

Imitation Learning and Response Facilitation in Embodied Agents

Stefan Kopp and Olaf Graeser

Artificial Intelligence Group, University of Bielefeld
P.O.Box 100131, D-33501 Bielefeld, Germany
{skopp, ograeser}@techfak.uni-bielefeld.de

Abstract. Imitation is supposedly a fundamental mechanism for humans to learn new actions and to gain knowledge about another's intentions. The basis of this behavior seems to be a direct influencing of the motor system by the perceptual system, affording fast, selective enhancement of a motor response already in the repertoire (*response facilitation*) as well as learning and delayed reproduction of new actions (*true imitation*). In this paper, we present an approach to attain these capabilities in virtual embodied agents. Building upon a computational motor control model, our approach connects visual representations of observed hand and arm movements to graph-based representations of motor commands. Forward and inverse models are employed to allow for both fast mimicking responses as well as imitation learning.

1 Introduction

Human children and adults effortlessly mimic and imitate others. They do so in a variety of situations and different types of imitative behavior can be distinguished [6]. *Stimulus enhancement* refers to the increased probability to act upon an object, on which another individual has acted in a similar way before. *Response facilitation* is the selective enhancement of a motor response already in the repertoire simply by seeing a conspecific performing an action. Finally, (*true*) *imitation* refers to the case in which an action that has not been part of the own repertoire before is learnt by observing others and can be reproduced with a possible time delay. One can observe all of these kinds of behavior already in human infants [12]. Yet, the question why we imitate each other so often is hard to answer. Many rationales or purposes have been put forward, among them the acquisition of new behaviors, the realization and signaling of the fact that another one is like me [11], or the mutual alignment and convergence in social interactions [13]. In addition, imitative behaviors such as mimicking gestures were suspected to gain knowledge about the intentions of conversational partners [4].

One central mechanism that seems to underlie imitative behavior is the selective influencing of the motor system by the perceptual system. These links open the possibility to equate an observed action with one's own actions through an implicit form of simulating them in the motor system. With the finding of 'mirror regions' with neurons that respond to both self-generated actions and the actions of others [8],

research in neuroscience has started to discover properties of our brain that may provide explanation for this. Motor areas seem to start to selectively *resonate* as soon as an appropriate visual stimulus is present. This provides functionally equivalent representations of both planned actions and observed actions. The motor system may thus play a central role not only in controlling bodily actions, but also in understanding behaviors of others, reaching into higher cognitive levels of operation and of social interaction and embodied communication [14].

The motivation of this work originates from the question whether it is possible (and helpful) to endow virtual humanoid agents with this ability, namely, to exploit their own motor system for understanding or adapting to other embodied agents in social interaction—including human users. In other words: can an embodied agent utilize his knowledge about the own body, the motor actions he is capable of with this body, and the utility of these actions for pursuing goals, to arrive at a better and faster understanding of the goals, needs, or affective states behind an interlocutor’s bodily actions?

In this paper we address two important questions that must be answered prior to this: How can an agent be enabled to pick up and adapt to a human-like repertoire of motor behavior from observing others, and how can such a repertoire and its automatic activation be coupled to perceptual processes? In terms of imitative behavior we thus try to model capabilities of response facilitation and true imitation in embodied agents. As a side effect, making embodied agents capable of true imitation promises to provide a new means of building up natural behavior lexicons, namely, by interactively teaching agents how to employ their own motor resources for demonstrated behaviors. This exceeds the usual use of motion capture techniques to collect low-level animation data that must be mapped onto the agent’s body and edited every time to fit particular context conditions.

We start with discussing related work in the next section. In Sect. 3, we describe the computational model of motor control for hand-arm movements in animated humanoid agents that we start out from. To study the modeling of imitative behavior we employ a scenario with two virtual humanoid agents, *Max* and *Moritz*. *Moritz* acts as the demonstrator and can perform arbitrary movements by executing predefined keyframe animations, whereas *Max* takes on the role of the learner and imitator. Sect. 4 presents our approach to connect the agent’s perceptual representation with his motor representations via forward and inverse models in order to enable imitation. Examples of the imitation learning as well as fast mimicking responses that are possible with this model are given in Sect. 5.

2 Related Work

A number of researchers have addressed the imitation learning of movements by artificial systems. Billard et al. [2] developed a biologically inspired, connectionist architecture that simulates mirror neurons as link between visual and motor representations. The model employs a hierarchy of neural networks to remodel the primates brain’s visuo-motor pathways. In this sense, it includes a temporal cortex for movement recognition, a pre-motor cortex for activity transfer into primary motor nodes, and a cerebellum for sequences of movements. Learning of motor sequences is

achieved by Hebbian update of the connections between these components. In a more recent approach, Billard et al. [3] propose a model to probabilistically learn the relevant features for imitation of an object manipulation task. The resulting strategies range from simply reproducing a goal (i.e. hitting the object) to reproducing the exact movement, the generation of which includes calculating and interpolating target points by means of inverse kinematics or by Hidden Markov Models representing a sequence of joint angles. Other computational studies of motor control suggest explicit internal models that predict the sensory consequences of a motor command (called *forward models* as they model the causal relationship between actions and their consequences) and that are employed to functionally implement *inverse models* (mapping from sensor stimulus onto efferent motor signals). For example, Amit & Mataric [1] use motor primitives both to detect and execute movements. This architecture combines base primitives that encode forward models for classes of movements and, additionally, serve as an inverse model able to determine parameters of an observed instance of that movement type. During demonstration, such parameters are passed on to higher levels for learning specific instances of a movement and even complex, composite movement sequences using Hidden Markov Models. Wolpert et al. [15] proposed the MOSAIC architecture of paired forward and inverse models (predictors and controllers, respectively). The idea is that the brain simultaneously runs multiple forward models to predict the consequences of motor behaviors in order to determine the dynamics of the body when interacting with objects. The MOSAIC model is able to learn a set of predictors that cover best experienced behaviors, and it ensures that the corresponding controllers are used in the right context. Applied for imitation, this model tries to predict the next state of the demonstrator's body (positions of limbs or joint angles). By comparing the predictions with the demonstrator's next state, the controller with the smallest prediction error is activated to mimic the seen movement.

Demiris & Hayes [7] distinguish between passive and active imitation. In the former, the imitator runs a "perceive-recognize-reproduce" cycle and the motor systems are involved only during the "reproduce" phase; in the latter, the imitator's motor systems are actively involved in perception, again, by comparing predictions of the available forward models with the observed movement. If none of the forward model matches, significant postures will be extracted via passive imitation to form a new behavior that is then added to the imitator's repertoire.

Most recently, Buchsbaum & Blumberg [5] have presented a graph-based approach for motor planning and a hierarchical action-trigger-model to account for the meaning of movements. Nodes in their graph represent poses, movements are transitions between them, with special nodes for predefined rest postures. After the complete observation of a movement the imitator tries to map an observed movement onto a path through the pose graph. That is, this model does not account for response facilitation, i.e. imitation in overlap with the demonstration. The meaning of a movement is inferred by connecting every action (e.g. eating) in the pose graph to a trigger (e.g. being hungry) referring to a goal. The goals and their triggers are organized in a hierarchical fashion and may be directed towards an optional object (e.g. grasping is a sub-goal of eating and is transitive, i.e. directed to an object). This representation allows for inferring from an observed action to the goal structure behind it or to properties of the object acted upon.

In our approach (see Sect. 4), we have combined and extended aspects from several of the models discussed here, notably, active and passive imitation, coupled forward and inverse models, and graph-based representations.

3 Computational Motor Control

In previous work we have developed an incremental model for the realization of multimodal utterances in embodied conversational agents [9]. A part of this model is a component for the on-the-fly creation of coverbal gestures animations, which tries to achieve a high degree of control and flexibility w.r.t. shape and time properties, while at the same time ensuring naturalness of the movement. A special motor planner accomplishes this task. It receives a set of timed morphological features of a gesture, e.g. hand shapes or arm movement specifications, and seeks a solution to drive the kinematic skeleton of the humanoid agent properly. To this end, a functional-anatomical decomposition of motor control is adopted. Specialized motor planning modules for the hands, the wrists, the arms, as well as the neck and the eyes instantiate *local motor programs* (LMPs) that autonomously control submovements, i.e. movements within a limited set of degrees of freedom (DOF) and over a designated period of time (see right part of Fig. 1). To combine the LMPs for a motor planning modules with a representation of control signals for the target submovement (left part of Fig. 1). When activating itself at execution time, every LMP turns this representation into a kinematic controller for the corresponding submovement.

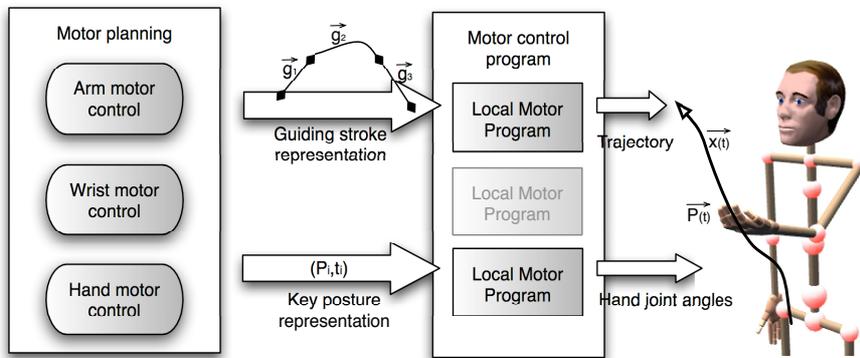


Fig. 1. Overview of the motor control and representation model

Since human arm movement exhibits external kinematic regularities, and since gestures often must reproduce external form features, arm movement trajectories are created directly in working space, as opposed to planning and controlling them in joint angles or torques. Our approach to forming wrist trajectories relies on well-known observations on complex arm movements [11]. Motor control for such

trajectories is incrementally exerted so as to successive segments are performed more or less “ballistically”. Due to skeleton structure and muscle dynamics, each segment is more or less straight or curvilinear (either C- or S-shaped), stays within a fixed movement plane, exhibits a symmetrical bell-shaped velocity profile, a quasi-linear relation between amplitude and peak velocity, and an approximate logarithmic relation between its extent and its duration. The segmentation corresponds to the points of maximum curvature or change of the movement plane, at which movement speed usually drops. Relying on these assumptions, both path and kinematics of an arm movement can be defined locally at these break points and in terms of a small set of control parameters. Consequently, in our model, the motor command representation for arm movement consists of a sequence of linear or curvilinear *guiding strokes*, which are concatenated to form the desired trajectory. Each guiding stroke bridges from one break point to the next by stating the target position, the time to get there, the basic shape of the trajectory, the moment of maximum velocity, and a velocity gain factor for movement at the end point. For curvilinear guiding strokes, the normal vector of the movement plane as well as the overall form (left/right C, left/right S) must be defined. In addition, the form of the curvilinear segment can be specified by the degree of curvature (from nearly straight to semicircle), the roundness (from nearly rectangular to nearly triangular), and the skewness (flattened toward the beginning or the end).

During motor planning, an LMP is created with such a sequence of guiding strokes. At execution time, this LMP continually estimates how long a hypothetical preparatory movement from the current hand location to the required start position of the first guiding stroke would take and decides whether it’s time to activate itself. Once activated, the LMP forms a trajectory by (1) inserting a first preparatory guiding stroke, (2) checking for collisions with the torso and inserting, if necessary, circumventing guiding strokes, (3) setting up all position constraints, (4) estimating the velocities at interior segmentation points, and (5) calculating a parametric spline curve that satisfies the spatio-temporal gesture features, while reproducing a naturally segmented velocity profile (for details see [9]). In result, the LMP possesses a controller that defines for each point in time the target position of the wrist in space. The MCP then solves inverse kinematics for the 7 DOF arm. The arm's redundancy is interpreted as swivel angle of the elbow about the shoulder-wrist axis and is either controlled by a dedicated LMP or is heuristically determined from the target wrist position, the longitudinal axis of the hand, a tendency to minimize wrist bending, and damping for low arm elevations. The solution arm configuration is calculated by selecting, for the 3-DOF shoulder and wrist joint, an Euler angle set that satisfies the twist limits at current joint altitude and elevation. Finally, other LMPs that directly affect joint angles influence the posture by overriding the respective set of angles. For example, as illustrated in Fig. 1, a LMP for hand-internal movement may have been created from a sequence of key hand postures along with parameters specifying the movement dynamics along the transitions and, after completing itself, directly sets the joint angles of the fingers.

4 Imitation Learning and Response Facilitation

The motor control model described in the previous section provides two levels of representation of a movement: First, the overt movement can be represented in terms of the wrist trajectory through space, the wrist orientation vectors, and the hand configurations. These features are directly accessible to visual observation and can be considered a *perceptual representation* of an observed movement. Secondly, when producing a movement, the agent needs to construct a structured *motor representation*. It consists of a sequence of motor commands that determine how the self-organizing, peripheral motor resources in the body (the LMPs) will be producing movement, and with which degree of control or autonomy. As described above, each motor command lays down the initiation of a ballistically executed movement segment corresponding to a motor “impulse”. Depending on the motor systems involved, a movement segment representation can be a guiding stroke that encapsulates control parameters for arm movement through space (goal position or configuration), for wrist movement (target orientation), or for hand-internal movement (target postures), each possibly along with a parameter for the manner of execution (time of maximum velocity).

Since the agent plans and stores actions in terms of the more abstract and more efficient motor representation, the first step in utilizing motor resources for perceptual and understanding processes is to realize the mapping from perceptual representations to motor representations, also allowing him to imitate the observed action. Most models assume a repertoire of motor actions on the observer’s part. In result, the mapping problem becomes one of selecting and activating the right action from this set, and we want this property from our model too. That is, if the agent disposes of a motor command for a movement *similar* to the observed one, the model should provide fast facilitation of an imitative response. If the sighted action is not in the agent’s repertoire, however, the model should also account for ways to learn new motor representations, thus allowing to continually extend the agent’s motor repertoire through learning by (true) imitation.

We approach this problem in a scenario where two virtual humanoid agents, *Max* and *Moritz*, are facing each other. *Moritz* acts as a demonstrator and can execute arbitrary animations, whereas *Max* takes on the role of the learner and imitator. That is, although he is equipped with general motor capabilities as to how a motor representation can be executed (see Sect. 3), *Max* has no knowledge about how to command his motor system, and hence how to control movements of his body in order to carry out certain actions. The goal, here, is to enable *Max* to acquire this knowledge just by watching *Moritz* making some movements.

4.1 Architectural Pre-requisites

Max is based upon a cognitive architecture that comprises a virtual view sensor for simulating his field of view, an ultra-short term sensory buffer for compensating sensor drop outs, and a perception component. The perception manages a perceptual memory by adding, modifying, and removing *percepts* as a result of virtual sensory stimulations. It can further apply simple and fast pattern detectors that can turn newly arisen constellations of percepts (e.g. a brown bushy region atop of a skin-colored elliptical region) into more complex percepts (e.g. *Moritz*’s head). When *Max* first

sees Moritz, his perception recognizes Moritz’s hands, wrists, and elbows and asserts percepts that contain the positions of the corresponding perceived body parts. All positions are coded in a body-centered frame of reference and are updated every time step. For simplicity, the hand percepts comprise information about the configurations of Moritz’s finger joints, i.e. we do not employ an inverse hand model yet.

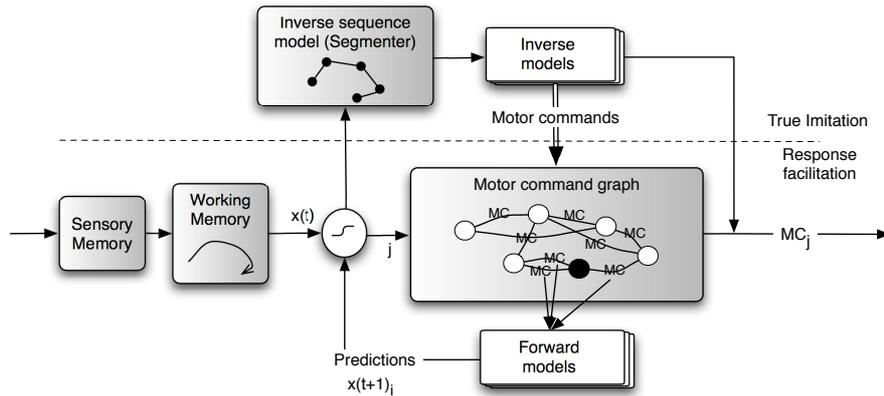


Fig. 2. The proposed model to map perceptual representations (in the working memory) to motor representations for imitative response. A motor command graph is used to context-sensitively create forward models that make predictions about the ongoing movement. The motor command that is predicting correctly and best (MC_j) is selected for execution. Via a more indirect route, inverse models are employed for imitation learning of new motor commands.

Our imitation model is shown in Fig. 2. A sensory memory of fixed size is always updated with the most recent percepts; older entries are deleted while new entries are added. This memory holds data for about only a tenth of a second and it is used to detect the onsets and ends of an observed movement. A visual working memory continually stores positions and directions of the wrist movement as well as the finger joint angles for the part of the movement that is currently under consideration. These perceptual representations are mapped onto motor representations, as described in the following sections.

4.2 Motor Command Graph

Building upon Buchsbaum & Blumberg’s approach [5] we employ a graph-based representation of movement. However, we extend it in twofold ways: First, we use a graph for representing not direct movement, but the motor commands that lead from one configuration of the motor system to another. We hence use the term *motor command graph* (see Fig. 2). Secondly, we employ multiple graphs at once in order to account for the possibly concurrent execution of submovements in different motor subsystems within a complex hand-arm-motion. Currently, we apply four motor command graphs for movement of the left and the right wrist and hands, respectively. Nodes in these graphs correspond to static motor system states, e.g. positions of the wrists in a body-centered frame of reference, or configurations of joint angles of one hand. At each point

in time, the agent is either at a certain state (henceforth, the *active* node) or on a movement that bridges between two states. Special *hub-nodes* in the graphs (cf. [5]) serve as representations of typical start states, positions and configurations in which a complete movement normally starts and ends¹. The edges between nodes represent single motor commands that the agent thinks can be applied in a certain state. For instance, edges in the wrist position graphs are guiding strokes that define single movement segments. Note that there can be multiple edges with the same start and end nodes, but for different movements in-between. Edges in the hand graphs represent parametric transition functions that define the duration and velocity profile of a movement from one hand configuration (start node) into another (end node).

4.3 Imitation Learning

In our imitation model, motor learning occurs whenever a demonstrated movement is new to Max, i.e., when no edge coming off the currently active state in the motor command graph can predict the incoming visual perceptions. If Max observes a completed but unknown movement, a two-stage inverse model is employed to construct appropriate motor commands from the percepts stored in the working memory. The first inverse model accomplishes segmentation, i.e. it is an inverse model of the sequencing of motor commands; the second one is an inverse model mapping single perceptual segments to candidate motor commands that may be used to produce them.

Arm movement is perceptually represented as wrist trajectory, given as a sequence of movement vectors calculated from each two successive sample positions. Each vector thus gives the direction of movement at that point in time as well as the speed (length of the vector divided by the time needed for this movement sample). Segmentation of arm movement is carried out based on both movement speed and direction. For segmentation based on *speed*, the velocity profile is searched for local minima that fall below a threshold relative to their neighboring maxima, i.e. they represent significant drops of movement speed. These points are taken as break points since two distinct guiding strokes will be needed to reproduce the observed velocity profile. Note that a minima with a considerably high speed compared to its surroundings would rather hint to a curved, yet single movement segment.

For the segmentation based on movement *direction* a divisive clustering method is employed. In the first step, all movement vectors are normalized to a length of one and are collapsed into one cluster. The average element of this cluster, the vector pointing into the gross movement direction, is then compared with every element of this cluster. If at least one element is significantly different, i.e. pointing into a significantly other direction, a new cluster with the direction of the most different element as average value is created. In that case, all clusters are cleared, except from their average values, and the movement vectors will again be distributed among the clusters based on their similarity with the average values. This procedure is iterated for every cluster until all elements in each cluster resemble the respective average value up to a certain threshold. Finally, after clustering, the affiliation of the movement vectors in working memory to a cluster is checked. A break point is set if two succeeding movement

¹ Note that the usage of hub-nodes presupposes that the imitator has recognized a structural congruency with the demonstrator, and that both can be in coincident bodily states.

vectors fall into different clusters, i.e. a significant change in movement direction compared to the overall curvature of the trajectory has been detected.

After segmentation, an inverse model is applied to each segment in order to calculate the parameters needed to produce it by a guiding stroke. To this end, the equations for trajectory formation (see [9]) are applied in inverse direction. A new edge is formed from the resultant guiding stroke and added to the arm motor command graph, leading from the currently active node to a newly inserted node that corresponds to the end state of this segment. After adding the new edge and the new node to the graph, the new motor command is executed to move the imitator's wrist into the new position, whose corresponding node now becomes the active node.

Likewise, if Max observes a completed, unknown finger movement by the demonstrator, the movement will be segmented both, at angular velocity minima and according to a clustering of movement directions, to detect the start, end, and via postures of the sighted hand movements. These configurations are used as key postures to create a new edge in the hand motor command graph, leading away from the currently active node into a new node corresponding to the new hand configuration. Upon adding the new edge and the new node to the movement graph, the according movement is executed and the new node becomes active.

The overall imitation process is illustrated as pseudo-code algorithm in Fig. 3. The described imitation learning occurs in the top part, whenever no motor command is available that could predict what is being observed. This process of learning new

```

do {
  edges ← motor commands that can predict the observed movement
  -- imitation learning --
  if (no edge found) {
    segment after velocity profile and by clustering direction vectors
    for each segment {
      motor command ← apply inverse model (segment)
      create and insert new edge and end node to graph
    }
    execute newly learned motor command sequence
    active node ← final node of sequence
  }
  -- response facilitation --
  else if (one edge predicts correctly and best) {
    execute motor command of the selected edge
    while (demonstration ongoing and imitative movement correct {
      keep on executing motor command
    }
    if (imitation deviates from demonstration) {
      stop imitative movement
      exclude edge from future predictions of this segment
      return to last correct motor state node
      active node ← last correct motor state node
    }
  }
  else active node ← end node of executed motor command
} while (true)

```

Fig. 3. The imitation process in pseudo-code

motor commands, and adding new edges and nodes to the motor command graph, respectively, is iterated as long as the demonstrator performs a movement. Complex movements are that way translated into sequences of motor commands that are stored in the graph representation as paths spanning several edges and nodes. Parts of the movement that are known already are readily incorporated into the newly learned motor command sequence.

4.4 Response Facilitation

The motor command graph constructed in this way can serve as basis for identifying, imitating, and even predicting the action of other's. In our approach, just like in previous work by others, forward models are employed to selectively activate motor knowledge of the agent that can successfully explain a demonstrated behavior (see Fig. 2). However, in our approach, these forward models are created dependent on the motor context, instead of testing all behaviors in the repertoire every time.

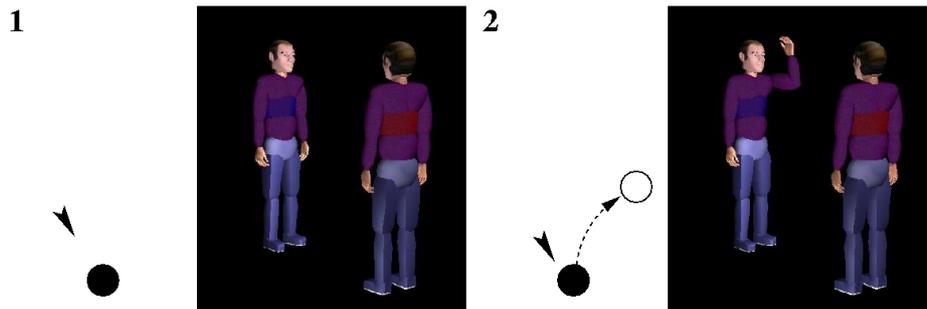
As illustrated in Fig. 3, this process facilitates immediate imitative responses, when active motor representations are continuously being executed and compared to the demonstration. If the demonstrator performs a movement, Max stores the movement directions in his working memory and compares all edges leading away from the current node instantaneously to the observation. That is, forward models of all motor commands that seem applicable are being computed in parallel to derive predictions of the possible future courses of movement. These predictions are compared to the observed movement. Every motor command that causes a prediction error below a certain threshold is considered a possible candidate to explain and reproduce the observed movement. If there is more than one candidate, this observation, prediction, and comparison cycle will go on until only one edge with a low prediction error prevails. This is the winning edge, which is used to construct a motor program to reproduce the observed movement. That is, Max starts to imitate Moritz immediately, possibly well before the demonstration has ended (active imitation or response facilitation). Note that, if the demonstrated movement is in principle familiar to Max but now performed with a different speed, the predictions of the corresponding forward model (or edge) will not be correct. That is, movements with different execution speeds would be learned as separate edges by the algorithm in Fig. 3. One solution to this problem is to map the length of the movement seen so far onto the trajectory predicted by a forward model, yielding the time parameter at which the corresponding motor command (with the speed demonstrated at first) would have produced a movement of the same length. From this, the speed difference can be calculated and used to re-parameterize the motor command.

While imitating, Max continually compares his movements with the observed ones. Since Max reproduces behavior with his very own motor control and body, the imitation may well differ from the demonstrated one. In our current approach, we take an imitation to be successful as long as it meets with a sufficient degree of accuracy the "movement goals" as far as captured by the motor representations, i.e. target positions/configurations and wrist trajectories. Demonstrations of, in this sense, successfully imitated movement segments are removed from the working memory. If, however, a difference between the observed and self-performed movements becomes too big, the execution of the movement will be interrupted and the imitator returns into the last node at which accordance with the demonstrated movement was found.

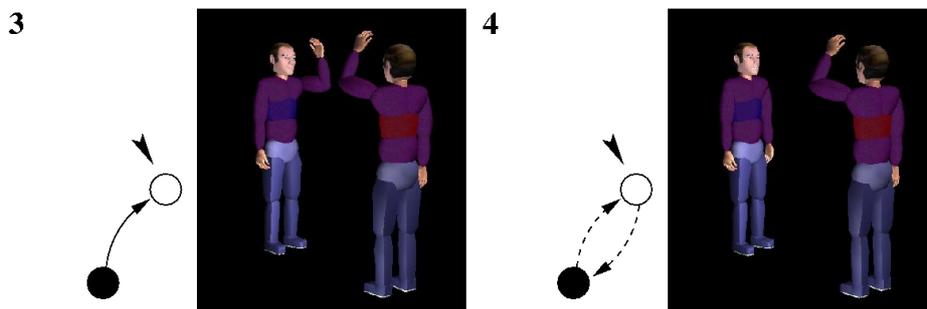
Since the chosen edge from this node on was not the right but the best fitting one, the perceived movement is new. In this case, the last part of the observed movement as stored in the imitator's working memory will be subjected to the inverse model for segmenting in order to learn and impart new edges to the motor command graphs.

5 Example

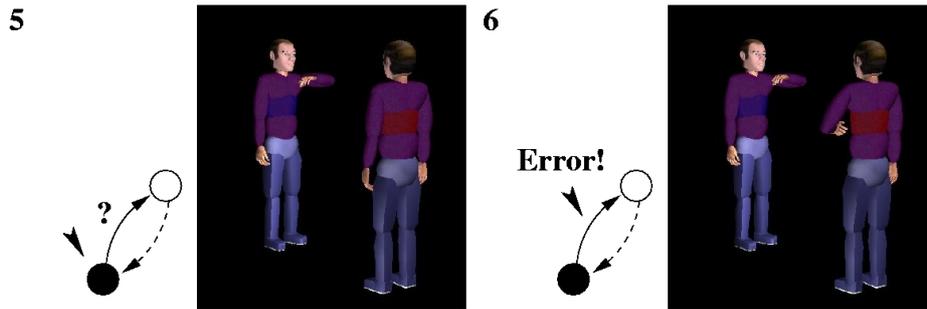
To demonstrate our model, let us consider a sequence of three successive imitation games between Max and Moritz. The following figures show snapshots of the virtual scene (Max is to the right) and illustrate how the motor command graph is being constructed and processed. In the beginning (1), Max and Moritz are standing face-to-face and in relaxed poses. Max is in his hub node (black circle; active node indicated by the arrowhead) and is assuming this posture corresponds to Moritz's current pose. Moritz starts demonstrating a left arm movement (2). Max observes the movement up to its end and then applies his inverse models to segment it (here, it consists of a single guiding stroke) and to derive a corresponding motor command that gets added as new edge to the graph (dashed arrow).



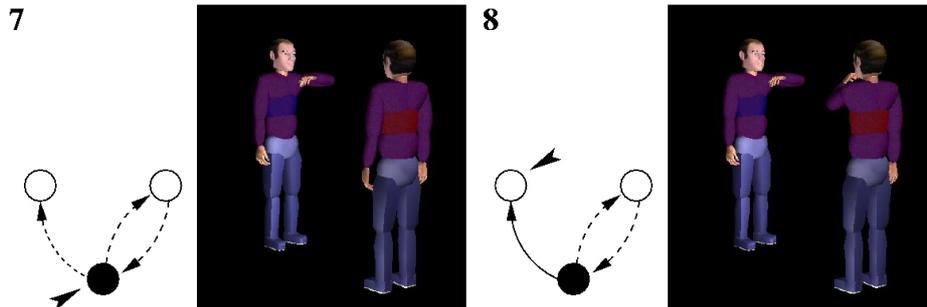
Performing a true imitation, Max carries out the newly acquired motor command and arrives in the new motor system state (3). Moritz moves his arm back to rest, observed by Max who learns a motor command for this movement segment as new edge leading back to the hub state (4), and can then move his arm accordingly.



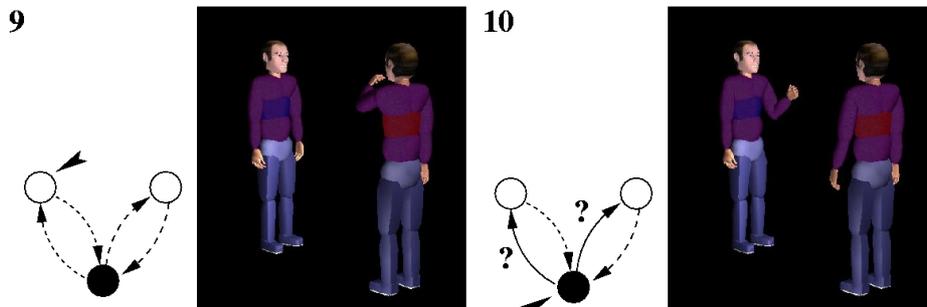
Moritz then starts to demonstrate another movement, starting out in the same direction as the previous one. Max tests whether any of his motor commands (only one at this time) matches (5) and finds that the previously learnt command successfully predicts the movement up to now. This facilitates an immediate imitation (6), until the error becomes so large as to rule out the motor command. Max thus stops his imitation.



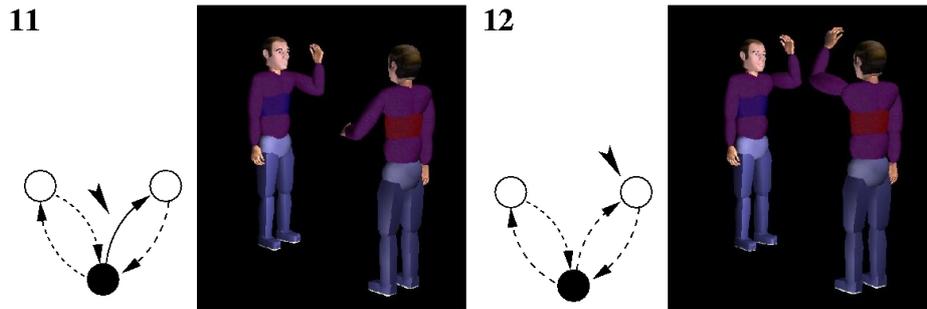
Returning to the last correct node, Max employs his inverse models to learn the new motor command and adds a corresponding new node and edge to the graph (7). He then imitates this new movement and arrives in a new active motor system state (8).



Seeing Moritz returning to the rest pose, Max learns the motor command for this movement too (9), and a new demonstration by Moritz starts (10). This time, Max finds two possible candidate edges to explain the observed movement.



Max thus remains in the current motor state (here, the hub node) until one of the candidates fails to predict the observed movement. At this time (11), the identification of a motor command triggers an immediate imitation (response facilitation), leading into a motor state that Max assumes identical to Moritz's (12).



6 Conclusion

In this paper, we have presented work towards embodied agents that dispose of rich motor control representations to control their body for goal-directed actions, and that can employ these motor resources also for understanding and predicting the behavior of other, structurally congruent individuals. We have described an imitation model that enables our agents to map from perceptual representations to motor representations, possibly finding its outlet in immediate mimicking, and to learn an unknown motor behavior through (true) imitation.

A lot of work remains to be done. For one thing, our agents learn and perform any movement that they perceive and are able to process using their inverse and forward models of motor control. Yet, it would be more plausible to let an agent run through a stage of “body babbling”, in which he can create a sense of his own body and its motor capabilities. We can now also imagine replacing the virtual demonstrator Moritz by a human demonstrator, using a tracking system for detecting the positions of the wrists and using data gloves for perceptions of the finger configurations. What needs to be done then is, among other things, an adjustment of the arm length and the body height to transform body-centric wrist positions of the demonstrator into according target wrist positions for the imitator. Finally, our model right now pertains to „meaningless“ imitation. A movement is being reproduced based on own motor capabilities, but without reconstructing and attaching meaning to it, in the sense of actions that are fulfilling a specific goal. The work by Buchsbaum & Blumberg (2005) has pointed into possible directions to tackle this problem. It may also be a venue for addressing the open problem of *complex* (true) imitation, i.e. learning and imitating sequences or hierarchies of motor action by employing structured symbolic representations—an approach we have previously used for imitating iconic gestures [10]. With these steps, for which a lot of work that remains to be done cannot even be foreseen at this point, we may eventually arrive at humanoid agents that are capable of modeling the similarities and differences between the own and others' internal states, as they arise out of and make their way into our bodies.

References

1. R. Amit, M.J. Mataric (2002). Learning Movement Sequences from Demonstration. Int. Conf. on Development and Learning, 302-306
2. A. Billard, M.J. Mataric (2001). Learning human arm movements by imitation: evaluation of a biologically inspired connectionist architecture. *Robotics & Autonomous systems* 37(2-3): 145-160
3. A. Billard, Y. Epars, S. Calinon, S. Schaal, G. Cheng (2004). Discovering optimal imitation strategies. *Robotics & Autonomous Systems*: 47(2-3): 69-77
4. Blakemore, S.-J., & Decety, J. (2001). From the perception of action to the understanding of intention. *Nature Reviews Neuroscience*, 2, 561–567.
5. D. Buchsbaum & B. Blumberg (2005). Imitation as a First Step to Social Learning in Synthetic Characters: A Graph-based Approach, ACM Symp. on Computer Animation, 9-18
6. Byrne, R.W. (1995). The thinking ape. Evolutionary origins of intelligence. Oxford University Press.
7. Y. Demiris, Gillian Hayes (2002). Imitation as a dual-route process featuring predictive and learning components: a biologically-plausible computational model. In: *Imitation in Animals and Artifacts*, Chapter 13, MIT Press.
8. Iacoboni, M., Woods, R. P., Brass, M., Bekkering, H., Mazziotta, J. C., & Rizzolatti, G. (1999). Cortical mechanisms of human imitation. *Science*, 286 (5449), 2526–2528.
9. S. Kopp, I. Wachsmuth (2004). Synthesizing multimodal utterances for conversational agents, *Computer Animation & Virtual Worlds*, 15(1): 39-52
10. S. Kopp, T. Sowa, I. Wachsmuth (2004). Imitation games with an artificial agents: From mimicking to understanding shape-related iconic gestures. In Camurri, Volpe (eds.): *Gesture-Based Communication in Human-Computer Interaction*, 436-447, Berlin: Springer-Verlag
11. Latash M.L. (1993) *Control of Human Movement*. Human Kinetics: Urbana, IL.
12. Meltzoff, A. N. (2005). Imitation and other minds: The "Like Me" hypothesis. In S. Hurley & N. Chater (Eds.), *Perspectives on imitation* (pp. 55-77). MIT Press.
13. Rizzolatti, G., Fogassi, L., & Gallese, V. (2001). Neurophysiological mechanisms underlying the understanding and imitation of action. *Nat. Rev. Neurosc.*, 2, 661–670.
14. Wallbott, H.G. (1995). Congruence, contagion, and motor-mimicry: Mutualities in nonverbal exchange. In I. Markova, C. Graumann, & K. Foppa (Eds.), *Mutualities in dialogue*. Cambridge University Press.
15. Wachsmuth, I. & Knoblich, G. (2005). Embodied communication in humans and machines - a research agenda. *Artificial Intelligence Review* 24(3-4): 517-522.
16. D.M. Wolpert, K. Doya, M. Kawato (2003). A unifying computational framework for motor control and social interaction, *Philos Trans R Soc Lond B Biol Sci.* 358(1431): 593-602