serve as the perceptual signal to stop walking. Thus, the human subject puts the robot through the action, and the robot learns to link what it perceives with the required issuing of commands to initiate and terminate motor functions. A crucial aspect of this adaptive behavior mechanism is that the robot is continuously testing its "hypothesis" - the current Generalized Sequence for the command in question - and relying both on regularities in the stream of relations coming from the perceptual environment, and relations coming from commands issued by the human in order to correct or shape the ongoing behavior that is being learned.

3.2 "Align" Behavior

The second experiment also tests the ability of the system to learn to terminate an ongoing motor behavior based on a change in a perceptual state that occurs as a function of the execution of that behavior in a different context. In this case, the AIBO is to search for the ball, orient its head to the ball, and then - maintaining this orientation -turn its body so that the neck is straight, and the AIBO body is aligned with the ball. In this configuration, when the AIBO starts to walk, it will be pointed in the direction of the ball.

For this behavior the ball is placed a few meters away from the AIBO, away from the direction in which it is currently looking. In the current example this will be to the right of the AIBO. The user starts by invoking the command `align_right`. In the first demonstration, there is no Generalized sequence, and so the system waits for instructions. The user then commands the robot to `track` or locate and look at the ball. Once this is achieved, the user then commands the robot to rotate its body towards the right, while continuing to fixate the ball, thus bringing the body into alignment with the head. When this alignment is achieved, the user commands the robot to stop turning. The goal here is that in the general case, independent of how far the ball initially is to the right, the robot will turn, and stop turning when it is aligned with the ball.

Examination of Fig. 10 indicates that just prior to the changing of command `rotate(robot, right)` to false, there is a pair of state changes in which the value of the proprioceptive relation `neck(robot, center)` becomes true, and then the value of the proprioceptive relation `neck(robot, right)` becomes false.

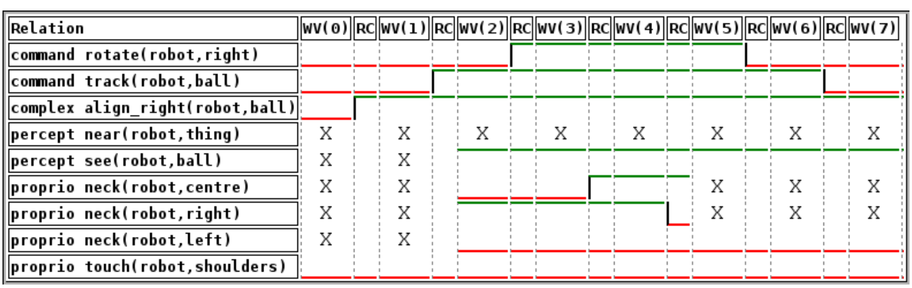

Fig. 10. Chronogram of World Vectors and Relation Change states in learning and generalizing the complex behavior `align_right(robot, ball)`. generalized sequence after 6 learning trials. See text.

Though not clearly indicated on the chronogram these two events occurred with a very short separating interval (<1 second). The next event is the user issuing the command to stop turning. Again, the crucial issue will be to automatically establish the link between the sending of the command to stop turning and the perceptual events corresponding to the centering of the head and neck with the body. If we compare the first history with the last generalied sequence, we can observe that there is a substantial shortening of the sequence is due to the fusion and deletion generalization (Fig. 3). The "X"s distributed throughout correspond to vectorial generalization illustrated in Fig. 5. The presence of an "X" indicates that the particular relation value can be either true or false at that point, and the generalized sequence can still proceed. This corresponds to relations whose values are not pertinent at the given time for the passage from the previous to the subsequent State.

Experiments 3.1 and 3.2 thus demonstrate that the generalization method is capable of determining the pertinent perceptual relation changes that trigger the onset and offset of motor commands. This provides generalized sequences that are sensory-motor programs that can be autonomously executed by the robot with a good deal of invariance to modifications in initial conditions.

3.3 "Turn Around" Behavior

The third and final experiment tests the ability of the system to learn to coordinate the initiation and termination of two distinct behaviors based on a succession of changes in perceptual states that occur as a function of the execution of these behaviors. In this case, the AIBO is to start walking and then stop when it detects a potential collision. It should then begin turning to the right, and stop turning when it sees the red ball.

Fig. 11 illustrates the generalized sequence for this behavior after 5 interactive training demonstrations. Examination of this generalized sequence provides a clear way to understand the successive elements of the behavior that were used to each this complex behavior. To begin training this behavior, the AIBO is placed a few meters from a wall, with the red ball placed about 1 meter to the right of the future collision point. The user starts by invoking the command `turn_around`. In the first demonstration, there is no Generalized sequence, and so the system waits for instructions. The user then commands the robot to center its head/neck in the forward direction. Once this is achieved, the user then commands the robot to begin to march forward. When the user sees that the robot is getting close to the wall, the user commands the robot to stop marching. Again, the goal here is that in the general case, independent of how far the robot initially is from an obstacle, the robot will stop walking when it comes within some threshold distance of that object.

In Fig. 11 this corresponds to the State change in which the perceptual relation `near(robot, thing)` becomes true. Indeed, this occurs just before the user-issued command to stop marching. Thus in the generalized sequence, this state change must precede the self generated command to stop marching. After this command, the next event is the user issuing the command to begin turning right.

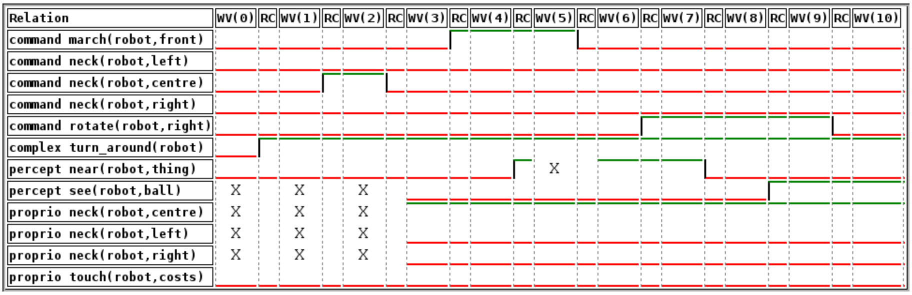

Fig. 11. Chronogram of World Vectors and Relation Change states after generalizing the complex behavior "turn_around"; generalized sequence after 5 learning trials

Again, a crucial issue will be to automatically establish the link between the sending of this command to turn right, and the enabling pre-conditions. The robot continues turning right until the next sequence of relation events which are the value of near(robot, thing) changing to false, followed by the relation `see(robot, ball)` changing to true. Fig. 11 thus represents the combined sequence of perceptual events and command events that define the Generalized Sequence for the complex behavior `turn_around`. This generalized sequence is used by the system as a behavioral program that allows the successive selection of the appropriate motor behaviors based on the combination of perceptual and motor relations that define the preconditions for these successive motor commands.

4 Discussion

The current research provides new results from experiments with a robotic system that can acquire an open set of new behaviors through interaction with the sensory-motor milieu and a human teacher. Part of the novelty of the system is that it combines perception and action into a coherent representation in terms of a temporal sequence of perception and action state changes. This combined sequence is then the object of a generalization process whereby, through multiple demonstrations of a given behavior to be learned, the system extracts the minimal pertinent description of the relevant actions and their required preconditions. This resulting generalized sequence is thus an integrated sensory-motor program that the system can execute autonomously. In three experiments based on complex behaviors of varying complexity, we have demonstrated that the system indeed learns to generalize the behavioral sensory-motor sequences, and that it can perform this for arbitrary combinations of its base set of sensory and motor relations (illustrated in Tab. 1). Another novel aspect of the current approach is that it is not at all tied to the specifics of the robot platform used. It only requires that the robot has the equivalent of the situation modeler which provides logical predicate values for different sensor and motor command status values. Indeed, part of the long term aim of this research is to demonstrate that this Generalized Sequencing Model can easily adapt to a variety of robot platforms.

This provides the basis for an interesting set of extensions. In this context we are currently extending the system to allow embedding of these complex behaviors to create hierarchical complex sequences. A second line of extension is related to the command of the system via spoken language. In the current context, user commands can be issued via a console interface and by spoken language using the CSLU RAD system in which single spoken commands are recognized and used to send the appropriate URBI command to the robot. Our future work in this area will allow the use of predicate-argument commands (such as "get the X" where X can take different arguments like ball, bone, etc.) and corresponding grammatical constructions, rather than single words. We have begun to investigate this use of grammatical constructions in Dominey & Boucher 2005 [2], and Dominey et al. 2005 [3].

Acknowledgements

J-DB is supported by a Doctoral Fellowship from the Rhone Alps Region as part of the Cluster Project "Presence." PFD is supported in part by funding from LAFMI, ACI TTT and the ESF ECRP. We thank Gerard Bailly, James Crowley and Jean Caelen for constructive comments on an earlier version of this research.

References

1. Billard A, Mataric MJ (2001) Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture. Robotics and Autonomous Systems, 37, 145-160.
2. Dominey PF, Boucher JD (2005) Learning To Talk About Events From Narrated Video in the Construction Grammar Framework, Artificial Intelligence, 167 (2005) 31-61
3. Dominey PF, Alvarez M, Gao B, Jeambrun M, Weitzenfeld A, Medrano A (2005): Robot Command, Interrogation and Teaching via Social Interaction, Proc. IEEE Conf. On Humanoid Robotics 2005.
4. Kaplan F, Oudeyer P-Y, Kubunyi E, Miklosi A (2005) Robot clicker training, Robotics and Autonomous SystemsMarch 2002, pp. 197-206(10)
5. Nicolescu M.N., Matari M.J. (2003): Natural Methods for Robot Task Learning, Proc. AAMAS'03.
6. Nicolescu M.N., Mataric M.J. (2001): Learning and Interacting in Human-Robot Domains, IEEE Trans. Sys. Man Cybernetics B, 31(5) 419-430.
7. Saunders J, Nehaniv CL, Dautenhahn K (2006) Teaching robots by moulding behavior and scaffolding the environment, Proceedings of HRI'06, March 2-4, Salt Lake City, Utah USA
8. Siskind JM (2001): Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. Journal of AI Research (15) 31-90
9. Steels, L. and Baillie, JC. (2003): Shared Grounding of Event Descriptions by Autonomous Robots. Robotics and Autonomous Systems, 43(2-3):163–173. 2002
10. Zöllner R., Asfour T., Dillman R.: Programming by Demonstration: Dual-Arm Manipulation Tasks for Humanoid Robots. Proc IEEE/RSJ Intern. Conf on Intelligent Robots and systems (IROS 2004).