



## Discovering optimal imitation strategies

Aude Billard<sup>a,b,\*</sup>, Yann Epars<sup>a</sup>, Sylvain Calinon<sup>a</sup>, Stefan Schaal<sup>b</sup>, Gordon Cheng<sup>c</sup>

<sup>a</sup> ASL3-I2S-STI, EPFL, 1015-Lausanne, Switzerland

<sup>b</sup> CLMC-Lab, Computer Science Department, University of Southern California, Los Angeles, CA 90089-2520, USA

<sup>c</sup> HRCN, ATR-CNS, 2-2 Hikaridai, Seika-cho, Soraku-gun, 619-02 Kyoto, Japan

### Abstract

This paper develops a general policy for learning the relevant features of an imitation task. We restrict our study to imitation of manipulative tasks or gestures. The imitation process is modeled as a hierarchical optimization system, which minimizes the discrepancy between two multi-dimensional datasets. 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 imitative tasks and controls task reproduction by a full body humanoid robot.

© 2004 Elsevier B.V. All rights reserved.

*Keywords:* Imitation learning; Optimization; Classification; Humanoids

### 1. 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 formal mathematical framework for imitation learning [12,15].

Imitation learning needs to address the following three key questions: ‘what to imitate’, ‘how to imitate’ and ‘when to imitate’ [12]. Previous work has essentially focused on the question of “how to imitate” [3]. 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) [9–11,17], state-actions (turn left, move forward) [6,7], the path followed by

the manipulated object [2,14] and the joint trajectories of the demonstrator’s motion [4,5,8], which could be reproduced by pre-defined motor programs. The present work aims at complementing these previous approaches, by defining a general policy for learning ‘what to imitate’, i.e. which of the features of the task are relevant to the reproduction.

Recent work [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 tasks involving object manipulation and gestures by a humanoid robot. We take the perspective that the relevant features of the movement, i.e. those to imitate, are the features that appear most frequently, i.e. the invariants in time. We develop

\* Corresponding author.

E-mail address: [aude.billard@epfl.ch](mailto:aude.billard@epfl.ch) (A. Billard).

a general formalism for determining the metric of imitation performance.

Section 2 presents a general mathematical formalism to express the metric of imitation performance and the optimal control policy, that optimizes this metric. Sections 4 and 5 describe the implementation of the formalism to drive learning by imitation of tasks involving object manipulation and gestures by a humanoid robot. Section 6 discusses the limitations of the formalism and concludes this paper by offering a general perspective on the future work.

## 2. Formulation of the imitation task

An imitation task can be decomposed into the serial implementation of two processes: an *observation process* and an *imitation process*. The observation process consists of extracting relevant (i.e. in our case, invariant over time) features from a *demonstrated dataset*. The imitation process consists of generating an *imitated dataset* that minimizes the discrepancy between the demonstrated and imitated datasets.

### 2.1. Formalism

Let  $D$  be the dataset generated by the demonstrator while driven by a controller  $U$ .  $U$  is such that  $D(U) = \{\bar{X}, \bar{X}_0, \bar{\Theta}\}$ , where  $\bar{X} = \{x, \dot{x}, \ddot{x}\}$  and  $\bar{X}_0 = \{x_0, \dot{x}_0, \ddot{x}_0\}$  (3-dim Cartesian position, speed and acceleration), are the Cartesian trajectories of the hand and the object respectively, and  $\bar{\theta} = \{\theta, \dot{\theta}, \ddot{\theta}\}$  (angular position, speed and acceleration) the trajectory of the demonstrator's arm joints.

The imitation process consists of determining a controller  $U'$  (Fig. 1), that generates a dataset  $D'(U') = \{\bar{X}', \bar{X}'_0, \bar{\Theta}'\}$ , such that  $J$ , the *cost function* or the *metric* of the imitation task, is minimal:  $\delta J(D, D') = 0$ .

Each imitation task is defined by a set of constraints  $s \in [1, S]$ . For each constraint  $s$ ,  $\exists$  a controller  $U_s$  generating a dataset  $D_{U_s}$ , such that the associated metric  $J_{U_s}$  is minimal:  $\delta J_{U_s}(D_{U_s}, D'_{U_s}) = 0$ .

#### 2.1.1. Metric of imitation

We express the metric or cost function of the imitation task  $J$  as a linear combination of constraint-dependent cost functions  $J_{U_s}$ :

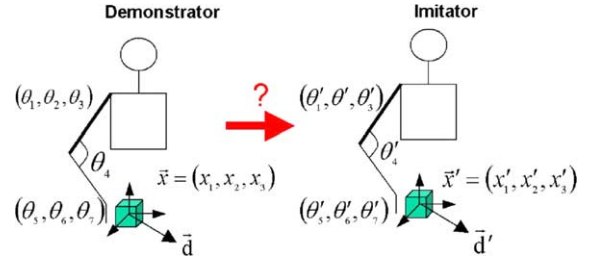


Fig. 1. Imitation of a manipulation task. The demonstrator's world is mapped into the imitator's world. Demonstrator and imitator's worlds are modeled as two datasets  $D = \{\bar{X}, \bar{X}_0, \bar{\Theta}\}$ ,  $D' = \{\bar{X}', \bar{X}'_0, \bar{\Theta}'\}$ .

$$J(D, D') = \sum_{u=1}^U \sum_{s=1}^S w_{u_s} \cdot J(D_{u_s}, D'_{u_s}). \quad (1)$$

The weights of the constraints-dependent cost functions are proportional to the likelihood  $P(U_s)$ , that the dataset  $D$  has been generated by the control strategy  $U_s$ :

$$w_{U_s} = \frac{P(U_s)}{\sum_{i=1}^U \sum_{j=1}^S P(i_j)}. \quad (2)$$

The weights are normalized, such that

$$\sum_{i=0}^U \sum_{j=1}^S w(U_s) = 1. \quad (3)$$

#### 2.1.2. Optimal imitation policy

We hypothesize that the constraints  $s$ , once identified during the observation process, remain the same during the imitation process. In the particular case where all control strategies are mutually exclusive, the optimal control strategy is unique. If  $P(U'_s)$  is the likelihood that the imitated dataset has been generated by the control strategy  $U'_s$ , we have  $P(U_s) = P(U'_s)$ .

In the case where different control strategies can coexist, because they would be acting on separate sub-datasets (e.g. one strategy tends to minimize a constraint on the objects trajectories, while another strategy tends to minimize a constraint on the arm joint trajectories), then the optimal control strategy is the combination of all the optimal control strategies for each sub-dataset, such that

$$P(U_s) = P(U'_s) \quad \forall s \in [1, S]. \quad (4)$$

This probabilistic framework is highly suitable to the analysis and production of Cartesian and joint trajectories, when using probabilistic methods, such as probabilistic distributions, Bayesian networks and Hidden Markov Models, as we will see in Section 5.

### 3. 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 indicating the end of the demonstration, the robot reproduces only the part of the demonstration that was found significant. In the work presented here, the demonstrator either performs a manipulation task using two color boxes (green and pink) on a table stand, see Fig. 2, or draws in the air stylized letters (see Fig. 5). When manipulating objects, the demonstrator performs four 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);
3. moves the boxes in a specific sequence (box 1, box 2, box 3);
4. moves all boxes using the same hand–box relationship (use the left hand only, or use the hand closest to the target).



Fig. 2. Left: observation phase—the demonstrator moves each box from left to right. Right: ATR DB robot imitates the invariant of the demonstration, a motion of any block along the  $x$ -axis.

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.

#### 3.1. Robot control

The work was conducted first in the Xanim simulator [16] and then implemented on DB, a 30 degrees of freedom (head 3, arms  $7 \times 2$ , trunk 3, legs  $3 \times 2$ , eyes four degrees of freedom) hydraulic humanoid robot, located at the Advanced Telecommunication Research Institute. The Xanim simulator is a dynamic simulation of the DB robot (see Fig. 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

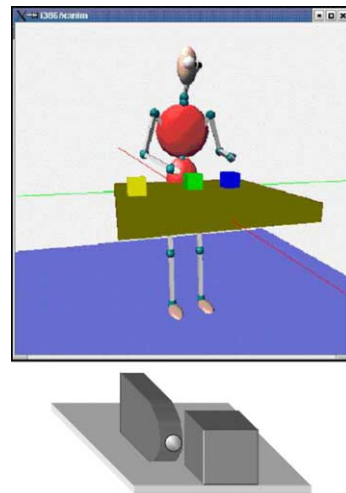


Fig. 3. Top: the Xanim simulator calculates the dynamics of a 30 degrees of freedom avatar and of 3 colored boxes [16]. Bottom: the end-effector of the robot is a solid plate, whose orientation in 3D can be modulated. In simulation and during control, the position of the end-effector is considered to be at the end of the plate (sphere in the drawing).

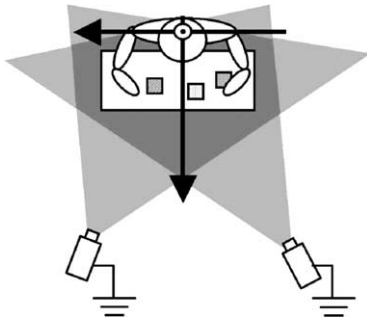


Fig. 4. A stereo color vision system used for tracking motions of colour boxes during demonstration and imitation.

action ‘pushing a box to the left’ follows a trajectory made of three points, located 10 cm to the right of the box, at the center of the box and 10 cm to the left of the box.

In tasks involving the observation of gestures, the joint trajectories of the demonstrator are segmented at fixed time interval, as described in Section 4.2.3. During reproduction of gestures, the joint velocities and trajectories are fitted with a third-order spline.

### 3.2. Data recording

The Cartesian trajectories of the colored boxes are tracked by a stationary pair of cameras (see Fig. 4). The system tracks the position and velocity of the three colored blobs at the rate of 60 Hz. The trajectories of the blobs are segmented into four 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; 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 shoulder of the demonstrator). Arm-choice and close-far measurements were available only in simulation.

Joint displacements, i.e. angular measurements of the four degrees of freedom of the demonstrator’s right arm (flexion, abduction and humeral rotation of the shoulder and flexion of the elbow), are recorded by a set of five Xsens sensors, that measure the absolute orientation of the torso, upper and lower arms (see Fig. 6). Data are captured at the rate of 100 Hz.

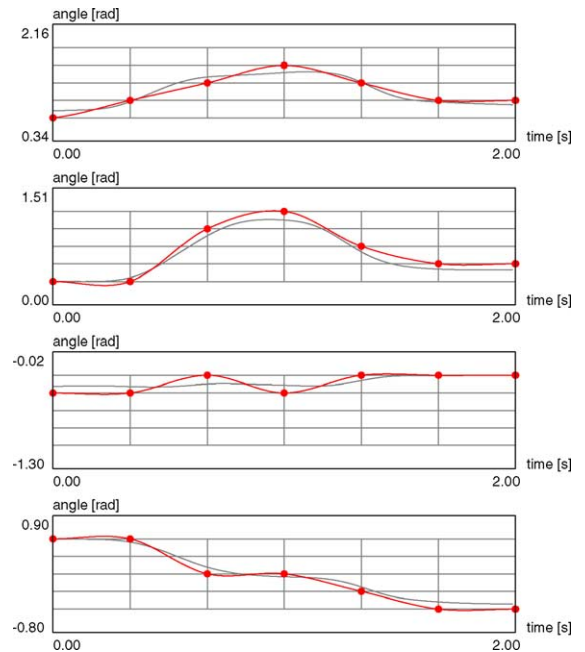


Fig. 5. The demonstrator, who wears an Xsens recording system, draws in the air four stylized figures: A, B, C and D. The robot reproduces a generalization over four examples of each drawing, varying the speed and amplitude of the demonstrated gestures if desired.

## 4. Experiments

The formalism, described in Section 2, was implemented to determine the metric and optimal control policies of different tasks consisting, either, of the manipulation of objects, or, of drawing stylized letters (see Fig. 2). These experiments were divided into four categories according to four levels of complexity or *levels of imitation* (see Section 1). A number of control strategies  $s$  were associated to each level  $l$  of imitation, as follows:

- $l_1$  – reproducing only the goal, reaching for the same box ( $s_{11}$ ) or for the same set of boxes in sequence ( $s_{12}$ ), irrespective of the path followed by the box (directed by the demonstrator) or of the gesture.
- $l_2$  – reproducing the path followed by the target, i.e. moving the box in a specific direction ( $s_{21}$ ).
- $l_3$  – reproducing the same hand–object relationship, i.e. using only left or right hand ( $s_{31}$ ), or using the hand closest to the target ( $s_{32}$ ).

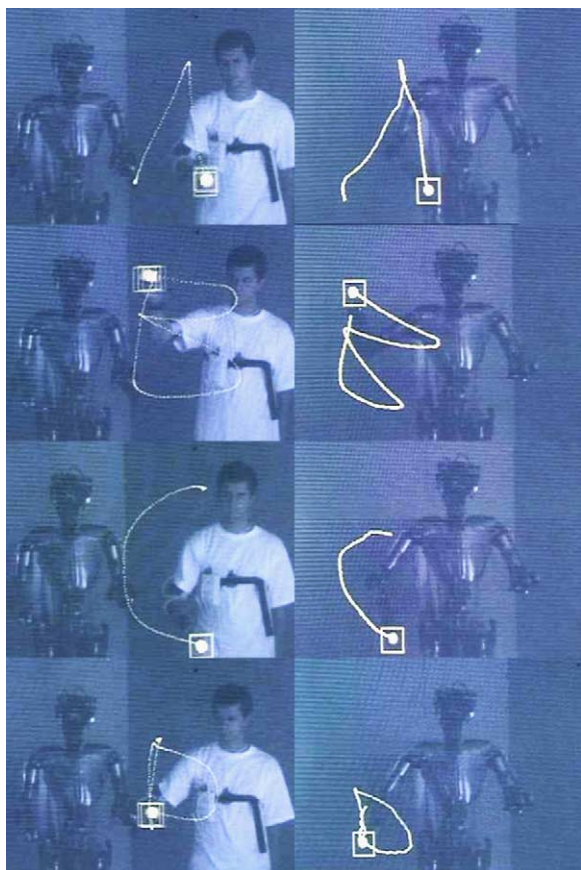


Fig. 6. From top to bottom: joint trajectories of the elbow and shoulder (flexion, abduction and humeral rotation) during drawing of the stylized figure A. The thin line corresponds to the demonstration. It is segmented into seven-time intervals of fixed length. Each segmentation point is then mapped to a quantization of five angles. Each trajectory is then encoded into one HMM of seven states and five observation symbols. The thick line corresponds to the trajectory produced by the HMM, using a third-order spline fit to interpolate between the segmentation points.

- $l_4$  – reproducing the exact gesture, i.e. reproducing the trajectory of each joint  $i$  ( $s_{4_i}$ ) with  $i \in [1, 4]$ .

#### 4.1. Strategy determination

Let  $h(i)$  with  $i \in [1, 3]$ ,  $d(i)$  with  $i \in [1, 8]$ ,  $a(i)$  with  $i = \{1, 2\}$ , and  $c(i)$  with  $i \in [1, 2]$  be the number of events hits, directed hits, arm-choice and close-far, respectively (see Section 3.2 for the definitions of the events). Then, in order to determine which strategy has

been used when observing the demonstration (valid for levels 1–3 only), we compute the following probabilities:

$$P(s_{1_1}) = \max_{i=\{1,2,3\}} \left( \frac{h(i)}{\sum_{j=1}^3 h(j)} \right) \quad (5)$$

$$P(s_{2_1}) = \max_{i=\{1,\dots,8\}} \left( \frac{d(i)}{\sum_{j=1}^8 d(j)} \right) \quad (6)$$

$$P(s_{3_1}) = \max_{i=\{1,2\}} \left( \frac{a(i)}{\sum_{j=1}^2 a(j)} \right) \quad (7)$$

$$P(s_{3_2}) = \max_{i=\{1,2\}} \left( \frac{c(i)}{\sum_{j=1}^2 c(j)} \right). \quad (8)$$

Once the strategy used by the demonstrator is determined, we define the metric and imitation controller. This is described in the following sections.

#### 4.2. Metrics and control policies

##### 4.2.1. Level 1 and level 2

For these two levels, the dataset consists of the trajectories of the objects  $D(U) = \{\vec{X}_O\}$ . The constraints are  $\vec{X}'_O = \vec{X}_O$  and the metric is expressed as

$$J = J_{1\&2}(\vec{X}_O, \vec{X}'_O) = \|\vec{X}_O - \vec{X}'_O\|.$$

The imitation control strategy consists, then, of (1) choosing the target object by drawing from the object distribution  $P(X'_O)$ ; (2) defining a set of three target points to reach the object  $\vec{X}_T = \{\vec{X}'_O - \vec{C}, \vec{X}'_O, \vec{X}'_O + \vec{C}\}$ , where  $\vec{C}$  is a fixed distance hand-object (see Section 3); (3) reaching each of the target points in sequence by inverse kinematics:<sup>1</sup>  $\Theta' = f^{-1}(X'_O)$ .

##### 4.2.2. Level 3: reproducing the hand-object relationship

The dataset consists of the trajectories of the hand and of the objects  $D(U) = \{\vec{X}, \vec{X}_O\}$ . The constraints are  $\vec{X}'_O = \vec{X}_O$  and  $\vec{X}' = \vec{X}$  (the constraints can also

<sup>1</sup> The inverse kinematics on the robot arm's seven degree of freedoms is solved by numerical iterations, based on the method *pseudo-inverse with optimization* (see [18]). The optimization criteria minimizes the distance to a set of rest positions.



include an equality between the velocities and the accelerations, which give an implicit measure of torque). The metric is expressed as

$$J = w_{1\&2} \cdot J_{1\&2}(\vec{X}_O, \vec{X}'_O) + w_3 \cdot J_3(\vec{X}, \vec{X}'),$$

where

$$J_3 = \|\vec{X} - \vec{X}'\|, w_{1\&2} = \frac{\sum_{i=1}^2 \sum_j P(s_{ij})}{\sum_{i=1}^3 \sum_j P(s_{ij})}$$

and

$$w_3 = \frac{\sum_j P(s_{3j})}{\sum_{i=1}^3 \sum_j P(s_{ij})}.$$

The imitation control strategy consists, then, of (1) calculating the distance hand–object  $\vec{Z} = \vec{X} - \vec{X}'_O$ ; (2) defining a set of three target points around  $\vec{Z}$ , as in levels 1–2 (cf. Section 4.2.1); (3) reaching each of the target points in sequence by inverse kinematics.

#### 4.2.3. Level 4: reproducing gestures

The gestures are encoded in a set of discrete Hidden Markov Models with  $N$  states and  $M$  observation symbols [13]. Each joint angle trajectory  $i \in [1, I]$  is segmented into  $N$  time intervals  $\Delta t_n$  of equal length and mapped into a set of  $M$  angles (the observations). Fig. 5 shows an example of trajectory segmented into  $N = 7$  fixed intervals and  $M = 5$  angles. Each segmentation point is referred to by  $\theta_{i_n}$  with  $n \in [1, N]$ .

The dataset consists, then, of  $I$  series of  $N$  angles  $D(U) = \{\theta_{i_n}\}$ . The constraints are:  $\forall n$  and  $\forall i$ ,  $\theta_{i_n} = \theta'_{i_n}$ . The metric is expressed as

$$J = \sum_{i=1}^I w_{4_i} \cdot J_{4_i}(\{\theta_{i_n}\}, \{\theta'_{i_n}\}),$$

where  $I$  is the number of joint angles,  $w_{4_i}$  the maximal log-likelihood over all the Markov sequences generated by all the HMMs associated with the joint trajectory  $i$  (see Section 5) and

$$J_{4_i} = \sum_{n=1}^N \|\theta_{i_n} - \theta'_{i_n}\|.$$

The imitation control strategy consists, then, of (1) generating the most likely sequences of  $\theta'_{i_n}$  by combining the most likely sequences generated by each

HMM; (2) determining the whole joint angle trajectory by applying a third order spline fit on the points given by the sequence; (3) going to each angular position by inverse dynamics.

## 5. 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 strategies of imitation, see the example of Fig. 7. The same was confirmed using video data of human demonstration and implementing the reproduction on the ATR DB robot<sup>2</sup> (see Fig. 2).

Imitation of the fourth level of imitation, i.e. recognition and reproduction of a gesture, was conducted separately on human data recorded with the Xsens recording system. Sixteen HMM models were trained to encode 4 gestures, consisting of drawings of the stylized letters A, B, C and D. Each gesture was presented four times to increase the generalization of each model. Each model's ability to recognize a trajectory was measured by calculating the likelihood that the observed trajectory could be generated by this model.<sup>3</sup> A model is said to be a good predictor, when its output likelihood is higher than a threshold. The threshold gives a measure of tolerance of noise and was determined experimentally. When training a model with randomly generated sequences, with the same length, number of states, and number of observation symbols as those used to encode the gestures, we observed that the log-likelihood decreased quasi-linearly when inserting errors<sup>4</sup> in the sequence (see Fig. 8). We chose a threshold of  $-0.3$  in log-likelihood, corresponding to approximately 30% of noise. The threshold is used, also, to determine whether a new model should be created to account for a newly presented gesture,

<sup>2</sup> Note that reproduction of strategies  $s_{2_1}$  and  $s_{2_2}$  could not be tested, as only the left arm of the DB robot was functional during our visit at ATR.

<sup>3</sup> Note that the likelihood returned by an HMM is highly dependent on the length of the observation sequence and the number of states. In order to compare successfully the likelihoods returned by the models, we applied a normalization in time and angular amplitude (see Fig. 5).

<sup>4</sup> The introduction of errors or noise consisted of changing randomly the value taken by one of the states of the sequence.

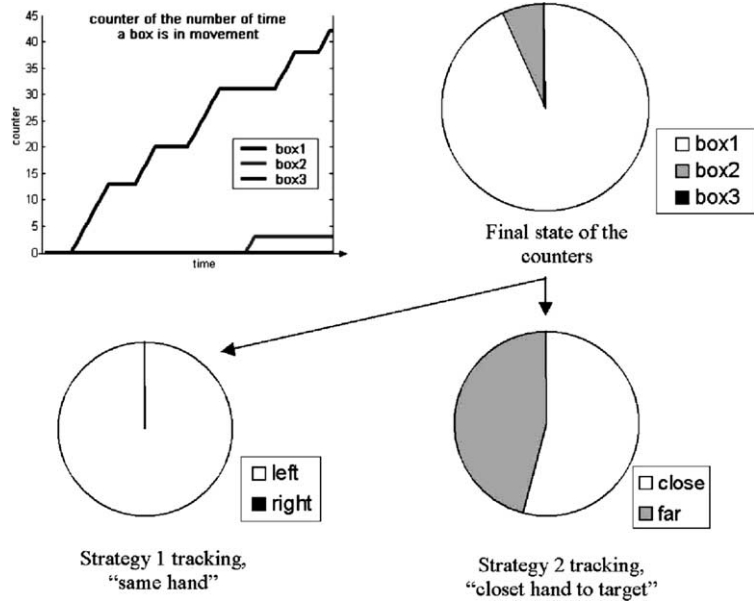


Fig. 7. Top left: incrementation of the counters recording the number of times each box is touched. In this example, box 1 is moved frequently, while box 2 is touched only once and very briefly, and box 3 is not touched at all. The pie chart represents the distribution of probability of each of the three possible strategies that could have accounted for that particular example.

and, thus, gives a measure of the metric of imitation performance. Reproducing each gesture consists of generating the most likely sequence, using the Viterbi algorithm. The trajectory reproduced by the model (and subsequently by the robot) is, then, a generalization over the different training samples (see Fig. 9).

The system learned and recognized correctly the four stylized letters A, B, C and D (see Table 1). The HMM encoding of the gestures is a parameter free representation, see Section 4.2.3, that is normalized in time and amplitude. This allows us to vary the amplitude and speed of the gesture at will during reproduction.

However, because storage depended on a fixed segmentation of the trajectories in time, strong dis-

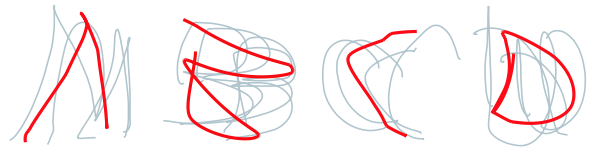


Fig. 9. Thin lines: Cartesian trajectories of the demonstrator's hand used to train the Hidden Markov Models (HMMs); bold line: trajectory reproduced by the robot, generalization over the different training samples and most likely trajectory generated by the different HMMs.

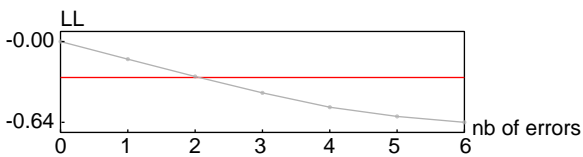


Fig. 8. Log-likelihood of an HMM when trained with randomly generated sequences of a fixed length of 7. The likelihood decreases quasi-linearly with the number of errors inserted in the sequence.

Table 1  
Imitation performance for the different levels of imitation

Levels of imitation	Robot learning (%)	Simulation retrieval (%)
1	10	0
2	–	0
3	5–30	–
4	5	0

The error (cost function averaged over the trials) is given in percentage.

tortions from the demonstrated trajectories could be observed in both the learned and reproduced trajectories. In order to overcome this problem, current work re-implements the model using continuous Hidden Markov Models and a feature-based segmentation method, based on zero-velocity crossing, as done in [4].

## 6. Conclusion

We addressed the problem of determining which features of a manipulation task are relevant and should be imitated. We proposed a metric of 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 different algorithms for feature extraction, applied to different datasets. The model was successfully applied to the reproduction of manipulation and drawing tasks in a dynamic simulator and on a humanoid robot, using kinematic data of human motion.

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. This simplicity was necessary in order to validate the approach. Future work will consider more complex sets of data. Presently, we are conducting a systematic evaluation of the method for extracting strategies in joint space, over a larger dataset of gestures.

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.

## Acknowledgements

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

## References

- [1] A. Alissandrakis, C.L. Nehaniv, 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) (2002) 482–496.
- [2] C.G. Atkeson, S. Schaal, Learning tasks from a single demonstration, in: *Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA'97)*, vol. 2, 1997.
- [3] A. Billard, Imitation: a review, in: M.A. Arbib (Ed.), *The Handbook of Brain Theory and Neural Network*, 2nd Ed., MIT Press, 2002, pp. 566–569.
- [4] A. Billard, M. Matarić, Learning human arm movements by imitation: evaluation of a biologically-inspired connectionist architecture, *Robotics and Autonomous Systems* 941 (2001) 1–16.
- [5] A. Billard, S. Schaal, A connectionist model for on-line learning by imitation, in: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2001)*, Hawaii, October 2001.
- [6] J. Demiris, G. Hayes, Imitative learning mechanisms in robots and humans, Published in the *Proceedings of the Fifth European Workshop on Learning Robots*, Bari, Italy (also as Research Paper No. 814 at the Department of Artificial Intelligence at the University of Edinburgh, July 1996, pp. 9–16).
- [7] P. Gaussier, S. Moga, J.P. Banquet, M. Quoy, From perception-action loop to imitation processes: a bottom-up approach of learning by imitation, *Applied Artificial Intelligence* 7 (1) (1998) 701–729.
- [8] A.J. Ijspeert, J. Nakanishi, S. Schaal, Learning rhythmic movements by demonstration using nonlinear oscillators, in: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2002)*, 2002, pp. 958–963.
- [9] M. Kaiser, R. Dillmann. Building elementary robot skills from human demonstration, in: *Proceedings of the International Conference on Robotics and Automation*, vol. 3, 1996.
- [10] I. Kamon, T. Flash, S. Edelman. Learning visually guided grasping: a test case in sensorimotor learning, in: *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 28: 3, 1998.
- [11] Y. Kuniyoshi, M. Inaba, H. Inoue, Learning by watching: extracting reusable task knowledge from visual observation of human performance, *IEEE Transactions on Robotics and Automation* 10 (6) (1994) 799–822.
- [12] C. Nehaniv, K. Dautenhahn, Of hummingbirds and helicopters: an algebraic framework for interdisciplinary



studies of imitation and its applications, in: J. Demiris, A. Birk (Eds.), *Learning Robots: An Interdisciplinary Approach*, World Scientific, 1999.

- [13] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition, in: *Proceedings of the IEEE*, 77: 2, February 1989, pp. 257–285.
- [14] B. Scassellati, Imitation and mechanisms of joint attention: a developmental structure for building social skills on a humanoid robot, in: C. Nehaniv (Ed.), *Computation for Metaphors, Analogy, and Agents*, vol. 1562, *Lecture Notes in Artificial Intelligence*, Springer-Verlag Series, 1999.
- [15] S. Schaal, Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences* 3 (6) (1999) 233–242.
- [16] S. Schaal, The sl simulation and real-time control software package, Technical Report Computer Science Tech Report, University of Southern California, 2001. <http://www-slab.usc.edu/publications/schaal-TRSL>.
- [17] M. Skubic, R.A. Volz, Acquiring robust, force-based assembly skills from human demonstration, *IEEE Transactions on Robotics and Automation* 16 (6) (2000) 772–781.
- [18] S. Tevatia, G. Schaal, Inverse kinematics for humanoid robots, in: *Proceedings of the International Conference on Robotics and Automation (ICRA2000)*, 2000.



**Aude Billard** is an SNF Professor at the School of Engineering at the Swiss Federal Institute of Technology (EPFL). She is also an Adjunct Assistant Professor at the Computer Science Department at the University of Southern California. She received her BSc (1994) and MSc (1995) in Physics from the EPFL, with specialization in particle physics at the European Center for Nuclear Research (CERN). She received her MSc in Knowledge Based System (1996) and her PhD in Artificial Intelligence (1998) from the Department of Artificial Intelligence at the University of Edinburgh. She, then, was a Post-doctoral Fellow at IDSIA and LAMI (EPFL, 1998–1999). Aude Billard was a Research Associate (1999–2000) and a Research Assistant Professor (2000–2002) at the Computer Science Department at the University of Southern California. Dr. Billard's research interests cover the fields of artificial neural networks, robotics, neural modeling, computational neuroscience and more generally machine learning. Her work tackles special topics, such as programming through demonstration, imitation learning and language acquisition.



**Yann Epars** obtained his BSc and MSc in Robotics Engineering from the Swiss Federal Institute of Technology (EPFL) in March 2003. He is now a Monbusho Scholar at the Department of Mechanical Engineering at the University of Tokyo. His MSc research at EPFL was on the development of statistical learning algorithms for robot programming by demonstration.



**Sylvain Calinon** is a PhD student with the Autonomous Systems Lab at the Swiss Federal Institute of Technology, Lausanne (EPFL). He received his BSc/MSc in Robotics Engineering from the Swiss Federal Institute of Technology, Lausanne (EPFL) in March 2003. His research interests cover robot programming by demonstration and human–robot interaction.



**Gordon Cheng** is currently a Senior Research Scientist and Head of the Department of Humanoid Robotics and Computational Neuroscience, Computational Neuroscience Laboratories of ATR International. He was a Center of Excellence (COE) and a Science and Technology Agency (STA) Research Fellow; both fellowships were taken while working in the Humanoid Interaction Laboratory, Intelligent Systems Division at the Electrotechnical Laboratory (ETL), Japan. Before coming to Japan he conducted his PhD study at the Department of Systems Engineering, The Australian National University, Australia.



**Stefan Schaal** is an Associate Professor at the Department of Computer Science and the Neuroscience Program at the University of Southern California, and an Invited Researcher at the ATR Computational Neuroscience Laboratory in Japan, where he held an appointment as Head of the Computational Learning Group during an international ERATO project, the Kawato Dynamic Brain Project (ERATO/JST). He is also an Adjunct Assistant Professor at the Department of Kinesiology of the Pennsylvania State University. Before joining USC, Dr. Schaal was a Post-doctoral Fellow at the Department of Brain and Cognitive Sciences and the Artificial Intelligence Laboratory at MIT, an Invited Researcher at the ATR Human Information Processing Research Laboratories in Japan, and an Adjunct Assistant Professor at the Georgia Institute of Technology. Dr. Schaal's research interests include topics of statistical and machine learning, neural networks, computational neuroscience, nonlinear dynamics, nonlinear control theory, and biomimetic robotics. He applies his research to problems of artificial and biological motor control and motor learning, focusing on both theoretical investigations and experiments with human subjects and anthropomorphic robot equipment.