

Discovering Imitation Strategies through Categorization of Multi-Dimensional Data

Aude Billard, Yann Epars, Gordon Cheng
and Stefan Schaal

STI - I2S - ASL3, EPFL, 1015-Lausanne,
Switzerland

CS Dept, HNB-103, Univ. of Southern California
CNS/HRCN, ATR, 2-2 Hikaridai, Seika-cho,
Soraku-gun, 619-02 Kyoto, Japan

{Corresponding Author: aude.billard@epfl.ch}

Abstract—An essential problem of imitation is that of determining “what to imitate”, i.e. to determine which of the many features of the demonstration are relevant to the task and which should be reproduced. The strategy followed by the imitator can be modeled as a hierarchical optimization system, which minimizes the discrepancy between two multi-dimensional datasets. We consider imitation of a manipulation task. To classify across manipulation strategies, we apply a probabilistic analysis to data in Cartesian and joint spaces. We determine a general metric that optimizes the policy of task reproduction, following strategy determination. The model successfully discovers strategies in six different manipulation tasks and controls task reproduction by a full body humanoid robot.

I. INTRODUCTION

This work aims at developing a general policy to drive robot learning by imitation and robot programming through demonstration. It follows a trend of research that aims at defining a mathematical framework for imitation learning [11], [13].

Imitation learning needs to address the following three key questions: “what to imitate”, “how to imitate” and “when to imitate” [11]. Previous work has essentially focused on the question of “how to imitate”. The imitation mechanism was, then, aimed at a precise reproduction of a pre-specified sub-set of task features, such as hand-object actions (picking up a block, rotating the block), [9], [16], [15], [10], state-actions (turn left, move forward) [6], [7], the path followed by the manipulated object [2], [12] and the joint trajectories of the demonstrator’s motion [5], [8], which could be reproduced by pre-defined motor programs. The present work aims at complementing previous work and at addressing the question of “what to imitate”, by defining a general policy for learning which of the features of the task are relevant to the reproduction.

Recent work by (Alissandrakis et al., 2002)[1] illustrated nicely the problem of determining “what to imitate” in a chess world case-study, in which the imitator agent can follow either of three strategies, *end-point level*, *trajectory level*, *path level*, to reproduce either subparts

or the complete path followed by the demonstrator. We follow a similar taxonomy and apply it to the learning and reproduction of a manipulation task by a humanoid robot. We take the perspective that the features of the movements to imitate are those that appear most frequently, i.e. the invariants in time.

The model builds upon previous work [3], [4] and is composed of a hierarchical time delay neural network that extracts invariant features from a manipulation task performed by a human demonstrator. The system analyzes the Cartesian trajectories of the objects and the joint trajectories of the demonstrator’s arms. By comparing the probabilities of occurrence of the different events, the model determines whether the goals¹ of the manipulation task are: a) to move a specific object, b) to move the objects in a specific direction, c) to move the objects in a specific sequence, d) to perform a specific gesture. The observation of the manipulation task is then used to drive the reproduction of the task by a full body humanoid robot.

II. EXPERIMENTAL SET-UP

A typical imitation experiment consists first of the demonstrator performing the task with the robot observing and extracting the relevant steps of the demonstration. After a signal signifying the end of the demonstration, the robot reproduces only the part of the demonstration found significant. In the work presented here, the demonstrator performs a manipulation of three color boxes (green, pink and yellow) on a table stand, see Figure 1. The demonstrator performs five types of manipulation tasks, in which he either:

- 1) moves only a specific box, irrespective of the direction of movement or the hand used to move the box,
- 2) moves all boxes in a specific direction (along the x, y or z-axis)

¹The goals are not all mutually exclusive and the imitation can tend to satisfy several goals.

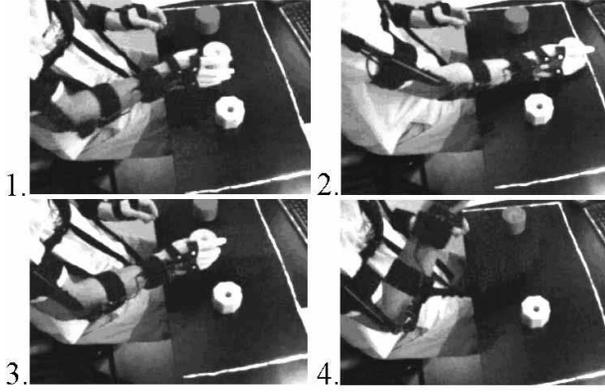


Fig. 1. Snapshots of a demonstration sequence. The task consists of moving 3 boxes with either the left or the right hand with a specific gesture: 1-2) first move the box along the x-direction (forward) for approximately 20 cm, then move the box along the z-direction (upward) for approximately 20 cm.

- 3) moves the boxes in a specific sequence (box1, box2, box3),
- 4) moves all boxes using the same hand-box relationship (use the left hand only, or use the hand closest to the target),
- 5) moves all boxes following a specific gesture, i.e. following a specific joint trajectory, irrespective of the location and orientation in space (e.g. clapping the hands before moving a box, drawing a circle in the air before touching the box), see Figure 1.

The demonstrator repeats each manipulation task five times, in order to make the interesting feature salient. The robot repeats only the invariant across the different demonstration and does not simply copy the whole sequence.

A. Data Recording

The Cartesian trajectories of the colored boxes are tracked by a fixed pair of cameras, see Figure 2. The system tracks at a rate of 60Hz the position and velocity of the three colored blobs. The trajectories of the blobs are segmented into 3 types of events: **hits**, i.e. a displacement of a box by the demonstrator/imitator, **directed hits**, i.e. a displacement of the box in a specific direction, **arm-choice**, i.e. the frequency of use of left and right arms, and **close-far**, the frequency of use of the hand closest to the box for performing a hit (based on a measure of the distance from the box to the left and right hand-side of the body). Arm-choice and close-far measurements were available only in simulation.

Joint displacements, i.e. angular measurements of the 14 degrees of freedom of left and right arms (flexion, abduction and humeral rotation of the shoulders and wrist, flexion of the elbows), are recorded by a SenSuiT exoskeleton equipped with an array of Hall-sensors (see Figure 1). Data are captured at the rate of 100Hz.

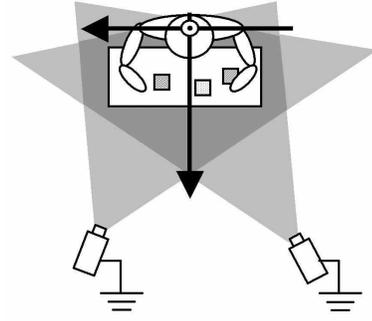


Fig. 2. A stereo colour vision system used for tracking motions of colour boxes during demonstration and imitation.

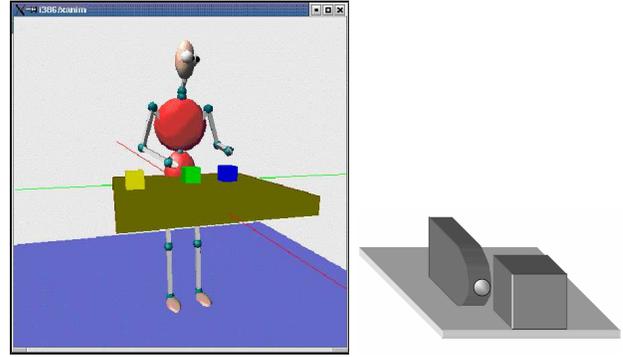


Fig. 3. **Left:** The Xanim simulator calculates the dynamics of a 30 degrees of freedom avatar and of 3 colored boxes [14]. **Right:** The end-effector of the robot is a solid plate, whose orientation in 3D can be modulated. In simulation and during control, the position of the end-effector is considered to be at the end of the plate (sphere in the drawing).

The trajectories of the joints are segmented to detect the starts and stops of voluntary motion and changes in direction of movement, following Equation 1. Since different degrees of freedom have different dynamic properties, due to their different lengths and muscular composition, we applied different segmentation parameters to each. Let $\tilde{\theta}_i(\Delta t_j)$ be the mean angular displacements of joint i during the time interval Δt_j . θ_i^M and θ_i^m are the maximal and minimal values of θ_i over the whole trial. A segmentation point is created at the crossing between the two intervals Δt_1 and Δt_2 , if:

$$\theta_j^m < |\tilde{\theta}_i(\Delta t_1) - \tilde{\theta}_i(\Delta t_2)| < \theta_j^M \quad (1)$$

a) Parameterization:

After segmentation, the speed and direction of movement of each joint is coded in the output of a neuron, see Section III-C.0.b. There are two neurons per degree of freedom (DOF), coding for positive and negative direction of movement, respectively. Let $y_i(t)$ be the output of neuron i at time t , and $t_{1,n}$ the series of time steps at which the segmentation has occurred, and $\tilde{\theta}_i(\Delta t_j)$ the

mean velocity of the joint i between two segmentation points, then:

$$y_i(t_j) = \tilde{\theta}_i(\Delta t_j) \quad (2)$$

B. Robot Control

The work was conducted first in the Xanim simulator and then implemented on DB, a 30 degrees of freedom (Head: 3, Arms 7 *2, Trunk 3, Legs 3*2, Eyes 4 D.O.F.) hydraulic humanoid robot, located at the Advanced Telecommunication Research institute. The Xanim simulator is a dynamic simulation of the DB robot (see Figure 3 right). The external force applied to each joint is gravity. Balance is handled by supporting the hips; ground contact is not modeled. There is no collision avoidance module. The dynamics model is derived from the Newton-Euler formulation of Rigid Body Dynamics.

The robot/avatar's movements are force-controlled based on desired trajectories (Inverse kinematics transforms a kinematic plan from cartesian to joint space). The trajectory to imitate is specified as a set of target points in Cartesian space. For instance, the action "pushing a box to the left" follows a trajectory made of three points, located 10cm to the right of the box, in the middle of the box and 10cm to the left of the box.

In observation of the gestures, the joint trajectories of the demonstrator are segmented following velocity crossings, as described in Equation 2. During reproduction, the joint velocities are fitted by a series of Gaussians, see example in Figure 8. If $y_i(t)$ is the output of neuron i associated to joint i , then the reproduced velocity $\theta'_i(t)$ between the segmentation points t_j and t_{j+1} is:

$$\theta'_i(t) = \exp\left(-\frac{(t - y_i(t))^2}{r^2}\right), t_j \leq t \leq t_{j+1} \quad (3)$$

$y_i(t)$ is constant over the time interval and is equal to $y_i(t_j)$, see Equation 2. r is a constant.

III. FORMALISM

A. Metric of Imitation

The imitation task can be decomposed in an *observation process* and a *reproduction process*. The observation process categorizes the dataset following *levels of imitation* $l = 1, \dots, L$ and *strategies*. For each level of imitation l , a set of strategies $s_l = s_{l_1}, \dots, s_{l_{S_l}}$ is associated. We define the imitation metric M as a weighted sum of level- and strategy-dependent metrics $M_{s_{l_j}}$.

$$M = \sum_{i=1}^L \sum_{j=1}^{S_i} w_{ij} \cdot M_{s_{i_j}} \quad (4)$$

The observation process determines the weights of the metric, according to the probability $P(s_{l_j})$ of observing

the imitation level l and the associated strategy s_{l_j} . The weight w_{l_j} associated to the strategy s_{l_j} is given by

$$w_{l_j} = \frac{P(s_{l_j})}{\sum_{i=0}^L \sum_{j=1}^S P(s_{i_j})} \quad (5)$$

The reproduction process determines the optimal control strategy s_l that produces a dataset $D' = \{X', \Theta'\}$, such that the metric $M(D, D')$ is minimal. If $P(s_{l_j})$ and $P'(s_{l_j})$ are the probabilities of having observed level l and the associated strategy s_{l_j} in the demonstration and the reproduction respectively, then the metric

$$M_{s_{l_j}} = |P(s_{l_j}) - P'(s_{l_j})| \quad (6)$$

Then, the reproduction process chooses the goal of the reproduction with a probability $P'(s_{l_j})$, such that $P'(s_{l_j}) = P(s_{l_j})$.

Probabilities are normalized such that $\sum_{i=0}^L \sum_{j=1}^S P(s_{i_j}) = 1$. The residual is the probability of seeing a random strategy:

$$P(s_0) = 1 - \sum_{i=1}^L \sum_{j=1}^S P(s_{i_j}) \quad (7)$$

B. Imitation Strategies

In the manipulation task considered here, the dataset $D = \{X, \Theta\}$ is composed of the trajectories of the manipulated objects, $X = \{x, \dot{x}, \ddot{x}\}$ (3-dim Cartesian position, speed and acceleration), and of the joint trajectories of the 14 degrees of freedom of the two arms: $\Theta = \{\theta, \dot{\theta}, \ddot{\theta}\}$ (angular position, speed and acceleration).

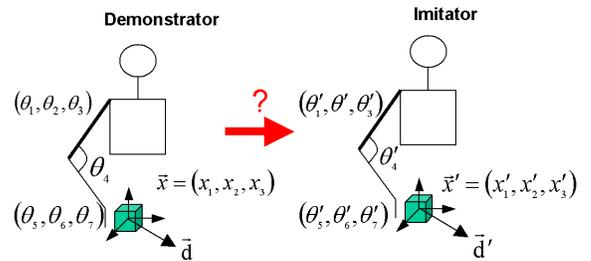


Fig. 4. Imitation of a manipulation task. The demonstrator's world is mapped into the imitator's world. Demonstrator and imitator's worlds are modeled as two dataset $D = \{X, \Theta\}$, $D' = \{X', \Theta'\}$.

We consider four levels of imitation l_1, \dots, l_4 , to which one or two strategies are associated:

- l_1 : *reproducing only the goal*, reaching for the same box (s_{1_1}) or for the same set of boxes in sequence (s_{1_2}), irrespective of the path followed by the box and the gesture.
- l_2 : *reproducing the path followed by the target*, i.e. moving the box in a specific direction (s_{2_1})

- l_3 : reproducing the same hand-object relationship, i.e. using only left or right hand (s_{3_1}), or using the hand closest to the target (s_{3_2}).
- l_4 : reproducing the exact gesture, i.e. reproducing the trajectory of the joints (s_{3_1}).

C. Strategy Determination

Let $h(i)$ with $i = \{1, 2, 3\}$, $d(i)$ with $i = \{1, \dots, 8\}$, $a(i)$ with $i = \{1, 2\}$, and $c(i)$ with $i = \{1, 2\}$ be the number of events **hits**, **directed hits**, **arm-choice** and **close-far** (see Section II-A for the definitions of the events). Then, we compute:

$$P(s_{1_1}) = \max_{i=\{1,2,3\}} \left(\frac{h(i)}{\sum_{j=1}^3 h(j)} \right) \quad (8)$$

$$P(s_{2_1}) = \max_{i=\{1, \dots, 8\}} \left(\frac{d(i)}{\sum_{j=1}^8 d(j)} \right) \quad (9)$$

$$P(s_{3_1}) = \max_{i=\{1,2\}} \left(\frac{a(i)}{\sum_{j=1}^2 a(j)} \right) \quad (10)$$

$$P(s_{3_2}) = \max_{i=\{1,2\}} \left(\frac{c(i)}{\sum_{j=1}^2 c(j)} \right) \quad (11)$$

b) Sequence Determination:

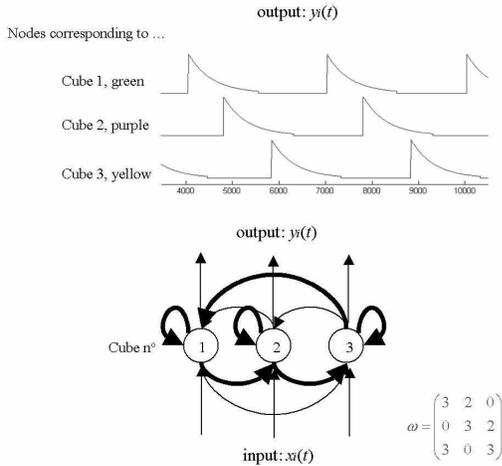


Fig. 5. **Top:** Firing of the 3 neurons in response to the motion of the 3 boxes. **Bottom:** A 3-node fully recurrent time delay neural network learns the sequence of node firing. The bold lines represent the connections with maximal weight. Reactivation of any of the 3 nodes result in correct rehearsal of the sequence.

In order to determine the probability of being in strategy s_{2_1} , in which the invariant is the sequence of displacement of the boxes, a time delay neural network is trained on the time series of *hits*. For instance, a sequence of move (box1, box2, box3) results in the series $H = \{h_1, h_2, h_3\}$. Each hit is associated to one node. The network is fully recurrent, see Figure 5. Each network connection is associated a time

delay τ and a weight w . If y_j and y_j^p are the observed and predicted activity of the neuron j , we have:

$$dy_j^p/dt = -\tau_{jj} + \sum_i w_{ij} \cdot y_j \cdot \delta(\tau_{ij}, y_j) \cdot \delta(y_j, 0) \quad (12)$$

The function $\delta(x, H)$ is a threshold function that outputs 1 when $x > H$ and 0 otherwise.

During learning, weights and time delay are updated following pseudo-Hebbian rules following Equations 13 and 14.

$$\delta w_{ji}(t) = a \cdot y_i(t) \cdot y_j(t) \quad (13)$$

$$\tau_{ji}(t) = \left(\frac{\tau_{ji}(t-1) \cdot \frac{w_{ji}}{a} + \frac{y_i(t)}{y_i(t)}}{\frac{w_{ji}}{a} + 1} \right) \cdot y_i(t) \cdot y_j(t) \quad (14)$$

where a is a constant factor by which the weights are incremented.

The error $E = \int |y(t) - y^p(t)| dt$ on the neural network prediction is a measure of the probability of observing the sequence. $P(s_{2_1}) = \delta(\epsilon, E) \in \{0; 1\}$, where ϵ is the minimal error for succesful learning.

c) Gesture Recognition:

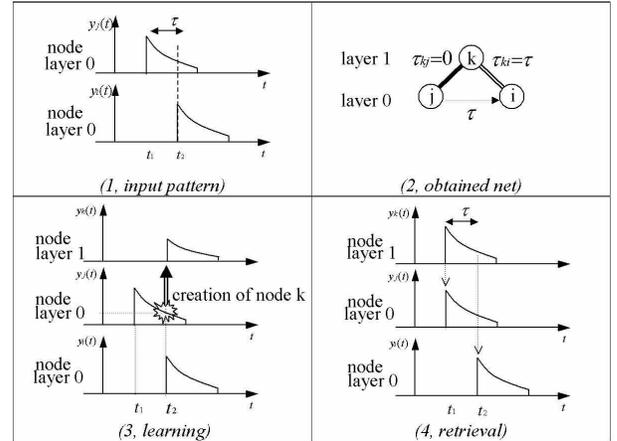


Fig. 6. Principle of functioning of the tree-based time delay neural network, used to learn sequential activation of joint motion. New nodes are created in response to new node correlations.

To determine whether we are in strategy s_{3_1} , i.e. a situation in which the gesture is an invariant, a tree-based time delay neural network is trained on the time series of joint segments, see Figure 7.

The angular trajectory of a joint is segmented and mapped to the activity of two nodes, that fire for positive and negative velocity, see equation 2 and Figure 8. Patterns of joint motion (time series of segmentation points) are stored in the network, following Equations 13 and 14, see Figure 6.

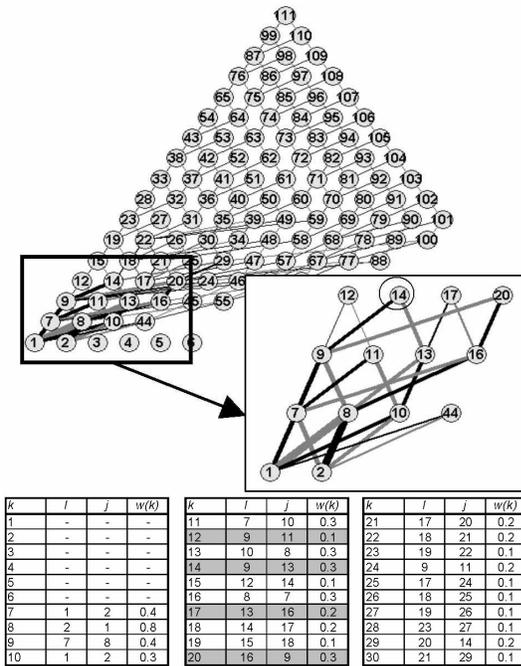


Fig. 7. Growth of the tree-network during learning of the motion, presented in Figure 1.

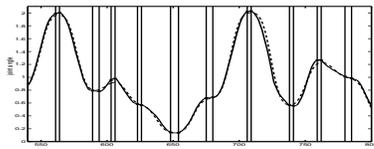


Fig. 8. Angular trajectory of the elbow joint during the motion shown in Figure 1. Vertical lines show the segmentation along the angular trajectory. The dashed line is the reproduced motion after network rehearsal and using a Gaussian-based fit.

IV. EXPERIMENTS AND RESULTS

The model was first tested in simulation, with the avatar playing in turn the role of demonstrator and imitator. The model could correctly disambiguate between all five first strategies of imitation, see the example of Figure 9. The same was confirmed using video data of human demonstration and implementing the reproduction on the ATR DB robot², see Figure 10 and video.

Imitation of the 3rd level of imitation (5th/6th strategies), that is recognition and reproduction of a gesture, was conducted separately on human data recorded with the SenSuit recording system. The system correctly disambiguated between random gestures and highly correlated ones, such as oscillatory motions, see Figure 8. It was, however, poorer on low correlated data (clapping hands

²Note that reproduction of strategies $s_{2,1/2}$ could not be tested, as only the left arm of the DB robot was functional during our visit at ATR

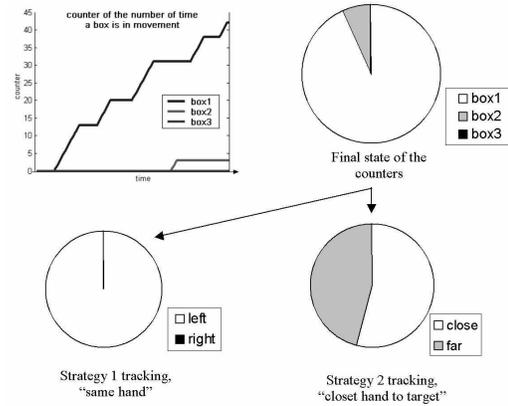


Fig. 9. **Top Left:** Incrementation of the counters recording the number of times each box is touched. In this example, box1 is moved frequently, while box2 is touched only once and very briefly, and box 3 is not touched at all. The pie chart represents the distribution of probability of each of the 3 possible strategies that could have accounted for that particular example.

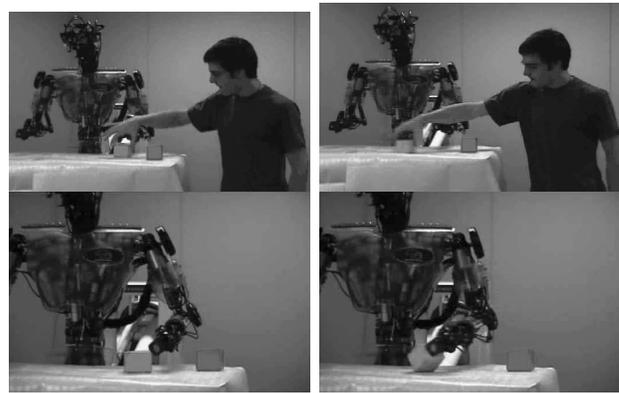


Fig. 10. **Top:** Observation Phase - The demonstrator moves each box from left to right. **Bottom:** ATR DB robot imitates the invariant of the demonstration, a motion of any block along the x-axis.

and reaching for target anywhere in space).

V. CONCLUSION

We addressed the problem of determining which features of a manipulation task are relevant and should be imitated. We proposed a metric of the imitation performance that determines the optimal imitation strategy, based on a measure of probability of observing a particular manipulation strategy. The metric uses a linear combination of probabilities to compare the results of separate algorithms for feature extraction, applied to different datasets. The model was successfully applied to the reproduction of six manipulation tasks in a dynamic simulator and on a humanoid robot, using kinematic data of human motion.

d) Discussion and Future Work:

Each of the methods used for discovering regularities in the data is not novel nor optimal. The novelty of this

work lies in the combination of these methods to extract a higher-level form of redundancy in the datasets, that no single method could extract alone, in order to determine a general imitation metric.

The work remained simple in the manipulation tasks it addressed. We considered only planar motions of the objects, manipulation sequence of no more than three time steps, and simple gestures for manipulating objects. This simplicity was necessary in order to validate the approach. Future work will consider more complex set of data. Presently, we are conducting a systematic evaluation of the method for extracting strategy in joint space, over a larger dataset of gestures, and will implement it, in the next months, on the ATR robot.

Although we attempted to give to the imitation metric a general definition, a number of assumptions on its form remained task-specific and should be revisited in future work. The linear combination of sub-metrics might not be valid in tasks where sub-metrics are correlated. Sub-metrics should be less dataset-specific and should specify classification algorithms that extract the nature (probabilistic, sequential) of correlation across any dataset.

ACKNOWLEDGMENTS

This work was supported by grants from the Swiss National Science Foundation, the EPFL student travel funds, and the Human Science Frontier Program.

VI. REFERENCES

- [1] A. Alissandrakis, C.L. Nehaniv, and K. Dautenhahn. Imitating with alice: Learning to imitate corresponding actions across dissimilar embodiments. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 32:4:482–496, 2002.
- [2] C.G. Atkeson and S. Schaal. Learning tasks from a single demonstration. In *Proceedings of the IEEE/RSJ Int. Conference on Robotics and Automation (ICRA'97)*, volume 2, 1997.
- [3] A. Billard. Imitation: A review. In *The Handbook of Brain Theory and Neural Network. 2nd Edition. Michael A. Arbib (editor)*, pages 566–569. MIT Press, 2002.
- [4] A. Billard and M. Matarić. Learning human arm movements by imitation: Evaluation of a biologically-inspired connectionist architecture. *Robotics & Autonomous Systems*, 941:1–16, 2001.
- [5] A. Billard and S. Schaal. A connectionist model for on-line learning by imitation. In *Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS2001)*, Hawaii, Oct. 2001.
- [6] J. Demiris and G. Hayes. Imitative learning mechanisms in robots and humans. In *published in the Proceedings of the 5th European Workshop on Learning Robots, Bari, Italy*, pages 9–16. also as Research Paper No 814 at the dept. of Artificial Intelligence at the University of Edinburgh, July 1996.
- [7] P. Gaussier, S. Moga, J.P. Banquet, and M. Quoy. From perception-action loop to imitation processes: a bottom-up approach of learning by imitation. *Applied Artificial Intelligence*, 7:1:701–729, 1998.
- [8] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Learning rhythmic movements by demonstration using nonlinear oscillators. In *Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS2002)*, pages 958–963, 2002.
- [9] M. Kaiser and R. Dillmann. Building elementary robot skills from human demonstration. In *Proceedings. International Conference on Robotics and Automation*, volume 3, 1996.
- [10] I. Kamon, T. Flash, and S. Edelman. Learning visually guided grasping: a test case in sensorimotor learning. In *IEEE Transactions on Systems, Man and Cybernetics, Part A*, volume 28:3, 1998.
- [11] C. Nehaniv and K. Dautenhahn. Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications. In In: J. Demiris and eds. A. Birk, editors, *Learning Robots: An Interdisciplinary Approach*. World Scientific Press, 1999.
- [12] B. Scassellati. Imitation and mechanisms of joint attention: A developmental structure for building social skills on a humanoid robot. In Nehaniv C., editor, *Computation for Metaphors, Analogy, and Agents*, volume 1562. Lecture Notes in Artificial Intelligence, Springer-Verlag series, 1999.
- [13] S. Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.
- [14] S. Schaal. The sl simulation and real-time control software package. Technical Report Computer Science Tech Report, <http://www-slab.usc.edu/publications/schaal-TRSL>, University of Southern California, 2001.
- [15] M. Skubic and R.A. Volz. Acquiring robust, force-based assembly skills from human demonstration. In *IEEE Transactions on Robotics and Automation*, volume 16:6, pages 772 –781, 2000.
- [16] M. Inaba Y. Kuniyoshi and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation, vol.10, no.6*, pages 799–822, 1994.