

# Combining Dynamical Systems Control and Programming by Demonstration for Teaching Discrete Bimanual Coordination Tasks to a Humanoid Robot

Elena Gribovskaya and Aude Billard  
LASA Laboratory – Ecole Polytechnique Federal  
de Lausanne(EPFL)  
<http://lasa.epfl.ch>  
CH-1015 Lausanne, Switzerland  
[elena.gribovskaya@epfl.ch](mailto:elena.gribovskaya@epfl.ch), [aude.billard@epfl.ch](mailto:aude.billard@epfl.ch)

## ABSTRACT

We present a generic framework that combines Dynamical Systems movement control with Programming by Demonstration (PbD) to teach a robot bimanual coordination task. The model consists of two systems: a *learning system* that processes data collected during the demonstration of the task to extract coordination constraints and a *motor system* that reproduces the movements dynamically, while satisfying the coordination constraints learned by the first system. We validate the model through a series of experiments in which a robot is taught bimanual manipulatory tasks with the help of a human.

## Categories and Subject Descriptors

I.2.6 [Artificial Intelligence]: Learning; I.2.9 [Artificial Intelligence]: Robotics.

## General Terms

Algorithms, Performance.

## Keywords

Programming by Demonstration(PbD), Learning by Imitation, Human-Robot Interaction(HRI), Dynamical Systems, Bimanual Coordination, Humanoid Robot

## 1. INTRODUCTION

Recent advances in Robot Programming by Demonstration (PbD) endow robots with abilities to learn different tasks from human demonstrators in a natural and effective way. PbD operates at different levels of the task representation: from copying low-level features of the motion [3, 19] to inferring the user's intention and using a symbolic representation [5]. We follow a low-level approach to PbD, which

encapsulates the task constraints at the trajectory-level and apply it to the study of learning of bimanual coordination skills in humanoid robots.

Discrete bimanual coordination in Robot Programming by Demonstration (PbD) has received little attention so far. Bimanual coordination is not treated as a constraints per se, but emerges from requiring that the robot reproduces exactly the demonstrated joint trajectories [21], or some given object/hand relation [4]. However, none of these works address the problem of inter-limbs coordination explicitly by extracting high-level features of coordination.

From the viewpoint of the conventional robot trajectory planning, two-arms manipulation is a relatively researched topic, yet under certain limited assumptions. E.g., in [6, 20] the authors propose the approach to planning of both kinematic and dynamic aspects of linear movements; however, it has some restrictions: only horizontal movements are considered, and the model itself doesn't reflect coupling between the two arms. The latter drawback excludes the control over synchronization and mutual adaptation between the arms. The method is not meant to deal with external perturbations and learning. Some researchers adopt behavior-based approach to manipulation [7], it is advantageous in the sense that a robot has several hard-coded models of behavior, between which it can switch depending on a situation, providing a versatile feedback on changing conditions. Nevertheless, this approach needs algorithms implementing each particular behavior, i.e., grasping, coordination, etc.

In our work we try to overcome the limitations of the existing engineering and RbD approaches to bimanual manipulation, specifically, to find constraints responsible for coordination and the way to learn them; we are also looking for the algorithm for bimanual task reproduction allowing to adhere to a learned model of a task and generate natural and robust trajectories. We were inspired by the intensive research in bimanual coordination in human movement science. The past few years, theories of coordinative structure and coordination dynamics of human motion [13] have been gaining ground. These theories successfully explain and predict the emergence of coordination patterns in rhythmical movements, see e.g. Haken-Kelso-Bunz model [9]. In [18] the nonlinear dynamical system framework of rhythmical pattern formation was extended to explain a tendency towards synchronization in discrete bimanual movements with different amplitudes to account for experimental data [13].

In this paper, we follow such a dynamical system approach

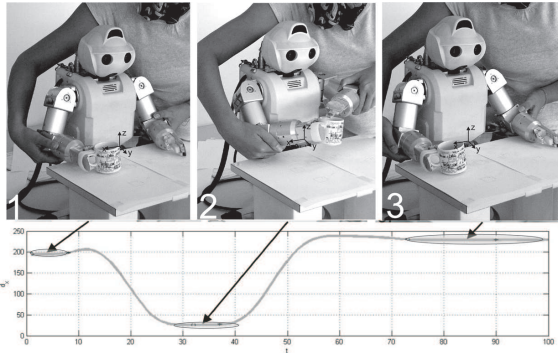
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HRI'08, March 12–15, 2008, Amsterdam, The Netherlands.

Copyright 2008 ACM 978-1-60558-017-3/08/03 ...\$5.00.

to encoding coordinated motion patterns. Specifically, we consider that a discrete coordinated movement can be described as a set of coordination patterns that change each other dynamically. We extend this approach to allow the system to be trained through a PbD paradigm.

According to dynamical system theory the movement patterns are described by collective variables, which contain information about the coordination patterns – spatio-temporal constraints typical of certain movements and governing co-operative behavior of the two arms. One part of the problem of explaining pattern formation in a coordinated task is to find an appropriate set of those variables. E.g., in [9] the relative phase between the two limbs is a suitable candidate for explaining the collective variable in rhythmical movements. Here, we hypothesize that, for discrete goal-directed movements, by analogy to the rhythmic case, the relative position between the two arms is an appropriate candidate for the collective variable. Stable positions (attractors) in this variable’s state space represent stable coordinated postures (coordination patterns), that must be reached in sequence to perform a task, see fig. 1 for an illustration.



**Figure 1:** Top: Example of a 3-step sequence of coordinated postures through which the robot transit when performing a *Tea task*. The three postures consist in: initial position, putting a piece of sugar in a cup, and back to the initial position. Bottom: Relative trajectory of the two hands along the  $x$  coordinate. The arrows superimposed to the trajectories show the location of the attractors (specified by ellipses), that correspond to the stable postures visible in the top pictures.

## 2. SYSTEM OVERVIEW

Our model is composed of two systems (see fig.2): 1) a *learning system*, responsible for extracting the task constraints and, thus, for skill acquisition; 2) a *motor system*, that dynamically generates the movements while satisfying the coordination constraints.

To generate unambiguous values for the two arms’ trajectories during the reproduction we have to specify in addition to the relative position between the two arms, the position of one of the arms. In bimanual tasks, one arm is usually more limited in its motion than the other. Therefore, we extract the key postures by analyzing the trajectories of the most constrained arm and of the relative position between the two hands, see fig. 4. We then infer the key postures of the less constrained arm based on this data.

In our experiment training of the robot was done through kinesthetic teaching, i.e. by a human demonstrator guiding passively the robot’s arms through the motions required to perform the task. Teaching the robot this way is advantageous in that it solves the correspondence problem. Kinesthetic coaching can be especially relevant for transferring coordination skills. Studying a coordinated movement involves the adoption of an optimal movement pattern. However, the optimality of such patterns is still a matter of contention in human movement science [11]; due to the fact that optimality to a great extent is different for each person, simply mimicking the postures will not necessarily be optimal. This problem is even more difficult in robotics because of the robot’s limited workspace. Robots cannot move their limbs into all the configuration available to humans. When the demonstrator moves the robot’s arms, she can feel the robot’s limitations, and properly adapt her motions to suit the robot’s particular abilities.

### 2.1 The Learning Process

During the learning process a robot builds a model of a skill by observing several demonstrations.

#### 2.1.1 Data acquisition

We use a Fujitsu HOAP-3 humanoid robot with 28 degrees of freedom (DOFs), 4 DOFs per arm. As we were interested solely in the arms’ movements (8 DOFs total), all other DOFs were set on constant positions. In the experiments reported here we produced 5 demonstration of each task. The recorded set of data for each task consists of  $N$  observations of joint angles values:  $\Theta = \{\theta_t^R, \theta_t^L; t = 1..N\}$ , where  $\theta_t^R, \theta_t^L \in R^4$  (here and further upper indexes “ $R$ ” and “ $L$ ” refer to either the right or the left arm).

#### 2.1.2 Data preprocessing

Data collected during kinesthetic teaching must be processed in order to harmonize a dimensionality between different trials, reduce the noise and the temporal dissimilarities. Thus the joint angles data  $\Theta$  recorded from the demonstrations are smoothed with 1D Gaussian filter of size 5; resampled to a unified number of observations  $M$ ; and temporally aligned by Dynamic Time Warping (DTW) [17].

During the experiments it was observed, that the temporal dissimilarities between the different trials have a negative impact on reliability of temporal constraints extracted by the *learning system*. A human demonstrator cannot preserve the same velocity along the analogous parts of complex movements, therefore misalignment between the different demonstrations are frequent. As a result, it becomes difficult at the subsequent stages of learning to match the patterns and get relevant time properties. Hidden Markov Models (HMMs) are known for their ability to deal with temporally heterogeneous data. However in our case, as we want not only to learn the correct sequence of postures, but also to learn their duration, we have to apply DTW prior HMMs to harmonize the signals in the time domain.

After pre-processing the recorded set of joint trajectories  $\Theta$ , the trajectories of the end-effectors  $\{x_t^R, x_t^L; t = 1..M\}$ ,  $x_t^R, x_t^L \in R^3$  and the relative position  $d_t = x_t^R - x_t^L$  across the two arms are calculated by vectorial projection.

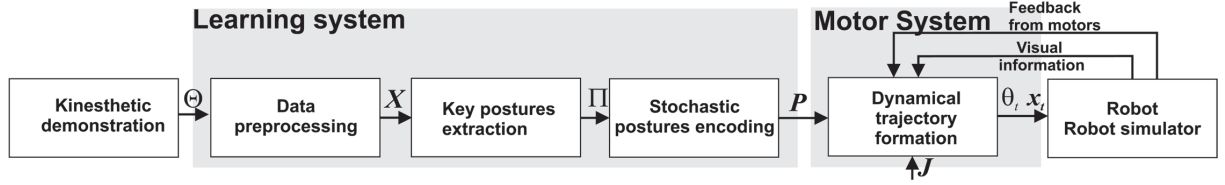


Figure 2: System overview: the arrows show the input-output flow across the system. The original signal  $\Theta$  is preprocessed to reduce the dimensionality and align temporally the signal. The resulting signal  $X$  is then segmented into a set  $\Pi$  of stable postures, which are further encoded in a HMM, to be reduced to a generalized set of spatio-temporal constraints  $P$ . The robot’s motion between each generalized posture are generated by a dynamical system, taking into account the robot’s geometry given by  $J$  the Jacobian. Control applies to both the position of arms  $x_t$  and to the joint angle configuration  $\theta_t$  at each time step.

### 2.1.3 Key postures extraction

In our work we consider a coordinated bimanual movement as a dynamical transition between stable coordinated postures. To automatically extract these postures from the data set  $X = \{x_t^R, d_t; t = 1..M\}$  we apply Mean Square Velocity analysis in Cartesian space to the trajectory of the relative position  $d$ .

In [14], a method was proposed to automatically segment a trajectory of a complex movement into episodes. We modified this approach to suit our task: each coordination pattern is described by the set of stable states of the relative position variable  $d$ . Stability is used here in a loose sense and corresponds to period of time during which  $d$  stay approximately constant (see fig.1) and the velocity decreases. In the ideal case it falls to zero. However, in practice, we can only observe it decreasing below a certain threshold. Thus we look in the Mean Square Velocity function  $\{V_t; t = 1..M\}$  specified by (1) for points  $t_i^b$  (an upper index “ $b$ ” indicates that a variable refers to the beginning of a segment) at which  $V_{t_i^b} < V^*$ , where  $V^*$  is specified by (2).

$$V_t = (\dot{d}_t^x)^2 + (\dot{d}_t^y)^2 + (\dot{d}_t^z)^2, t = 1..M; \quad (1)$$

where  $\dot{d}$  is the rate of change of the relative position  $d$ ;

$$V^* = \langle V \rangle - 0.5\sigma_V; \quad (2)$$

where  $\langle V \rangle$  and  $\sigma_V$  are accordingly the mean value and the standard deviation of Mean Square Velocity.

We then consider  $t_i^b$  as the beginning of the corresponding  $i$ th segment, and start to look for a point  $t_i^e$  (an upper index “ $e$ ” indicates that a variable refers to the end of a segment) at which  $V_{t_i^e} > V^*$ . If we find such point it means that the values of the relative position  $d$  on a interval  $[t_i^b, t_i^e]$  describes the key posture  $\hat{d}_i$ .

To specify the key postures of the right arm  $\hat{x}_i^R$ , we set up a correspondence between them and the key postures of the relative position  $\hat{d}_i$ . As a result the key postures of the right arm take the following values:  $\hat{x}_i^{R,b} = x_{t_i^b}^R$ ,  $\hat{x}_i^{R,e} = x_{t_i^e}^R$ ,  $i = 1..N_\Pi$ . The data set used for stochastic posture encoding has the following structure:  $\Pi = \{\hat{d}_i, \hat{x}_i^{R,b}, \hat{x}_i^{R,e}, t_i^b, \Delta t_i; i = 1..N_\Pi\}$ .

### 2.1.4 Stochastic postures encoding

We have several motivations to encode the postures stochastically with HMMs before their reproduction. These are: 1) to get rid of the spurious postures (the postures that occasionally appear as a result of an involuntary deceleration of a demonstrator’s motion, they are not relevant for the repro-

duction)(see fig.4 for illustration)<sup>1</sup>; 2) to join the postures close to one another (in the sense of time and distance); 3) to extract typical spatial and temporal characteristics of the postures.

We represent the key postures for the relative position  $d$  by a single continuous HMMs that encodes the set of the key postures in terms of both the time of their appearance and their duration  $\{\hat{d}_i, t_i^b, \Delta t_i; i = 1..N_\Pi\}$ .

The most general approach is to use a *fully-connected* model. Training this model needs a large set of training data, however the PbD paradigm assumes that a human demonstrator should not be asked to repeat a task an unbearable number of times. Hence, we incorporate prior knowledge about the data structure into the model through biased transitional probabilities. The states in our model represent certain events in the trajectory. Thus to guarantee the smoothness of the trajectories at the reproduction stage and to avoid knots, we constrain the possible transitions, by preventing backward transitions. We use a *left-right* model, with the particularity that self-transitions are not-allowed. Each emission probability is approximated with a single multivariate Gaussian distribution, and represents a key posture with its time properties.

The initialization of the parameters in HMMs plays a crucial role in the convergence of training: here we apply the Baum-Welsh algorithm [16], that locally maximizes the likelihood of data to obtain the estimation of HMMs parameters. Therefore if we initialize them randomly, the algorithm in most cases will converge to a wrong solution. The widely used method for initialization of HMMs is the K-means method [15]. However, it assumes that the number of clusters (corresponding to hidden states) is known, and consequently, before applying this method to initialize the HMMs, we first have to decide how many hidden states  $N_P$  our model should have.

The problem one encounters after segmentation is a different number of segments extracted for each trial. For K-means different cluster validation methods were proposed [12] in order to estimate whether a found partition properly explains the data and how reliable it is. We adopted the

<sup>1</sup>Given the accidental nature of such postures, the probability to transit into such a state is comparatively small, and we can thus eliminate them. Here, we fixed a threshold for the transitional probabilities of 0.2, under which transitions will not be allowed.

Dunn index according to speculations in [1] and the structure of our data, to validate the number of clusters our data contains. The final criterion reads as follows:

$$\Gamma(U) = \max_{1 \leq i \leq m} \left\{ \max_{1 \leq j \leq m} \frac{\delta(M_i, M_j)}{m \max_{1 \leq l \leq m} \Delta(M_l)} \right\}; \quad (3)$$

where  $U$  is a current partition, consisting of a set of clusters  $\{M_i, i = 1..m\}$ ,  $\delta(S, G) = \max_{s \in S, g \in G} (\|s - g\|)$  is a *between-cluster* distance,  $\Delta(S) = \max_{s, p \in S} (\|s - p\|)$  - *within-cluster* distance. The maximum value of the criterion (3) points out to the optimal partition.

The set of spatio-temporal constraints  $P$  has the following structure:  $P = \{\tilde{d}_i, \tilde{x}_i^{R,b}, \tilde{x}_i^{R,e}, \tilde{t}_i^b, \Delta \tilde{t}_i; i = 1..N_P\}$ , where  $\tilde{d}_i, \tilde{x}_i^{R,b}, \tilde{x}_i^{R,e}$  are, accordingly, updated values of the relative position and positions of the right arm in the beginning and in the end of a  $i$ th stable posture;  $\tilde{t}_i^b, \Delta \tilde{t}_i$  are the time of emergence and the duration of a  $i$ th posture.

## 2.2 Reproduction process

Next, we explain how the generalized set of postures learned by the HMM is used to control the reproduction of the task by the robot (see fig.2).

### 2.2.1 Hybrid controller

In previous work of ours [10], we proposed a hybrid controller for reaching movements in humanoid robots. This system is based on Grossberg's model of human reaching movement - Vector Integration to Endpoint (VITE) [2], and follows a current trend in human movement control: movements are not planned in a single frame of reference; rather, several frames of reference are involved in planning. This concept brings a certain degree of redundancy to the system, but we showed how such redundancy can be efficiently exploited to resolve the joint limit avoidance problem. In the work we report here, we took this controller as the basis for our motor system, see fig. 3.

The hybrid controller operates as follows. In the beginning of each part of a movement, the target Cartesian position of the left arm is reconstructed from the target position of the right arm and the stable posture that should be reached:  $\tilde{x}_i^{L,b} = \tilde{x}_i^{R,b} - \tilde{d}_i$ ;  $\tilde{x}_i^{L,e} = \tilde{x}_i^{R,e} - \tilde{d}_i$ , the target joint angles  $\tilde{\theta}^R, \tilde{\theta}^L$  are computed by inverse kinematics from the target Cartesian positions.

Then, the desired manipulator configurations  $\theta_{(t+1)}^{R,d}, \theta_{(t+1)}^{L,d}$  and  $x_{(t+1)}^{R,d}, x_{(t+1)}^{L,d}$  are obtained at each time step by the VITE controllers. This is done through the equations (4)-(6) (we present here only the equations for the right arm, as the equations for the left arm are identical):

$$\dot{\theta}_{(t+1)}^{R,d} = \dot{\theta}_t^R + \alpha_\theta^R (-\dot{\theta}_t^R + \beta_\theta^R (\tilde{\theta}_t^R - \theta_t^R)); \quad (4)$$

$$\dot{x}_{(t+1)}^{R,d} = \dot{x}_t^R + \alpha_x^R (-\dot{x}_t^R + \beta_x^R (\tilde{x}_t^R - x_t^R)); \quad (5)$$

$$\theta_{(t+1)}^{R,d} = \theta_t^R + \dot{\theta}_{(t+1)}^{R,d}; x_{(t+1)}^{R,d} = x_t^R + \dot{x}_{(t+1)}^{R,d}; \quad (6)$$

where  $\alpha_x^R, \alpha_\theta^R, \beta_x^R, \beta_\theta^R$  are empirically derived constants.

Generally, the desired arm configuration  $\theta_{t+1}^d$  is incompatible with the desired end-effector position  $x_{t+1}^d$ , i.e.  $x_{t+1}^d \neq K(\theta_{t+1}^d)$ . However, consistent positions exist in the certain neighborhood of the desirable position. To find such positions we minimize a function  $H(\theta_t^R, \theta_t^L, x_t^R, x_t^L)$  (7) imposing on it coherent constraints (8), that are mathematical expres-

sions of the robot's body constraints.

$$\begin{aligned} H(\theta_t^R, \theta_t^L, x_t^R, x_t^L) = & (\theta_t^R - \theta_t^{R,d})' W_\theta^R (\theta_t^R - \theta_t^{R,d}) + \\ & (x_t^R - x_t^{R,d})' W_x^R (x_t^R - x_t^{R,d}) + (\theta_t^L - \theta_t^{L,d})' W_\theta^L (\theta_t^L - \theta_t^{L,d}) + \\ & (x_t^L - x_t^{L,d})' W_x^L (x_t^L - x_t^{L,d}); \end{aligned} \quad (7)$$

where  $W_\theta^R, W_\theta^L, W_x^R, W_x^L$  are positive diagonal matrices, that control the influence of each controller <sup>2</sup>.

$$x_t^R = K(\theta_t^R); x_t^L = K(\theta_t^L); \quad (8)$$

where  $K(\cdot)$  is the kinematic function of the robot arm.

Solving (7) and (8) using the Lagrange multipliers, we get coherent values for the robot's joint angles that can then be used to control the robot:

$$\begin{aligned} \theta_{(t+1)}^R = & \theta_t^R + (W_\theta^R + J_t'^R W_x^R J_t^R)^{-1} (J_t'^R W_x^R (x_{(t+1)}^{R,d} - x_t^R) \\ & + W_\theta^R (\theta_{(t+1)}^{R,d} - \theta_t^R)); \end{aligned} \quad (9)$$

where  $J^R, J^L$  are the Jacobians of the right and the left arm:  $\dot{x}_R = J^R \dot{\theta}^R, \dot{x}_L = J^L \dot{\theta}^L$

To avoid the joint limits, we can gradually move from a cartesian controller to a joint angle controller when approaching the workspace boundaries, using time-dependent matrices  $W_\theta^R, W_\theta^L, W_x^R, W_x^L$ , see [10]:

$$\frac{w_{x,t}}{w_{\theta_i,t}} = 0.5\gamma(1 - \cos(2\pi \frac{\theta_{i,t} - \theta_{i,min}}{\theta_{i,max} - \theta_{i,min}})); \quad (10)$$

where  $w_{x,t}$  is the cartesian weight (i.e., a diagonal element of  $W_x$ ),  $w_{\theta_i,t}$  is the weight of angle  $i$  at a given time  $t$  (i.e., a diagonal element of  $W_\theta$ ),  $\theta_{i,min}$  and  $\theta_{i,max}$  are the corresponding joint angle boundaries,  $\theta^{i,t}$  is the corresponding angular position at time  $t$ , and  $\gamma$  is a constant setting the maximum value for  $w_x/w_{\theta_i}$ .

### 2.2.2 Spatial constraints

To preserve the coordination pattern learned from the demonstrations at given stages of the movement, we add to the above system a spatial constraint, according to:

$$\dot{x}_t^R - \dot{x}_t^L = 0 \quad (11)$$

Solving jointly the constraint optimization problem (7), (8) and (11) we get:

$$\theta_{(t+1)}^R = \theta_t^R + (M_1)^{-1} M_2; \quad (12)$$

$$\theta_{(t+1)}^L = \theta_t^L + [(J_t^L)^{-1} J_t^R] \theta_{(t+1)}^R; \quad (13)$$

$$\begin{aligned} M_1 = & W_x^R J_t^R + (J_t'^R)^{-1} W_\theta^R + W_x^L J_t^L \\ & + (J_t'^L)^{-1} W_\theta^L (J_t^L)^{-1} J_t^R; \end{aligned} \quad (14)$$

$$\begin{aligned} M_2 = & W_x^R (x_{(t+1)}^{R,d} - x_t^R) + (J_t'^R)^{-1} W_\theta^R (\theta_{(t+1)}^{R,d} - \theta_t^R) \\ & + W_x^L (x_{(t+1)}^{L,d} - \theta_t^L) + (J_t'^L)^{-1} W_\theta^L (\theta_{(t+1)}^{L,d} - \theta_t^L) \end{aligned} \quad (15)$$

### 2.2.3 Time constraints

Further, we add a time constraint to guarantee the synchronization of the arms (adaptation of one arm's velocity to another one) and the timing of the whole movement.

In general a discrete reaching movement is defined by its starting and ending positions. It can also be characterized by the mean movement time, as well as its precision [8]. Varying the parameters of the VITE model (4)-(6), while

<sup>2</sup>'''' refers to the transpose operator



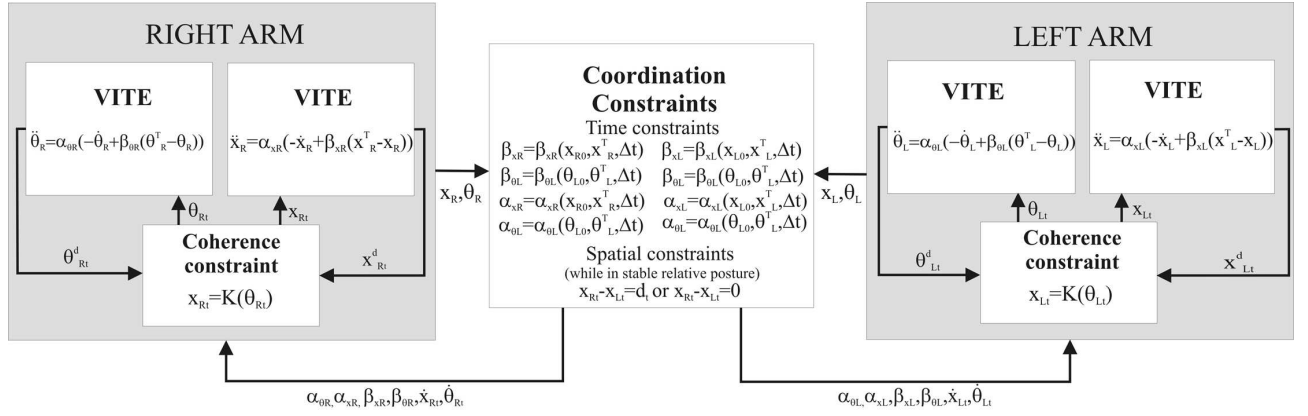


Figure 3: Overview of the robot's *Motor System*

the initial and target position are fixed, one can either increase or decrease the velocity or the movement duration. Here, we want on the contrary to find such values of the system's parameters that will guarantee that we have the desired time properties. We thus compute  $\alpha$  and  $\beta$  to make both arms reach the target positions simultaneously, even if their amplitudes are different (in agreement with experimental data on human motion, see [13]). We consider the continuous form of the VITE model and apply the following boundary conditions:

$$x(0) = x_0; v(0) = v_0; x(\Delta t) = x_1; v(\Delta t) = v_1; \quad (16)$$

where  $\Delta t$  is the duration of the movement (extracted by the *learning system*),  $x_0, v_0, x_1, v_1$  – respectively the Cartesian trajectories and velocities of the arm at the onset and end of the movement. The boundary conditions have to satisfy the following requirement:  $\|(x_1 - \tilde{x})/(x_0 - \tilde{x})\| \ll 1$ , where  $\tilde{x}$  – the target position. In addition, for each stable posture to satisfy the formal definition of an attractor of a dynamical system the velocity at the end of the movement must satisfy:  $\|v_1\| \ll 1$ .

The VITE model in the form we consider is a linear dynamical system. Therefore it has one attractor that can be either stable or unstable. We choose the eigenvalues  $\lambda_1$  and  $\lambda_2$  for our system that are complex conjugate with negative real parts:  $\lambda_1 = -m + ni, \lambda_2 = -m - ni$ , where  $m, n \in R, m > 0$  – are constants that must be specified,  $i$  – imaginary unity. Given the above, the attractor of the VITE model is asymptotically stable and the parameters  $\alpha$  and  $\beta$  take the following values:

$$\alpha = 2m; \beta = (4n^2 + \alpha^2)/(4\alpha). \quad (17)$$

Taking into account the above mentioned boundary conditions, we get for  $m, n$ :

$$n = \pi/\Delta t; m = \log((x_0 - \tilde{x})/(x_1 - \tilde{x}))/\Delta t. \quad (18)$$

If during the motion the robot encounters an external perturbation that affects the target position, the *motor system* has to react on it by adapting the velocity of both arms appropriately. In our system this goal is achieved by recomputing the parameters  $\alpha$  and  $\beta$ .

Table 1: A number of the postures found automatically by the *Learning system*

Name of task	Maximum number of segments	Number of postures
<i>Tea task</i>	6	4
<i>Cube task</i>	5	3
<i>Tray task</i>	7	4

### 3. EXPERIMENTAL RESULTS

We conducted three experiments to test our model for teaching the humanoid robot discrete coordinated movements. This procedure aimed to demonstrate how the robot's *motor system* could adapt to fulfill the coordination constraints of the tasks.

After learning, the robot has the implicit information about an object involved in a task, as its size and the initial position occur encoded in the collective variable and position of the leading arm. During the reproduction of a task the positions of the objects on the scene are tracked with the robot's stereovision system and partly with external stereovision system to increase the angle of view. The robot also has information concerning the time that should be spent for each part of the task, and about the fact that during certain parts of the movement, it should preserve the relative position between the two arms. All trajectories of the robot's arms are computed with respect to the frame of reference located at the center of the robot's waist.

Fig. 4 shows an example of posture encoding for the *Tea task*. The *learning system* determined four stable postures along the relative position  $d$  of the two arms. Only three of these postures (number 1,3,4) are relevant. Posture 2 is spurious, as it does not correspond to any specific movement pattern and appears only in a small number of trials. We can then exclude it from the reproduction because the probability to transit through it is lower than the specified threshold.

Table 1 collects information about the automatic selection of a number of key postures candidates in the three tasks. In general the segmentation procedure tends to extract more segments than there exist in reality. Statistical encoding with HMMs allows to optimize their number.

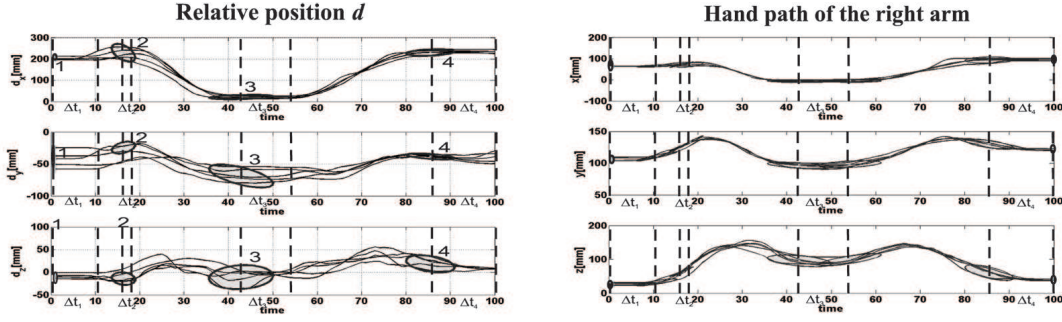


Figure 4: Encoding of the *Tea task*. On the pictures are plotted: the demonstrated trajectories of the relative position  $d$  with beginnings of the extracted postures 1-4 (they are represented with the corresponding variance information – bold ellipses); the demonstrated trajectories of the right arm. For the right arm, we extract both the beginnings and the ends of the postures.

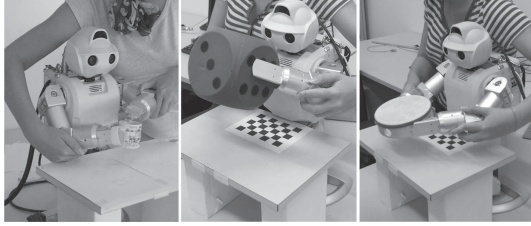


Figure 5: Set-ups of the 3 experiments. *Tea task*: put the piece of sugar into the cup. *Cube task*: grasp the cube, lift it, and put it on top of the pedestal. *Tray task*: grasp the tray with two arms, lift it, and move it forward.

Fig. 6 - 8 share the same legend: each figure contains the picture with a projection of reproduced trajectories into an axial plane, an accessible workspace of the robot in a current task and photos of an experiment with the superimposed trajectories. On the robot's workspace light areas point out to reachable positions.

Fig. 6 shows the results of the *Cube Task*. To test the *motor system* adaptability to external perturbations we first changed the position of the cube while the robot was trying to grasp it, and then we changed the position of the pedestal onto which the robot had to put the cube. During the second perturbation the robot was carrying the cube and thus had to preserve the relative position between its arms so as not to lose the cube. All positions of the objects on the scene were chosen so that they were reachable and thus within the robot's workspace.

Fig. 7 illustrates the results of reproducing the *Tea task*. In this experiment we also applied a perturbation twice: both times we simulated the situation where the right arm of the robot was suddenly pushed. This was achieved by sending the perturbed command to the arm's joints. The first time the jerk was initiated while the robot was bringing the two arms together, the second time, it was applied when the robot was putting the piece of sugar into the cup. The sudden changes were detected by measuring a discrepancy between the planned position of the robot's arm joints and the feedback from the motors. In each case, the robot readapts the position of both arms to make sure that the piece of sugar will not fall outside the cup.

Fig. 8 illustrates the results of reproducing the *Tray task*. We changed the position of the tray, while the robot was trying to grasp it.

Fig. 9 illustrates the preservation of the synchronization feature of the movement of the two arms of the robot during the *Cube task*. At time  $t_1$  while the robot was moving the arms towards the cube, we changed the position of the cube. Both arms adapted their trajectory simultaneously to handle this perturbation and reached the target simultaneously. In the same figure we show the velocity profiles of both arms in both Cartesian and joint-angle spaces. We see that the *motor system* produce smooth and bell-shaped velocity profiles similar to that of humans.

## 4. DISCUSSION

This work attempted to shed some light on what are the main aspects of bimanual coordination to guarantee satisfactory robot's performance in simple manipulatory tasks. We investigated two types of constraints: spatial constraints (e.g. the two arms must adopt a specific spatial relation to one another) and temporal constraints (the two arms must synchronize and should reach a target at the same time). Satisfactory performance was deemed achieved when the robot managed to go through the set of required postures, specifying bimanual constraints. The robot was, however, free to depart significantly from the arm trajectories shown during the demonstration in between each of these postures.

During the reproduction we applied several external perturbations to test the robustness of the system against those. While we could show that in some restricted parts of the workspace, the robot did handle well the perturbations, we must also point out two factors that may deteriorate the robot's performance: 1) the bimanual spatial constraints limit the robot's movements and, thus, reduce importantly the part of its workspace; 2) the magnitude of a perturbation should be sufficiently small and fast that the robot can still reach the target during the required duration of the movement; failing this, the system can potentially react with jerky movements. The above considerations determine the directions of our future work, namely: 1) to investigate bimanual coordination tasks with more pronounced asymmetry between the two arms; 2) to develop a mechanism in the *motor system* to automatically modify the time constraints, when those may become harmful for the robot.

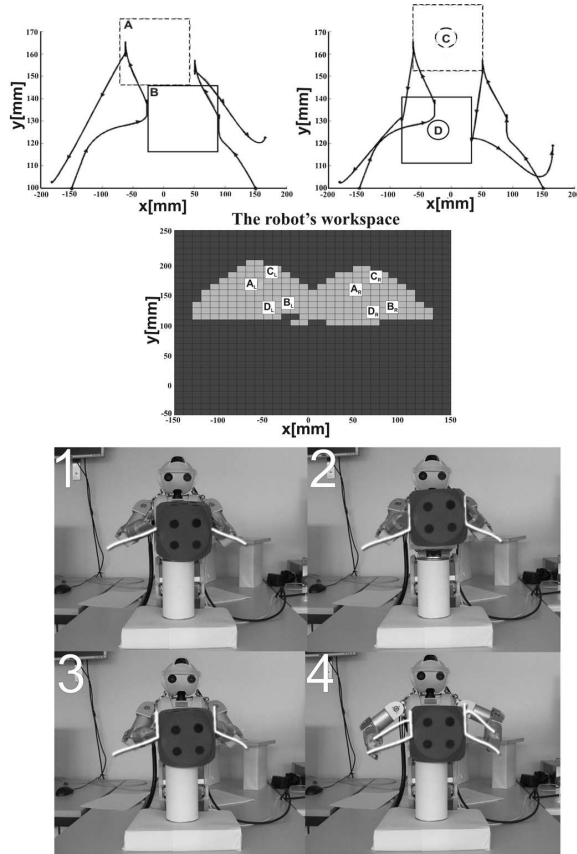


Figure 6: *Cube task*. Top-left: the robot tries to grasp the cube while its position is changed from A to B and it simultaneously adapts both arms to grasp the cube at position B. Top-right: while the robot is carrying the cube, the position of the pedestal is changed from C to D, the robot puts the cube on the new location preserving the relative position between its arms.

## 5. CONCLUSIONS

We presented a novel approach to learn discrete bimanual coordination skills in a humanoid robot. Its main aspects are: 1) a system that can automatically extract spatio-temporal coordination constraints across the two end-effectors; and 2) a *motor system* consisting of several coupled dynamical systems that is able to generate coordinated movements on-line, handle perturbations, while satisfying the learned coordination constraints.

The system was validated in three experiments where a humanoid robot was taught simple discrete bimanual coordinated tasks. We showed that the system could successfully reproduce the tasks under various external perturbations. Moreover, we identified a number of basic features of discrete bimanual coordination essential for effective manipulation.

## 6. ACKNOWLEDGMENTS

This work is supported by the European Commission through the EU Projects Robot@CWE and Felix-Growing.

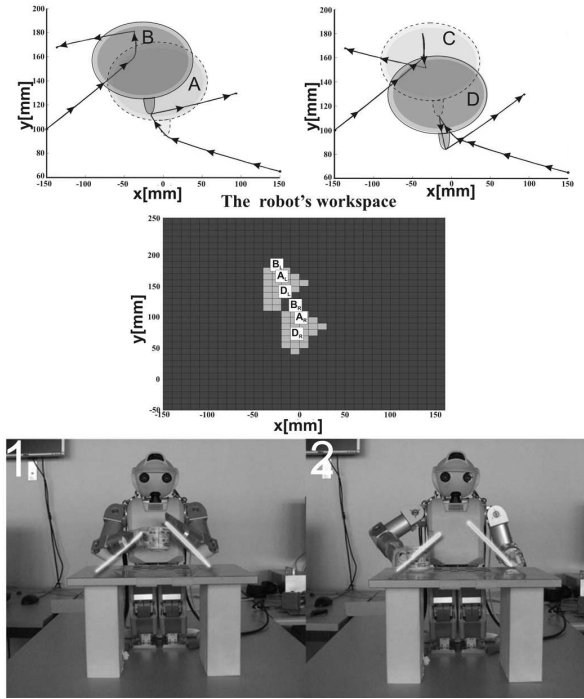


Figure 7: *Tea task* Top-left: the robot is bringing the two arms together, while the sudden change in position of the right arm is simulated, the robot simultaneously adapts both arms to put a sugar into the cup from the other position. Top-right: while the robot's arms are in the stable posture (the relative position between the arms is preserved) the right arm changes its position abruptly, however the *Motor system* preserves the posture and adapts the left arm.

## 7. REFERENCES

- [1] J. Bezdek and N. Pal. Some new indexes of cluster validity. *IEEE Transaction on System, Man, and Cybernetics*, 28:301–315, 1998.
- [2] D. Bullock and S. Grossberg. Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review*, 95(1):49–90, 1988.
- [3] S. Calinon and A. Billard. Stochastic gesture production and recognition model for a humanoid robot. In *Proceedings of IROS*, pages 2769–2774, September 2004.
- [4] S. Calinon and A. Billard. Incremental learning of gestures by imitation in a humanoid robot. In *Proceedings of the International Conference on HRI*, pages 255–262, March 2007.
- [5] Y. Demiris and B. Khadhour. Hierarchical attentive multiple models for execution and recognition of actions. *Robotics and Autonomous Systems*, 54:361–369, 2006.
- [6] J. Desai, M. Zefran, and V. Kumar. Two-arm manipulation tasks with friction assisted grasping. *Proceedings of the IROS*, 1:189–195, 1997.
- [7] A. Edsinger and C. Kemp. Manipulation in human environments. *Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots*, pages 102–109, 2006.
- [8] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.

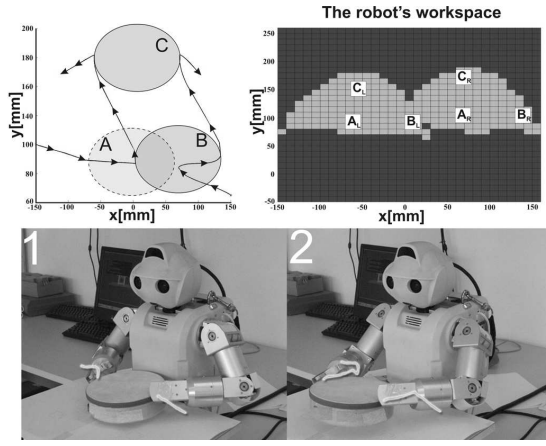


Figure 8: *Tray task*: Tray task: while the robot was trying to grasp the tray, it was moved aside to the boundaries of the workspace, even though, the perturbation applied was of big magnitude, the robot adapted successfully.

- [9] H. Haken, J. Kelso, and H. Bunz. A theoretical model of phase transitions in human hand movements. *Biological Cybernetics*, 51:347–356, 1985.
- [10] M. Hersch and A. Billard. A biologically-inspired model of reaching movements. In *Proceedings of the International Conference on Biomedical Robotics and Biomechanics*, pages 1067–1072, 2006.
- [11] N. Hodges and I. Franks. Modelling coaching practice: the role of instruction and demonstration. *Journal of Sport Sciences*, 20:793–811, 2002.
- [12] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.
- [13] J. Kelso, D. Southard, and D. Goodman. On the coordination of two-handed movements. *Journal of Experimental Psychological Human Perception*, 5(2):229–238, 1979.
- [14] J. Lieberman and C. Breazeal. Improvements on action parsing and action interpolation for learning through demonstration. In *Proceedings of International Conference on Humanoid Robots*, 1:342– 365, 2004.
- [15] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.
- [16] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of IEEE*, 77:257–286, 1989.
- [17] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, pages 43–49, 1978.
- [18] G. Schoner. A dynamic theory of coordination of discrete movement. *Biological Cybernetics*, 63:257–270, 1990.
- [19] A. Ude, C. Atkeson, and M. Riley. Programming full-body movements for humanoid robots by observation. *Robotics and Autonomous Systems*, 47:93–108, 2004.
- [20] M. Zefran, V. Kumar, E. Henis, and J. Desai. Two-arm manipulation: What can we learn by studying humans? In *Proceedings of the IROS*, 1995.
- [21] R. Zollner, T. Afour, and R. Dillman. Programming by demonstration: Dual-arm manipulation tasks for humanoid robots. In *Proceedings of the IROS*, 2004.

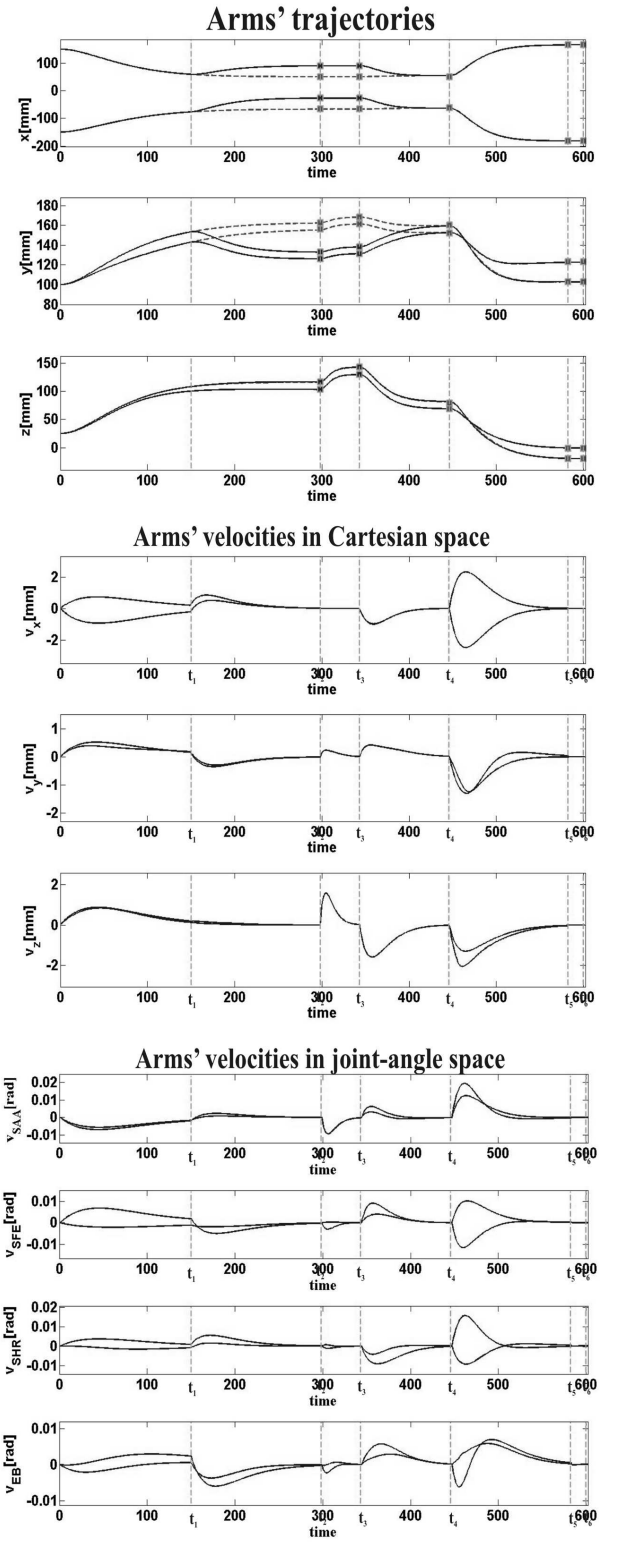


Figure 9: Synchronization in the *Cube task*: at  $t_1 = 150$  the position of the cube was moved. Simultaneously, the robot's *Motor system* adapted the trajectories of the two arms to reach the cube. The velocity profiles are smooth and bell-shaped. The time instances  $t_2..t_6$  refer to the boundaries of stable postures