

User Evaluation of an Interactive Learning Framework for Single-Arm and Dual-Arm Robots^{*}

Aleksandar Jevtić, Adrià Colomé, Guillem Alenyà, and Carme Torras

Institut de Robòtica i Informàtica Industrial, CSIC-UPC
C/ Llorens i Artigas 4-6, 08028 Barcelona, Spain
{ajevtic, acolome, galenya, torras}@iri.upc.edu

Abstract. Social robots are expected to adapt to their users and, like their human counterparts, learn from the interaction. In our previous work, we proposed an interactive learning framework that enables a user to intervene and modify a segment of the robot arm trajectory. The framework uses gesture teleoperation and reinforcement learning to learn new motions. In the current work, we compared the user experience with the proposed framework implemented on the single-arm and dual-arm Barrett’s 7-DOF WAM robots equipped with a Microsoft Kinect camera for user tracking and gesture recognition. User performance and workload were measured in a series of trials with two groups of 6 participants using two robot settings in different order for counterbalancing. The experimental results showed that, for the same task, users required less time and produced shorter robot trajectories with the single-arm robot than with the dual-arm robot. The results also showed that the users who performed the task with the single-arm robot first experienced considerably less workload in performing the task with the dual-arm robot while achieving a higher task success rate in a shorter time.

Keywords: Robot manipulators, user intervention, robot adaptation, gesture recognition, visual servoing, reinforcement learning

1 Introduction

Expectations from social robots are high, since they must successfully assist the users with daily tasks while maintaining their engagement in interaction over extended periods of time [1]. One way to achieve this is by enabling the robots to learn from the interaction with the users [2]. In this work, we studied the interaction of lay users when asked to modify previously learned tasks of single-arm and dual-arm robots (experimental setup is shown in Fig. 2). There is usually

^{*} This work was supported by the Beatriu de Pinós fellowship, reference num.: 2013 BP-B 00239, jointly funded by the Government of Catalunya, Spain and the European Commission FP7 COFUND programme. The work was partially supported by the EU CHIST-ERA I-DRESS project, reference num. PCIN-2015-147, and the national Spanish project RobInstruct, reference num. TIN2014-58178-R.

a strict division between the tasks that require one or two arms, however, we compare the same aspects of the two robot settings and the effects that a previous user experience with one setting may have on another. User performance was quantitatively evaluated in terms of user workload, task success rate and task execution time. The presented results are expected to provide future developers an insight into the challenges associated with the interaction design for lay users of such robots.

1.1 Relevant Work

Service robotics is a continuously growing field in various application domains [3]. Special attention is given to healthcare domain, where socially interactive assistant robots help patients with the recovery and compensate for increased costs of care and lack of nursing staff [4], [5]. Service robots are built for specific task and their functionality is defined with a set of skills necessary to perform them. Some studies show that specific types of users such as older adults prefer robot assistance over human assistance for tasks related to manipulating objects [6]. In this work, we tackle the problem on the level of trajectory adaptation and the proposed robot displacement tasks were used to evaluate the user experience.

Many challenges are associated with single-arm and dual-arm robots placed in human environments [7]. Some studies compared the two robot settings for manipulation tasks [8], but they did not involve robot learning from interaction. Learning from Demonstration (LfD) is a widely used framework for robot learning [9]. A single demonstration is often not sufficient and some refinements must be performed often through exploration in simulation or with a physical robot. Repeating a demonstration of the entire task is not always necessary, especially if the task is long or complex. In our previous work, we proposed an interactive learning framework that enables a user to select and modify a task segment [10] but this framework has never been evaluated in the experiments with users.

LfD methods often rely on robot vision to track and follow the user during a demonstration [11]. Gestures are commonly applied to initiate, stop or switch between robot tasks, but also give insight into the intended user behavior. Various methods for gesture recognition from RGB cameras have been proposed in literature [12]. In our previous work, we showed that the recognition of pointing gestures from depth images can be applied to real-time mobile robot guidance [13]. Here, the concept of visual robot guidance is implemented on robot manipulators and a Reinforcement Learning (RL) algorithm is applied to produce an improved robot motion segment from a series of user interventions. Our study compares the user experience with single-arm and dual-arm robots that use the previously proposed framework.

2 Algorithms

The proposed framework has two components: interaction and learning, however, only the interaction component was analyzed in this study. The user intervention

algorithm that allows users to modify the initial robot trajectory relies on the gesture recognition algorithm for switching between different robot states and hands motion following algorithm for robot teleoperation. The algorithms were implemented in C++ using the Robot Operating System (ROS) framework. A detailed description of the algorithms can be found in [10].

2.1 Robot Control

In order to allow the robot to correctly mimic the human motion, we used a similar framework as in [14] (see Fig. 1), where the tracking and gesture recognition algorithms provided desired poses, which were transformed to desired joint positions that satisfy the kinematics constraints by using an Inverse Kinematics (IK) algorithm [15]. Desired joints positions are provided at a different rate than the control loop's execution rate, therefore a dynamic system was used to interpolate the missing values. In order to store the trajectory, we record equally-spaced Cartesian goal points \mathbf{x}_t sent to the robot, and thus we could reproduce the user's hands motions by sending the same sequence of the desired points \mathbf{x}_t , $t = 1..N_{points}$. Given the joint position reference, computed by the IK algorithm over the user-generated Cartesian commands, a compliant feed-forward controller that combines a friction model with a PID error compensation was used to generate torque commands, as in [16]. The compliant controller makes the robot safer in case of an unexpected physical contact with the user or the surrounding objects.

2.2 Gesture Recognition and Motion Following

User tracking was implemented using the OpenNITM and NiTETM open source libraries. The position of the user's joints was provided in the camera frame of reference, which was used for both gesture recognition and hands motion tracking. No transformation of the joints positions to the robot frame of reference was necessary because joints relative displacements were used.

Switching between two robot states, "EXECUTE" and "FOLLOW", was initiated by raising the right hand. A voice command could have been used instead of gesture recognition, however, we decided to use this simple implementation to avoid possible deployment issues. Moreover, the initial tests showed no considerable effect in terms of delay when stopping the robot at a desired position.

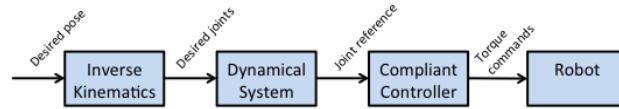


Fig. 1: Robot control diagram



Fig. 2: A user guiding dual-arm robot using hands motions tracked by Kinect camera. Left: the initial trajectory is shown in green. Right: red arrows represent the motions of both user hands and robot end-effectors in the teleoperation mode.

Hand-motion following consisted of tracking the position of each hand and reproducing the motion in real time with the corresponding robot's end-effector in a two-dimensional horizontal plane above the tabletop, as shown in Fig. 2. When switching to the "FOLLOW" state, the robot would store the initial position of a user's hand $\mathbf{p}_0 = (x_0, y_0)$ and the position of the corresponding end-effector, i.e. the cut point $\mathbf{p}_{cut} = (x_{cut}, y_{cut})$. All subsequent displacements of a hand were then reproduced by the end-effector according to following:

$$\mathbf{p}_{goal} = \mathbf{p}_{cut} + (\mathbf{p}_{hand} - \mathbf{p}_0) \quad (1)$$

where $\mathbf{p}_{hand} = (x_{hand}, y_{hand})$ is the new hand position detected by the camera with respect to the initial hand position, and $\mathbf{p}_{goal} = (x_{goal}, y_{goal})$ is the new goal sent to the robot's end-effector.

2.3 User Intervention

The implementation of the proposed user intervention is shown in Algorithm 1. The user was expected to observe the robot while performing the task, stop it at a desired position by raising the right hand and then, within the next 2 seconds, place his or her left hand (single-arm task) or both hands (dual-arm task) in a pose similar to holding a wheel when driving a car. All subsequent user's hands motions were then tracked and reproduced by the robot. To stop the intervention, the user would once more raise the right hand and relieve the control back to the robot. This action would also store the robot end-effector position as the final point of the modified segment.

A 2s delay was introduced (see line 6 in Algorithm 1) to allow the user to place the hands in a comfortable starting pose. Initial trajectory points were indexed and had associated timestamps to always ensure the identical execution of the trajectory. Cut and connect points were stored with their indexes. The cut point was stored as the trajectory point in which the robot was found when the user raised the right hand for the first time. The calculation of the connect point required additional computation, because the users were obviously not able to

Algorithm 1 User intervention

```

1: set state to "EXECUTE";
2: go to first trajectory point;
3: while in "EXECUTE" state and not end of trajectory do
4:   go to next trajectory point;
5:   if hand raised then
6:     set state to "FOLLOW"; // after a 2s delay
7:     store trajectory cut point;
8:     while in "FOLLOW" state do
9:       follow user's hands;
10:      if hand raised then
11:        set state to "EXECUTE";
12:      end if
13:    end while
14:    find closest trajectory connect point;
15:  end if
16: end while

```

precisely bring back the robot to its initial trajectory after modifying its segment. Therefore, the connect point was computed as the trajectory point that was closest to the point where the user relieved the control back to the robot, which occurred during the second raise-hand gesture. In the first experiment this calculation is performed for a single-arm robot; however, in the second experiment involving a dual-arm robot, the connect points are computed for each arm and the index of the one closer to the corresponding end-effector was selected for both arms.

The result of the user intervention is a new robot trajectory made of the unmodified trajectory segment preceding the cut point, a modified trajectory segment between the cut and connect points, and the final unmodified trajectory segment that follows after the connect point.

3 Methodology

3.1 Hardware

The proposed algorithms were implemented on two Barrett's 7-DOF Whole Arm Manipulator (WAM) robots (see Fig. 2). A first-generation Microsoft Kinect camera, which was used for user tracking, has a detection range from 0.8m to 4m with a vertical viewing angle of 43° and the horizontal viewing angle of 57°. It provides depth images at the resolution of 640 × 480 pixels at the maximal frame rate of 30fps. A desktop PC with Ubuntu 12.04 was used in this work, and it was powered by an Intel quad-core Q9550 @ 2.83GHz CPU with 4GB of RAM.

3.2 Experimental Setup

User tests were performed with a group of 12 participants, a mixed group of 6 female and 6 male adults of age between 25 and 42. All participants had a university degree (7 engineers, 1 computer scientist, 1 physicist, 1 astrophysicist, 1 biologist, and 1 pharmacist) and no previous experience in robotics. Each participant performed 10 trials of each experiment. The participants were divided into two subgroups of 6, applying different order of experiments for counterbalancing.

Each participant was asked to perform two experiments, which consisted of the same task but for different robot settings¹, i.e. a single-arm and dual-arm robot. Before the experiments, the researcher conducting the study described the WAM robot and the Microsoft Kinect camera features, and the experiment procedure to the participant. The procedure consisted of: 1) researcher's demo of the interaction with the dual-arm robot, 2) test trial by the participant, 3) two experiments performed by the participant, and 4) filling of the questionnaire about the performed experiments. The participants were assured that they were not within the robot workspace and that the experiments were safe. The researcher was present during the experiments to monitor the correct functioning of the robot, but he was seated behind the user and did not interfere.

In the first experimental setup a single-arm robot was programmed to perform a linear motion at a constant height above the table, from the point $A1 = (0.5m, 0.4m)$ to the point $B1 = (0.5m, -0.4m)$ in the robot reference frame. An empty plastic bottle was placed as an obstacle on the table at the point $C1 = (0.5m, 0m)$. Without a user intervention, the bottle would be knocked over by the robot. It was explained to the user that the goal of the experiment was to minimally modify the initial robot trajectory such that the robot end-effector avoids collision with the bottle.

In the second experimental setup, both robot arms simultaneously performed the same task described for the single-arm robot, by moving along the parallel lines above the table. The first robot followed the linear trajectory between the points $A1$ and $B1$, while the second robot followed the linear trajectory between the point $A2 = (0.5m, -0.4m)$ and the point $B2 = (0.5m, 0.4m)$, defined in each robot's frame of reference, respectively. Two plastic bottles were placed on the tabletop at the points $C1$ and point $C2 = (0.5m, 0m)$. As for the first experiment, the goal was to guide the robot end-effectors around the bottles. In both experiments, the robot end-effectors had fixed orientation facing downwards and their distance from the tabletop was constant during the whole experiment.

3.3 Performance and Workload Measures

The following metrics were used to assess the performance and the workload of the participants: task success, task completion time, trajectory length, and the NASA-TLX questionnaire.

Task success was measured as a function of knocked down obstacles during the task execution, and was computed as follows:

¹ Additional material at: <http://www.iri.upc.edu/groups/perception/adapt2>

$$\begin{aligned} S_{i,1} &= (1 - N_{ob}) \cdot 100\%, \forall i = 1 \dots 10 \\ S_{i,2} &= (1 - N_{ob}/2) \cdot 100\%, \forall i = 1 \dots 10 \end{aligned} \quad (2)$$

where $S_{i,1}$ and $S_{i,2}$ are the task success values for the single-arm and dual-arm robot tasks, respectively, N_{ob} is the number of knocked down obstacles, and i is the number of the trial. Higher task success values were indicators of a better performance.

The task completion time was the total time that one or both robots needed to reach the end point of the manipulation task. Lower completion times were indicators of a better performance.

The robot trajectory length was computed as the sum of distances between the Cartesian trajectory points in the horizontal (x, y) plane of the robot frame of reference:

$$D_{i,1} = \sum_{k=2}^N \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2}, \forall i = 1 \dots 10 \quad (3)$$

where x_k and y_k are the k -th trajectory point coordinates in the robot frame of reference, N is the number of trajectory points, and i is the number of the trial. In case of the dual arm task, the trajectory length was computed as a mean of trajectories' lengths from both robot arms. Shorter trajectories were an indicator of a better performance.

Participants completed the raw NASA-TLX questionnaire after each experiment. The questionnaire enables the collection of six dimensions of workload ranging from 0 to 100, and was used to assess the overall participant workload when performing the experiments, similarly as in [13]. The overall workload is computed as the average of the above-mentioned six dimensions.

3.4 Data Analysis

A two-way repeated measure ANOVA analysis was conducted on two within-subject variables (number of trials and robot setting), using order of experiments as between-subject variable. The number of trials was divided in ten levels and the robot setting in two levels (single arm and dual arm). Statistical significance was computed for all three performance metrics described in Section 3.3. The effect on the overall participant workload was computed in the two-way repeated measure ANOVA test using the robot setting as within-subject variable and the order of experiments as between-subject variable. We considered the results as significant for $p < 0.05$.

4 Results and Discussion

4.1 Performance Analysis

The results of the first ANOVA test show that there was a statistically significant effect of the robot setting on the task completion time, $F(1, 10) = 12.779$, $p = 0.005$, and robot trajectory length, $F(1, 10) = 7.97$, $p = 0.018$. In case of

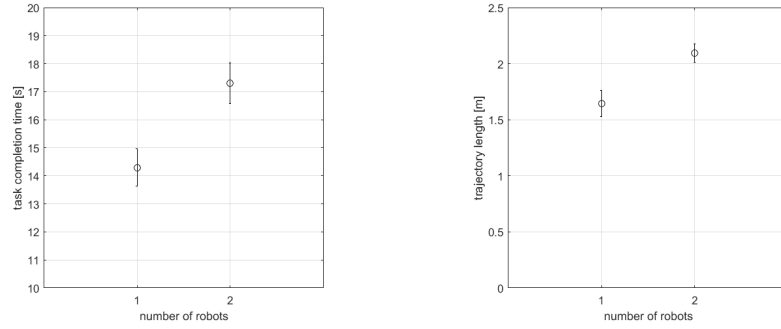


Fig. 3: Effect of the robot setting on: (a) task completion time, and (b) trajectory length. Error bars represent the standard error of the mean.

the interaction with the dual-arm robot, the participants needed more time to complete the task (Fig. 3a) and also guided the robots along longer trajectories (Fig. 3b). This could be explained by the fact that, in case of the dual-arm robot, the attention of the participants was divided between two arms, which reduced speed and accuracy of the robot guidance.

The between-subject analysis showed that the order of experiments had a statistically high significant effect on the task success, $F(1, 10) = 12.214$, $p = 0.006$, and a statistically significant effect on the task completion time, $F(1, 10) = 5.348$, $p = 0.043$, which is an indicator of the learning effect from previously using the robot regardless of the initial robot setting; it also demonstrates user's ability to adapt when switching from single-arm to dual-arm robot setting, and vice-versa. The robot trajectory length was not affected by the order in which experiments were performed.

The results also show that there was a statistically high significant effect of the number of trials on the task completion time, $F(9, 90) = 2.881$, $p = 0.005$, regardless of the order in which the experiments were performed, which again indicates the user's ability to learn over time to more efficiently interact with the robot. The number of trials did not have a statistically significant impact on task success and robot trajectory length.

4.2 Overall Workload

The analysis of the interaction of the within-subject variable and the between-subject variable, i.e. the order of experiments and the robot setting, shows a statistically significant joint effect on the participants' overall workload, $F(1, 10) = 5.482$, $p = 0.041$. It can be noted from the overall workload mean values shown in Fig. 4 that the participants experienced less initial workload if the first experiment was performed with a single arm; however, the workload drop was also lower during the second experiment using the same robot setting. But when the experiments started with the dual-arm robot the initial workload was much

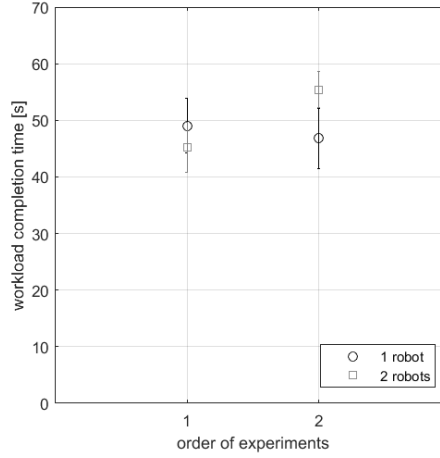


Fig. 4: Effect of the interaction of the robot setting and the order of experiments (*Order* = 1: starting with 1 robot; *Order* = 2: starting with 2 robots) on the overall workload. Error bars represent the standard error of the mean.

higher and the workload drop was more drastic during the second experiment, indicating that participants felt a huge relief for moving from a more difficult condition. It should also be noted that in both cases the second experiment produced less workload than the first experiment, which indicates the existence of the user's learning factor.

5 Conclusion

The study compared the performance and the workload of a group of lay users interacting with a single-arm and dual-arm robots performing the motion for the same task. The experiments were designed to evaluate several aspects of the user experience with the two robot settings. Although the experimental setup involved a simple trajectory adaptation task, it demonstrated that lay users with no background in robotics were able to successfully perform the task and improve over time while reducing the overall workload. The proposed approach can be extended to other application domains and more complex tasks, such as assisting the users in the activities of daily living, with either one or two robot arms, which will be a part of future work. Finally, the results suggest that a user could be introduced to interaction with a dual-arm robot by initially interacting with only one robot arm.

References

1. I. Leite, C. Martinho and A. Paiva, Social robots for long-term interaction: A survey. *International Journal of Social Robotics*, vol. 5, no. 2, pp. 291-308, Jan. 2013.

2. M. Mason and M.C. Lopes, Robot self-initiative and personalization by learning through repeated interactions. *6th ACM/IEEE International Conference on Human-Robot Interaction (HRI'11)*, pp. 433-440, 2011.
3. C. Torras, Service robots for citizens of the future, *European Review*, vol. 24, no. 1, pp. 17-30, 2016.
4. J.-J. Cabibihan, H. Javed, M. Ang Jr. and S.M. Aljunied, Why robots? A survey on the roles and benefits of social robots in the therapy of children with autism, *International Journal of Social Robotics*, vol. 5, no. 4, pp. 593-618, Nov. 2013.
5. H. Robinson, B. MacDonald and E. Broadbent, The role of healthcare robots for older people at home: A review, *International Journal of Social Robotics*, vol. 6, no. 4, pp. 575-591, Nov. 2014.
6. C.-A. Smarr, T.L. Mitzner, J.M. Beer, A. Prakash, T.L. Chen, C.C. Kemp and W.A. Rogers, Domestic Robots for Older Adults: Attitudes, Preferences, and Potential *International Journal of Social Robotics*, vol. 6, no. 2, pp. 229-247, Apr. 2014.
7. C.C. Kemp, A. Edsinger and E. Torres-Jara, Challenges for robot manipulation in human environments [Grand Challenges of Robotics], *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 20-29, Mar. 2007.
8. A. Edsinger and C.C. Kemp, Two arms are better than one: A behavior-based control system for assistive bimanual manipulation, *Recent Progress in Robotics: Viable Robotic Service to Human, Lecture Notes in Control and Information Sciences*, Springer, Heidelberg, vol. 370, pp. 345-355, 2008.
9. B.D. Argall, S. Chernova, M. Veloso and B. Browning, A survey of robot learning from demonstration, *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469-483, May 2009.
10. A. Jevtić, A. Colomé, G. Alenyà and C. Torras, Robot adaptation through user intervention and reinforcement learning. *Publication under review*. Available online: <http://www.iri.upc.edu/groups/perception/adapt.pdf>
11. M. Niolescu and M. Mataric, Learning and interacting in human-robot domains, *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 31, no. 5, pp. 419-430, 2001.
12. S. Mitra and T. Acharya, Gesture recognition: A survey, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 3, pp. 311-324, 2007.
13. A. Jevtić, G. Doisy, Y. Parmet and Y. Edan, Comparison of interaction modalities for mobile indoor robot guidance: Direct physical interaction, person following, and pointing control, *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 6, pp. 653-663, 2015.
14. F. Husain, A. Colomé, B. Dellen, G. Alenyà and C. Torras, Realtime tracking and grasping of a moving object from range video, *2014 IEEE International Conference on Robotics and Automation*, pp. 2617-2622, 2014.
15. A. Colomé and C. Torras, Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 944-955, 2015.
16. A. Colomé, A. Planells and C. Torras, A friction-model-based framework for reinforcement learning of robotic tasks in non-rigid environments, *IEEE International Conference on Robotics and Automation*, pp. 5649-5654, 2015.