Imitation Learning and Attentional Supervision of Dual-Arm Structured Tasks

R. Caccavale[1*], M. Saveriano[2*], G. A. Fontanelli[1], F. Ficuciello[1], D. Lee[2,3] and A. Finzi[1]
[1]DIETI, Università di Napoli Federico II, Napoli, Italy
{riccardo.caccavale,alberto.finzi,fanny.ficuciello,giuseppeandrea.fontanelli}@unina.it
[2]Human-centered Assistive Robotics, Technical University of Munich, Germany {matteo.saveriano,dhlee}@tum.de
[3]Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Germany

***Abstract*—In this work, we present an approach to imitation learning and flexible execution of dual-arm structured tasks. The proposed framework exploits imitation learning and attentional supervision to learn both a set of motion primitives and the associated tasks structure. During the teaching phase, attentional supervision allows the teacher to exploit attention manipulation, like object and verbal cueing, to facilitate the demonstration. In this phase, motion data are automatically segmented, annotated and learned in a compact form for on-line motion generation. During the execution phase, the learned task structure is exploited to synchronize left and right arm movements and to adapt task execution to the operative context. The proposed approach is demonstrated in a simulated kitchen scenario considering a pizza preparation task.**

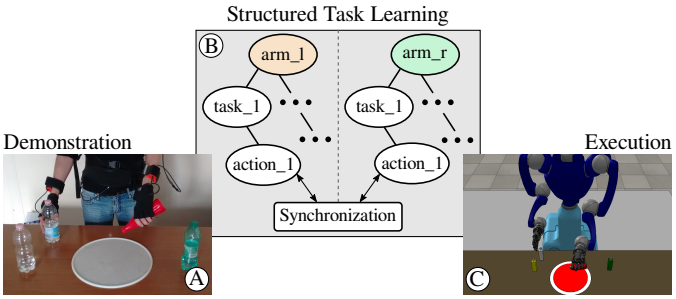


Fig. 1. Human demonstrations are supervised by an attentional system and then exploited to learn how to execute dual-arm hierarchical tasks.

I. Introduction

An effective robotic assistant needs to continuously learn new tasks and to adapt task execution to different situations. These tasks are usually structured in hierarchies of subtasks, where each subtask is associated with a set of primitive actions to be performed on some objects with a specific order. Learning how to coordinate these complex activities is particularly challenging for humanoid robots. Indeed, analogously to humans, humanoids have multiple redundant degrees-of-freedom, which can be exploited to simultaneously perform multiple subtasks and speed-up the task execution. A simultaneous execution of multiple movements, for instance a different subtask for each arm, requires a certain level of synchronization to avoid unsafe situations and to coherently execute the task. In order to teach a robot how to execute dual-arm structured tasks, we propose a framework that combines attentional supervision [1]–[4] and learning from demonstrations [5], [6] (see Figure 1). More specifically, the proposed approach allows an automatic segmentation of human demonstrations into motion primitives, while an attentional system monitors the generated primitives and relates them to a hierarchical task structure with the associated execution constraints. The attentional system allows the teacher to exploit attention manipulation, like objects accessibility, pointing gestures or verbal cueing, to facilitate and smoothly influence the teaching process [7]. Attentional mechanisms have been employed for robot teaching [8], [9] and imitation learning [10], however, in the proposed system these are fully integrated within a supervisory attentional system paradigm [1], [11]. In this setting, the attentional system not only supports implicit human-robot communication during both task teaching and execution, but also permits to represent and track human demonstrations at different levels of abstraction, from symbolic tasks/subtasks to physical motor commands. Other related works focus on learning structured tasks by demonstrations [12]–[16]. In [12] the authors present a system that permits to segment, recognize and understand human bi-manual demonstrations. Similarly, in [13], a graph structure to represent bi-manual tasks is learned from human observations. The main limitation of the aforementioned approaches is that they rely on a set of predefined motion primitives. On the other hand, the works by [17], [18] propose to learn bi-manual motion primitives from human demonstrations, but they do not consider the problem of sequencing and executing the multiple actions involved in a structured task. Alternative methods focus on learning a set of motion primitives from multiple demonstrations while automatically organizing them into graphs or automaton [19]–[21]. In contrast, we propose a different approach where attentional supervision is deployed to learn how to execute dual-arm structured tasks from a single human demonstration. In this case, imitation learning permits to learn low-level motion primitives relating them to the associated tasks and subtasks. The proposed segmentation and annotation of human demonstrations into basic primitives does not require any manual data processing. During task

*Joint first authorship.

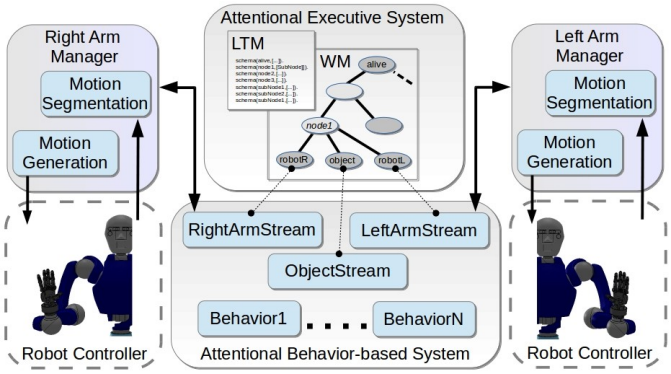


Fig. 2. The overall architecture for dual-arm task teaching and execution. The attentional system supervises task execution and learning, while the Arm Managers are responsible for activity segmentation, imitation learning, and motion execution. The attentional system manages the execution of high-level tasks (*Attentional Executive System*) and low-level sensorimotor processes (*Attentional Behavior-based System*).

execution, the attentional system continuously monitors the user and the robot activities to flexibly adapt action execution and sequencing; in this setting, the human can continuously interact with the robot switching to teaching sessions when novel tasks have to be demonstrated. The rest of the paper is organized as follows. Section II describes the proposed architecture for teaching/execution of dual-arm tasks. Section III details how human demonstrations are exploited to learn structured tasks. Experiments in a kitchen scenario are presented in Section IV. Conclusions and further extensions are presented in Section V.

II. Overview of the Proposed Framework

The proposed architecture for dual-arm task learning is illustrated in Figure 2. The different blocks of the proposed system are further described below.

A. Left and Right Arm Managers

At the control level, each arm of the robot is considered as a separate entity that is independently monitored and controlled. Indeed, arms synchronization is supervised and managed by the attentional system at a higher control level of the framework. The left/right Arm Manager (AM) handles several low-level aspects of the human-robot interaction. In particular, the AMs guarantee smooth transitions between teaching and execution phases; they also monitor the state of the arms with respect to the environment (arm-objects distance, action learned or executed); finally the AMs provide the motion data segmentation. Indeed, during a teaching session (see Section III), each AM generates a set of basic (point-to-point) motion primitives learned from human demonstrations. Learned primitives are encoded into stable dynamical systems (DSs), which allow compact representation and on-line motion generation. Moreover, stable DSs always converge to a given goal, and they can quickly react to eventual obstacles in the scene [22]–[24].

B. Attentional System

The attentional system exploits a hierarchical task representation to supervise and regulate the robot actions during the teaching and execution phases. Following a Supervisory Attentional System (SAS) approach [1], [2], actions control depends on two main mechanisms: the *contention scheduling* that regulates routinized activities allowing fast responses to environmental changes, and the *supervisory attentional system* that manages novel or complex activities and drives the system towards task accomplishment. In this work, we rely on the attentional system proposed by [3], [4]. The framework includes a Long Term Memory (LTM), a Working Memory (WM) and a set of sensorimotor processes (behaviors) (see Figure 2). The LTM contains the behavioral repertoire available to the system, including the denotations of hierarchically decomposed tasks and primitive actions (an instance can be found in (3)). The WM maintains the executive state of the system, which is represented by an annotated tree, whose nodes and edges represent, respectively, processes/behaviors and parental relations among sub-processes/sub-behaviors. These nodes can be either *concrete*, representing real sensorimotor processes, or *abstract*, which are for complex behaviors/tasks to be hierarchically decomposed. In this setting, the tasks and behaviors described by the LTM are to be allocated in the WM in order to be regulated and executed. The overall control cycle is managed by a special behavior (*alive*) that continuously updates the WM by allocating and deallocating hierarchical tasks/behaviors according to their denotations in the LTM. Once allocated in WM, the nodes are endowed with activation values that are regulated by top-down and bottom-up attentional mechanisms. The bottom-up regulation is given by a monitoring function $g(\sigma_b, \varepsilon_b) = \lambda_b$ that depends on behavior-specific stimuli σ_b and the behavioral state ε_b. In this work, analogously to [4], we consider the distance of targets as an estimation of behavioral accessibility. Behaviors are then associated with task-relevant elements of the scene and the stimulation σ_b is proportional to their proximity. This bottom-up regulation is top-down modulated by a magnitude value μ_b that summarizes the overall influence of the WM status on the attentional state of behaviors. Top-down and bottom-up influences are then combined in an *emphasis* value $e_b = \mu_b/\lambda_b$ representing the activation frequency for the behavior b. The behavioral activation level is then exploited to regulate behavioral competitions and conflicts. Indeed, multiple tasks can be allocated in the WM at the same time, therefore several behaviors can compete for the execution generating conflicts and impasses [25]. Contentions among alternative behaviors are solved exploiting the attentional activation: following a winner-takes-all approach, the behaviors associated with the higher emphasis are selected with the exclusive access to mutually exclusive resources.

III. Imitation Learning of Structured Tasks

During the teaching phase, the human activities are acquired through a motion capture system and on-line simulated in

a virtual environment. This imitation learning session is supervised by the attentional system, which has to connect the segmented training motions to the related tasks and subtasks. The attentional system tracks and monitors both the human and the robot task execution. This way, the low-level robotic actions taught by the user through imitation are labeled by the higher level tasks/subtasks managed by the attentional system.

A. Human Motion Retargetting

In this work, we exploit an Xsens motion capturing suit to collect task demonstrations from a human teacher. The Xsens provides the position and the orientation of 11 body parts, which are the sternum, the pelvis, the head, and the shoulder, elbow, wrist and hand of the two arms. The data from Xsens MVN can be read at 100 Hz. The robot to teach is RoDyMan, a 21 degrees-of-freedom (DoFs) humanoid robot. RoDyMan has 2 DoFs in the torso, two 7 DoFs arms, and 2 DoFs in the articulated head. The end-effectors are equipped with two anthropomorphic hands in order to provide enhanced dexterous manipulation skills. The robot has also a mobile base, which is not used in our setup. During the demonstration, the user can visualize the robot executing the task in a simulated environment (V-Rep). For this purpose, the human motions measured by the sensors have to be mapped to the robot kinematics in order to be properly replicated. The so-called motion retargetting problem is not trivial, since human and robot have different kinematics. In this work, the problem is addressed relying on Cartesian space variable and simple geometrical considerations. Figure 3 shows the reference frames for human and RoDyMan arms. The hand position $\boldsymbol{p}_h$ in the RoDyMan base frame is obtained using the following equation

$$\boldsymbol{p}_h = \boldsymbol{p}_s + \boldsymbol{R}_s \left(l_a \, \frac{\boldsymbol{p}_e - \boldsymbol{p}_s}{\|\boldsymbol{p}_e - \boldsymbol{p}_s\|} \; + \; l_{fa} \, \frac{\boldsymbol{p}_w - \boldsymbol{p}_e}{\|\boldsymbol{p}_w - \boldsymbol{p}_e\|} \right) \quad (1)$$

where $l_a = 0.350\,\text{m}$ and $l_{fa} = 0.305\,\text{m}$ are the RoDyMan arm and forearm length respectively. $\boldsymbol{p}_e$, $\boldsymbol{p}_w$ are the vector position of the elbow and wrist respectively provided by the Xsens data, while $\boldsymbol{p}_s = [0,\ 0.1081,\ 0.254]^T$ m is the shoulder position with respect to the base frame. Roughly speaking, the position of each end-effector of the robot is calculated from the position of the RoDyMan shoulder by summing two vectors of length l_a and l_{fa} respectively. These vectors are oriented as the relative human arm and forearm obtained from the Xsens. Notice that the positions provided by the Xsens depends to the torso movement, then it is necessary to rotate the vectors by the matrix $\boldsymbol{R}_s$ that represents the sternum orientation with respect to the Xsens world frame. For the end-effectors orientation, we simply use the hands orientation provided by the Xsens rotated in order to maintain the coherence with the RoDyMan hand frame. Therefore, the position of the two robots are scaled taking into account the difference in link dimensions between the robot and the human. The two computed homogeneous transformation matrix representing the target pose of the robot end-effectors are used in an inverse kinematics algorithm in order to control the robot. This approach allows us to

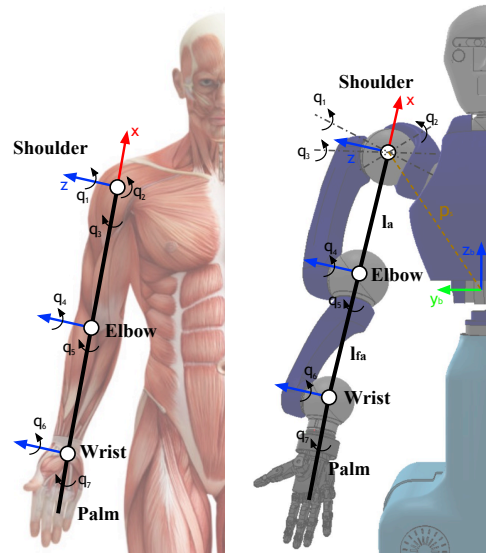


Fig. 3. Frames of the human and of the RoDyMan arms.

implement control policies using the redundancy of the robot in order to achieve secondary goals. In details, the null-space of the robot Jacobian matrix has been used to guarantee that: *i)* the torso positions are bounded, *ii)* the shoulders maintain a natural orientation, *iii)* the elbows are confined outside of a sphere centered at the center of torso to avoid self-collisions.

B. Motion Primitives Segmentation and Learning

Human demonstrations are to be segmented into the basic motion primitives that constitute the structured task. In our framework, segments are generated at run time while the user performs the task. Moreover, since our aim is to learn manipulation tasks which involve actions and objects, we need a segmentation strategy that is computationally inexpensive (on-line performance) and provides a matching between actions and manipulated objects. We exploit object-arm distance and human commands to rapidly and effectively segment continuous data. Similarly to [26], each object is included in a proximity region (a sphere of radius (we set $r = 0.1$ m) around the object). In this setting, a new segment is created when: *1)* one of the robot arms enters or leaves the proximity region, and *2)* the human commands something (usually to open or close the hand). As illustrated in Figure 4, during the demonstration the attentional system links the generated segments, in the form of symbols (strings), to the task structure associated to each arm. At the same time, the Arm Managers exploit the segmented data to learn a motion primitive for each symbolic action. We consider two classes of primitive actions, namely, Far-Object-Action (FOA) and Near-Object-Action (NOA). A FOA does not contain any point inside the proximity region of an object. FOAs consist of point-to-point motions ending at a specific target, and are modeled as a linear Dynamical Systems (DS). NOAs represent complex actions executed inside the proximity region, and are modeled using Dynamic Movement

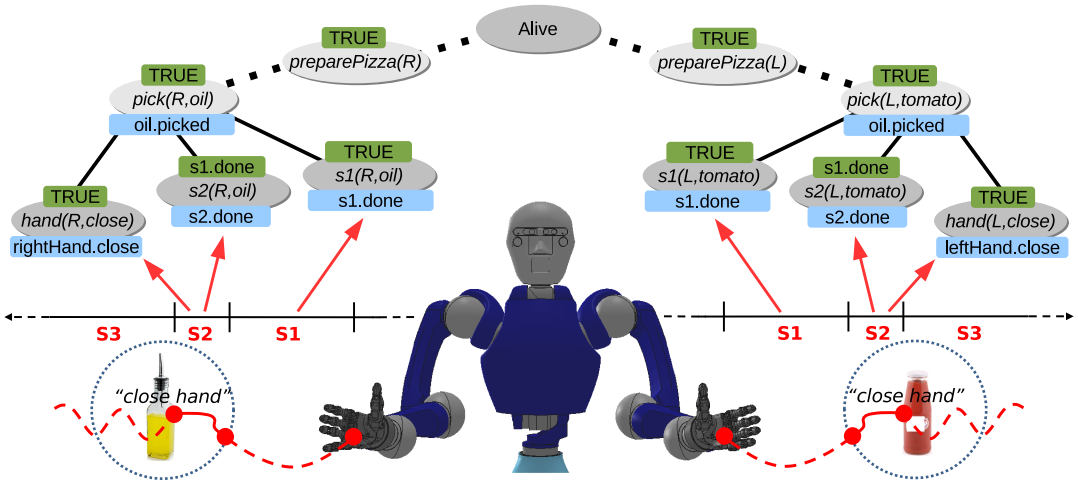


Fig. 4. Dual-arm action segmentation and hierarchical task decomposition. In the example, the robot has to pick-up tomato (*pick(L,tomato)*) and oil (*pick(R,oil)*) for the pizza-topping. The arm managers (down) perform action segmentations (S_1, S_2, S_3, ...) and learn motion primitives, while the attentional system (up) provides a duplicated representation of the *preparePizza* task for both arms connecting the generated segments to the task structure. The green and blue labels represent, respectively, releasers and post-conditions.

Primitives (DMPs) [27]. DMPs represent the motion as a non-linear DS:

$$\dot{\boldsymbol{p}} = \boldsymbol{v} \tag{2a}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{K}(\boldsymbol{g} - \boldsymbol{p}) - \boldsymbol{D}\boldsymbol{v} - \boldsymbol{K}(\boldsymbol{g} - \boldsymbol{p}_0)s + \boldsymbol{K}\boldsymbol{f}(s) \tag{2b}$$

$$\dot{s} = -\gamma s \tag{2c}$$

where $\boldsymbol{p}$ and $\boldsymbol{v}$ are, respectively, the robot position and velocity, and $\boldsymbol{K}$ and $\boldsymbol{D}$ are positive definite gain matrices. The non-linear force $\boldsymbol{f}(\cdot)$ allows to accurately follow the demonstration. Starting from the initial position $\boldsymbol{p}_0$, the DMP generates a smooth path towards the goal position $\boldsymbol{g}$. The clock signal $s \to 0$ deactivates the force term and guarantees convergence to the goal. A separate DMP is used to generate the orientation trajectory via Euler angles. Notice that the adopted segmentation strategy returns a Cartesian trajectory and the goal pose for each action. These data are sufficient to train the DMPs without further human intervention.

C. Task Learning and Execution

During the teaching phase, the segments and primitives generated from the human demonstration have to be connected to the high-level task structure and stored in the system repository (LTM). This process is managed by the attentional system exploiting environmental regulations and task-based constraints. In this setting, we assume that the abstract structure of the task to be learned is already represented in the LTM and allocated in the WM. We provide tasks and subtasks with a label that identifies which arm is involved in the represented activity. For instance, considering a simple pouring task, the activity can be hierarchically decomposed in the *take* and *pour* subtasks, which are denoted in the LTM by the following schemata:

$$\begin{aligned}
&\mathbf{schema}(add(A, O),\\
&\qquad \langle (subtask(A, take, O), A.hand.free),\\
&\qquad (subtask(A, pour, O), A.O.taken)\rangle,\\
&\qquad O.used).\\
&\mathbf{schema}(subtask(A, take, O), \langle\,\rangle, A.O.taken).\\
&\mathbf{schema}(subtask(A, pour, O), \langle\,\rangle, O.used).
\end{aligned} \tag{3}$$

In this case, the variable O represents the target of the pouring, while A is the arm involved in the action execution ($left$ or $right$). Schemata are also provided with preconditions and postconditions: the $take$ subtask is enabled when the hand of the arm A is free, while the $pour$ subtask is enabled when the left/right hand holds the object. Moreover, the latter subtask is finalized when the target-object O is used. Notice that, if both arms are enabled to execute a task, two instances of the related schema are allocated in WM with different values for A. During the human demonstration, the attentional system continuously monitors the environment and the task structure exploiting top-down and bottom-up regulations to enhance the activations of the subtasks which are accessible (i.e. closer to the associated target objects) and task relevant (i.e. stimulated by the task structure). When the left/right arm manager recognizes a new action all the left/right labeled subtasks compete to acquire the related segment (see Figure 4). In this process, the activation values are used to manage the competition following a winner-takes-all approach, where the most activated subtask acquires the segment. Furthermore, we provide preconditions for the new segments as follows. If the first segment attached to a subtask is a FOA, it is always enabled (i.e. $true$ precondition). If this is not the case, each segment is enabled after the execution of the previous one. These chaining constraints permit to keep the segments ordered according to the sequence acquired during the human demonstration. Instead, when a subtask starts with

a FOA segment it can be decoupled from the demonstrated sequence. Indeed, this segment represents the beginning of a new sequence, while the end-point of the previous segment is not needed as a precondition. Notice that such subtasks can be decoupled from the demonstrated sequence, in so enabling tasks reuse and flexible execution in different contexts. During the execution phase, all the enabled segments can compete for the execution, while the attentional system is to flexibly adapt the execution sequence to the operative context, exploiting the task structure and the action facilitation induced by the proximity of the salient elements in the scene. For instance, in a dual arm scenario, segments can compete for the associated arm (only one segment can be executed for each arm) and the target objects (only one arm can handle a single object); segments associated with different arms and objects can take place in parallel, otherwise a conflict occurs and the most activated segment is selected for the execution. This way, contention scheduling is here exploited as an implicit coordination mechanism during the execution of dual arm tasks.

IV. Case Study

In this section, we illustrate the system at work in a dual-arm robotic task to show that the proposed approach can be effectively applied for incremental learning and execution of structured tasks. Specifically, we consider a pizza topping task, inspired by the pizza domain proposed by the RODYMAN project. In this setting, the human operates in front of a table

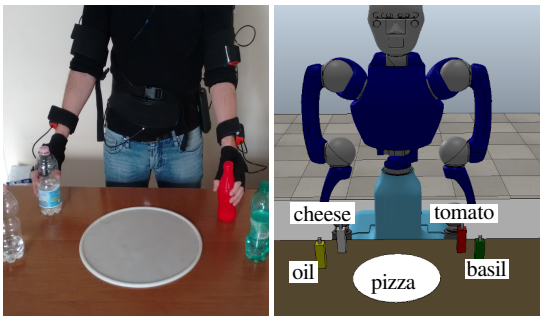


Fig. 5. Experimental setup.

where a set of real objects are disposed: the $pizza$, and four bottles containing $tomato$, oil, grated $cheese$ and $basil$ (see Figure 5, left). The environment is reproduced in simulation (Figure 5, right), where the human operator is replaced with the simulated humanoid robot. As detailed in Section III-A, a motion capture suite is deployed for the human motion recording and imitation. In this context, we consider two instantiations of the topping-task for both the arms of the robots. The experiment starts with two training sessions where the human demonstrates the entire task execution using only one arm per session. Both sessions last less then 4 minutes. At the end of the sessions the robot is able to reproduce the task with both arms. A subset of the learned segments is illustrated in Figure 6. In this case the task $add(tomato)$, learned by the left arm, is associated with 7 segments (3 NOA) and 2 grasping actions. In order to assess the system performance during the execution of the learned task, we executed 10 repetition of the pizza topping by randomly changing the position of the ingredients on the table. Notice that no collision avoidance mechanisms are employed during the task repetitions. Therefore, in order to avoid self collisions, the executive space is divided in a left, right and central area. The objects in the first two areas can be handled by only one arm (left and right arm respectively), while the central area, which contains the $pizza$, is shared. As a baseline for our tests,

TABLE I
Results for ten repetitions of the pizza task.

	Execution Time [s] (mean $\pm$ std)	**Success** Rate
Dual-arm	281.6 $\pm$ 13.83	1
Single-arm	332.4 $\pm$ 28.05	1

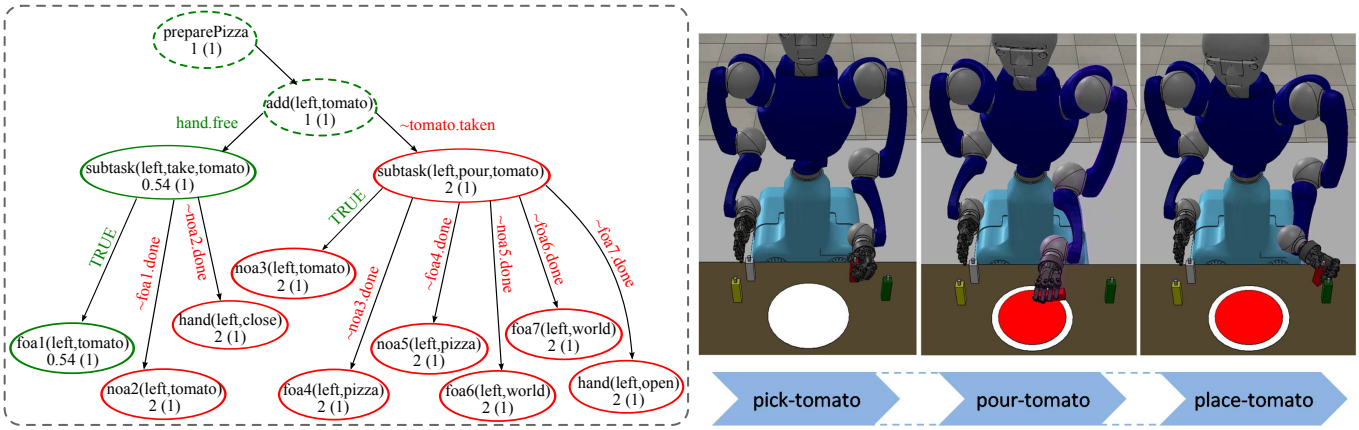
initially, we performed 5 task repetitions using only one arm per time. In this setting, the objects are randomnly positioned inside the left (3 times) or right (2 times) areas. Instead, in the last 5 repetitions, the objects are randomly positioned in all areas. In the second scenario, the robot can exploit the right-arm and left-arm learned tasks in order to prepare the pizza in a dual-arm setting. Snapshots of the dual-arm task execution are shown in Figure 7. In Table I, we illustrate the collected results. The executions have been always successful, in particular, the dual-arm repetitions improve the systems performances by 50.8 seconds (18.04%), suggesting that the system is able to efficiently combine the single-handed learned tasks, in so reducing the overall task execution time.

V. Conclusions and Future Work

We presented an integrated framework for learning and executing structured multi-arm tasks. In the proposed approach, the system can learn how to execute structured tasks from human demonstrations. In this context, human demonstrations are automatically segmented, while the generated segments are associated to motion primitives, which are supervised by an attentional system that associates them to a hierarchical task structure. At the execution time, the learned tasks can be exploited to sequence the task execution and to coordinate the motion of multiple arms. The framework has been tested in a kitchen scenario, where a humanoid robot is to prepare a pizza. The collected results show that the robot can quickly learn and robustly execute structured dual-arm activities that combine pick, place, and object manipulation actions. In our future research, we plan to incorporate a (self) collision avoidance mechanism in the arm managers and to perform experiments on the real robotic platform. Moreover, we are investigating low-level synchronization techniques in order to allow learning and execution of complex bi-manual tasks.

Acknowledgment

The research leading to these results has been supported by the FP7-IDEAS-320992 RODYMAN, by MARsHaL (DIETI), and partially funded by the Technical University of Munich,



(a) Representation in WM of the learned subtask *add(tomato)* for the left arm.

(b) Execution of the *add(tomato)* task

Fig. 6. Execution of the *add(tomato)* task performed by the left arm. The learned hierarchical structure (a) contains three NOA, one for tomato grasping and two for pizza reaching and tomato pouring. The simulated robot (b) reproduces the human motions following the learned hierarchical structure.

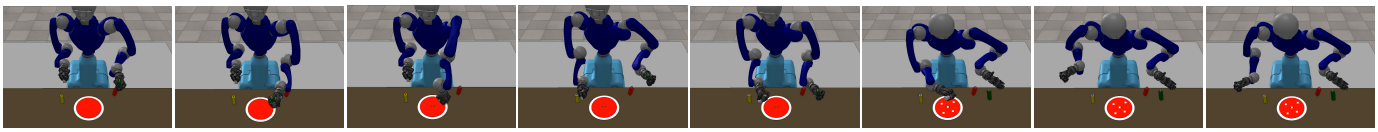


Fig. 7. Execution of the *add(cheese)* and the *add(basil)* tasks. The left and right arms cooperatively execute the task.

International Graduate School of Science and Engineering (IGSSE).

References

[1] D. A. Norman and T. Shallice, "Attention to action: Willed and automatic control of behavior," in *Consciousness and self-regulation: Advances in research and theory*, 1986, vol. 4, pp. 1–18.

[2] R. P. Cooper and T. Shallice, "Hierarchical schemas and goals in the control of sequential behavior," *Psychological Review*, vol. 113(4), pp. 887–916, 2006.

[3] R. Caccavale and A. Finzi, "Plan execution and attentional regulations for flexible human-robot interaction," in *SMC*, 2015, pp. 2453–2458.

[4] ——, "Flexible task execution and attentional regulations in human-robot interaction," *Trans. on Cognitive and Developmental Sys.*, 2016.

[5] D. Lee and Y. Nakamura, "Mimesis model from partial observations for a humanoid robot," *IJRR*, vol. 29, no. 1, pp. 60–80, 2010.

[6] M. Saveriano, S. An, and D. Lee, "Incremental kinesthetic teaching of end-effector and null-space motion primitives," in *ICRA*, 2015, pp. 3570–3575.

[7] M. N. Nicolescu and M. J. Mataric, "Natural methods for robot task learning: Instructive demonstrations, generalization and practice," in *Proc. of AAMAS*. ACM, 2003, pp. 241–248.

[8] C. Breazeal and M. Berlin, "Spatial scaffolding for sociable robot learning," in *Proc. of AAAI-2008*, 2008, pp. 1268–1273.

[9] A. Borji, M. N. Ahmadabadi, B. N. Araabi, and M. Hamidi, "Online learning of task-driven object-based visual attention control," *Image Vision Comput.*, vol. 28, no. 7, pp. 1130–1145, July 2010.

[10] E. Sousa, W. Erlhagen, F. Ferreira, and E. Bicho, "Off-line simulation inspires insight: A neurodynamics approach to efficient robot task learning," *Neural Networks*, vol. 72, pp. 123–139, 2015.

[11] R. Cooper and T. Shallice, "Contention scheduling and the control of routine activities," *Cognitive Neuropsych.*, vol. 17, pp. 297–338, 2000.

[12] K. Ramirez-Amaro, M. Beetz, and G. Cheng, "Understanding the intention of human activities through semantic perception: observation, understanding and execution on a humanoid robot," *Advanced Robotics*, vol. 29, no. 5, pp. 345–362, 2015.

[13] R. Jäkel, S. R. Schmidt-Rohr, S. W. Rühl, A. Kasper, Z. Xue, and R. Dillmann, "Learning of planning models for dexterous manipulation based on human demonstrations," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 437–448, 2012.

[14] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, no. 2, pp. 109–116, 2004.

[15] M. Pardowitz, S. Knoop, R. Dillmann, and R. D. Zollner, "Incremental learning of tasks from user demonstrations, past experiences, and vocal comments," *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 322–332, 2007.

[16] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot Programming by Demonstration," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, ch. 60, pp. 1371–1394.

[17] J. Umlauft, D. Sieber, and S. Hirche, "Dynamic movement primitives for cooperative manipulation and synchronized motions," in *International Conference on Robotics and Automation*, 2014, pp. 766–771.

[18] J. Silvério, L. Rozo, S. Calinon, and D. G. Caldwell, "Learning bimanual end-effector poses from demonstrations using task-parameterized dynamical systems," in *IROS*, 2015, pp. 464–470.

[19] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal, "Towards associative skill memories," in *HUMANOIDS*, 2012, pp. 309–315.

[20] S. Manschitz, J. Kober, M. Gienger, and J. Peters, "Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations," *Rob. and Aut. Sys.*, vol. 74, pp. 97–107, 2015.

[21] S. Niekum, S. Chitta, A. G. Barto, B. Marthi, and S. Osentoski, "Incremental semantically grounded learning from demonstration." in *Robotics: Science and Systems*, vol. 9, 2013.

[22] M. Saveriano and D. Lee, "Point cloud based dynamical system modulation for reactive avoidance of convex and concave obstacles," in *IROS*, 2013, pp. 5380–5387.

[23] ——, "Distance based dynamical system modulation for reactive avoidance of moving obstacles," in *Proc. of ICRA-2014*, 2014, pp. 5618–5623.

[24] M. Saveriano, F. Hirt, and L. Lee, "Human-aware motion reshaping using dynamical systems," *Pattern Recognition Letters*, 2017.

[25] M. M. Botvinick, T. S. Braver, D. M. Barch, C. S. Carter, and J. D. Cohen, "Conflict monitoring and cognitive control." *Psychological review*, vol. 108, no. 3, p. 624, 2001.

[26] M. Wächter, S. Schulz, T. Asfour, E. Aksoy, Wörgötter, and R. Dillmann, "Action sequence reproduction based on automatic segmentation and object-action complexes," in *HUMANOIDS*, 2013, pp. 189–195.

[27] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical Movement Primitives: learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.