2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
September 14-18, 2014. Chicago, IL, USA,

Implementation and experimental validation of Dynamic Movement Primitives for object handover*

Miguel Prada[1], Anthony Remazeilles[1], Ansgar Koene[2] and Satoshi Endo[3]

Abstract— **This article presents the implementation and experimental validation of a control system dedicated to human robot physical interaction during object handovers. Our control law defines the trajectory of the robotic arm towards a previously unknown handover location based on the Dynamic Movement Primitives formalism adapted for human robot object handover. The control law was deployed on a Kuka Light-Weight Arm equipped with the Azzurra anthropomorphic hand. We employed an industrial-like setting involving three different human postures for object handover in order to evaluate the performance and user experience of our control law and its generalizability. The evaluation was conducted over 1000 object handover trials between the human and robot partners, and the kinematic data and subjective experience were gathered for each trial. The outcomes of this evaluation validate the current implementation and guide the next steps towards more efficient and fluid human robot interaction.**

I. INTRODUCTION

It is broadly accepted that robots will have a great impact on human activities in a near future [6]. The recent advancements on robotic technologies foresee a strong collaboration between humans and robots for the achievement of domestic and industrial tasks. As stated in the European strategic agenda for robotics, such collaboration relies on natural physical interaction between the two partners, which highlights the importance of human-robot object handover actions [1].

Previously, various research has addressed the importance of fluid interaction using a human-robot object handover as an example task [21]. One of the main streams in the literature addresses an estimation of the most preferred handover location where the robot should bring the object at. In [2] cost functions are defined to maximize the comfort of the handover, considering both the mobile base positioning and the arm configuration. In [18], the A^* algorithm permits to estimate the best handover location based on a $3D$ cost map which reflects three cost functions respectively focused on the safety, visibility and comfort criteria.

Once the handover location is defined, several techniques are available to define the robot motion towards this site. In [8], different motion profiles (minimum-jerk, trapezoidal velocity profiles) are experimentally evaluated with humans, and, based on the results, a decoupled minimum jerk trajectory generator is proposed [9]. In [19], the Soft Motion Trajectory planner provides an active control of maximum jerks, accelerations and velocities.

As highlighted in [2], [21], the use of such off-line planning-like approaches permits to get a good reachability score but lacks adaptability to the human motion (the robotic hand may move too far and push the the human hand for example), and may not provide motions or handover configurations that feel natural to the human. Another alternative consists of explicitly learning good exchange strategies from the human. In [2], a learning technique is proposed in which the users are involved in the definition of the preferred handover configurations. In [23], the robot motions are defined from a human motion database. An adapted sliding-window search is proposed to find the most likely states in the database related to the current human position, and the related human receiver postures are then combined to define the robot motion. In this last work, the capability of appropriately responding to the human behavior implicitly depends on the database completeness, since the robot does not have mechanisms to handle novel user motions.

In [12] a reactive mechanism is proposed in which the robot is driven through visual servoing towards the human hand location. The robot velocities, however, are proportional to the distance to the object which tends to produce motion speeds that could be perceived as unsafe by the users, in particular at the beginning of the interaction.

The approach we present in this paper permits to produce robot motions inspired by human behaviors while maintaining a feedback mechanism to provide flexibility. This duality aspect is nicely handled by the concept of Dynamic Movement Primitives (DMP) that permits to combine a shape-driven control strategy (to reproduce human-like trajectories) with a goal driven strategy (to ensure an on line convergence towards the real observed handover location).

We previously introduced the theoretical specialization of the DMP mechanism to object exchange in [14] and compared it in simulation with human behaviors in [15]. In this paper we extensively describe an implementation of this control law onto the robotic platform designed in the context of the European project Coglaboration. The next section summarizes the key aspect of the control law itself. Section III describes this implementation onto the Light-Weight Robot (LWR) equipped with the Azzurra anthropomorphic hand. Section IV describes the experiments we performed to evaluate the control law during object handovers with human

*This work was supported by the Coglaboration project within the 7th Framework Programme of the European Union - Cognitive Sstems and Robotics, contract number FP7-28788, see also http://www.coglaboration.eu

[1]Dept. of Assitive Technologies, Helath Division of Tecnalia Research and Innovation, Parque Tecnologico de San Sebastian, E-2009 San Sebastian name.surname@tecnalia.com

[2]Dept. of Psychology, Univ. of Birmingham, B152TT, UK a.r.koene@bham.ac.uk

[3]Institute for Information-Oriented Control, Technische Universität München, Munich, Germany s.endo@tum.de

partners using an industrial scenario, and highlights the main outcomes of that evaluation.

II. ROBOT CONTROL LAW

This section highlights the main characteristics of the DMP formalism and its extension for the case of human robot object handover (see [14], [15] for more detail).

A. Dynamic Movement Primitives formalism

The DMP concept was initially introduced for trajectory learning and reproduction by Ijspeert in [10]. Since then, several works were derived from the basic formalism, to handle obstacle avoidance [7] or to boostrap reinforcement learning methods for example in [4], [22].

Let $x(t)$ be a one dimension trajectory that transits from $x(t_0) = x_0$ to $x(t_f) = g$. The DMP formalism consists of modeling the reference trajectory with a second order dynamical system named transformation system:

$$\tau\dot{v} = (1-s)K(g-x) + sK\left(\frac{f(s)}{s} + x_0 - x\right) - Dv \quad (1a)$$

$$\tau\dot{x} = v \quad (1b)$$

The first term $K(g-x)$ can be seen as a *goal attractor* that ensures the convergence towards the desired position g. The second term $K\left(\frac{f(s)}{s} + x_0 - x\right)$ acts like a *shape attractor* stimulating the system to follow the reference motion pattern. The contribution of these two terms are weighed by the *phase variable* s which exponentially decreases from 1 towards 0 following the canonical system:

$$\tau\dot{s} = -\alpha s \quad (2)$$

Thus, the shape attractor plays a predominant role at the beginning of the movement when $s \approx 1$ while the goal attractor becomes predominant towards the end of the trajectory when $s \to 0$.

The shape attractor encapsulates the learned trajectory through the forcing term f that represents a non linear function as a sum of weighted functions:

$$f(s) = \frac{\sum_{i=1}^{N} \psi_i(s) w_i}{\sum_{i=1}^{N} \psi_i(s)} s, \text{ with } \psi_i(s) = \exp(-h_i(s-c_i)^2) \quad (3)$$

The forcing term does not explicitly depend on time but on the phase variable s instead. The system is thus time-scalable by adjusting the time constant τ.

B. Specialization for object handover

The main benefit of using the DMP formalism for learning and reproducing a trajectory is that during the reproduction of the learned motion pattern, the target position g in (1a) can be modified to adjust the final position of the trajectory while maintaining dynamics similar to the reference trajectory. This is particularly convenient for our specific application in which the handover location, i.e. the target position, is not known a priori. Thus, in contrast with the work performed in [18], we do not perform an initial estimation of the hand-over location which is affected by the quality of the estimator and inflexible against unpredictable human behaivor. Instead, we propose to consider the target location g as the human hand location, therefore, constantly updated during a handover. The previously mentioned dominance shift in attractors permits to limit the impact of the target position on the early phase of the motion (when the human hand is far from the future handover site), and progressively increases its influence when the two partners get closer to the handover site, with the transformation system becoming linear when $s \to 0$.

Nevertheless, the goal attractor in the basic DMP model starts dominating the control law too early for this approach to work. As demonstrated in [14] the exponential evolution of the phase variable quickly decreases from 1, so that the purely shape-driven motion is only present during a limited period of time. In addition, the basic DMP model does not permit to adjust when and how fast the weight switches between the two phases. This is considered as a limitation for our object handover application, where human behavior studies have shown that the transition in between the feedforward (or shape-driven) and the feedback (or goal-driven) strategies tends to occur at the middle of the reaching motion [5].

In order to account for this issue, we proposed in [14] to decouple the weights applied to each of the terms in the transformation system from the phase variable, and to use instead an arbitrary function of the phase variable to compute these weights:

$$\tau\dot{v} = (1-w_g)(f_w + x_0 - x) + w_g K(g-x) - Dv \quad (4a)$$

$$\tau\dot{x} = v \quad (4b)$$

$$\tau\dot{s} = -\alpha s, \quad (4c)$$

where $f_w(s)$ is now defined as:

$$f_w(s) = \frac{\sum_{i=1}^{N} \psi_i(s) w_i}{\sum_{i=1}^{N} \psi_i(s)}. \quad (5)$$

In our implementation, the weighing function $w_g(t)$ is selected to be a sigmoid function related to the Cumulative Distribution Function (CDF) of the Normal distribution:

$$w_g(t) = 0.5\left[1 + erf\left(\frac{t-\mu}{\sigma\sqrt{2}}\right)\right], \quad (6)$$

where erf stands for the Gauss error function. This weighing function relies on two parameters, the mean μ of the Normal distribution and the standard deviation σ of the distribution, that respectively permit to decide when the transition occurs and how long it lasts. With these two parameters we can easily control when the goal-attractor starts influencing the generated motion.

Furthermore, in order to better adjust the convergence to a varying goal, we also proposed to incorporate a velocity feedback term to the transformation system:

$$\tau\dot{v} = (1-w_g)(f_w + x_0 - x) + w_g[K(g-x) + K_v\dot{g}] - Dv \quad (7)$$

The evolution of this last system is the one driving the robot motion according to the perception feedback that adjusts the goal location. We will now describe how this model is implemented onto our robotic platform.

III. IMPLEMENTATION

A. Overall architecture

The previously described technique has been used to design a controller for the hardware platform presented on Fig. 1, the LWR from Kuka equipped with the Azzurra hand from Prensilia [3]. The perception is performed with the Kinect sensor mounted above the arm. The software system has been implemented using mainly ROS [16] framework (the low level arm connexion also involves Orocos [20]), and distributed in components as shown in Fig. 2.

The *Perception* component is in charge of providing the pose of the target location with respect to a fixed reference frame, using the output of a Kinect sensor. Depending on the handover direction, that pose can represent the pose of the human hand (robot $\rightarrow$ human) or the pose of the object in the human hand (human $\rightarrow$ robot). A detailed description of the algorithms used by this component is beyond the scope of the current paper.

Interface to the robotic hardware is managed by the *LWR FRI*, and the *IH2 Azzurra* components. The first component interfaces with the robot arm and is, in turn, composed by an Orocos subsystem which deals with the low-level interface to the robot using the network-based *Fast Research Interface* (FRI); details are given in III-B. The later component communicates with the Prensilia hand as described in III-C.

The *DMP Controller* component receives data from the perceptual layer and from the robot interface components, and computes the motion commands sent to the robot in order to reproduce the learned trajectories (see III-D).

In order to coordinate the operation of the aforementioned components, the *CogLab Master* component runs a state machine which selects the appropriate configurations for each component at every step of the system operation. Section III-E gives an overview of this coordinator.

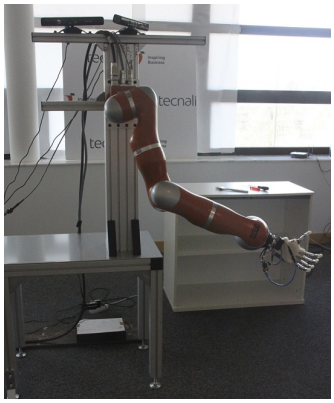

Fig. 1: Azzurra hand from Prensilia mounted onto the Kuka Light Weight Arm

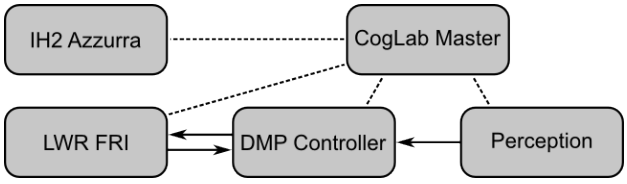

Fig. 2: Simplified software component architecture

B. LWR FRI component

As described above, the *LWR FRI* ROS node is in itself a composition of Orocos components. The main component of this subsystem is the one actually interfacing with the robot. The robotic arm is controlled using the FRI offered by the *Kuka Robot Controller* (KRC) [17]. This interface offers several control modes through a simple protocol using UDP messages over an Ethernet connection. Joint position is selected as the control input since it allows complete control over the robot posture, while avoiding the complexities of a joint torque-based controller, which does not benefit our approach of imitating observed human motions.

Since the FRI requires that the motion commands received are reachable within the selected control cycle period (set to $100Hz$ in our case). To meet this requirement, we have incorporated an interpolator component in this subsystem. It uses a state of the art on-line trajectory generation algorithm, based on the work of Kröger [11]. This component accepts velocity set-points and generates attainable commands for the robot, which respect specific velocity, acceleration and jerk limits at the target communication frequency. These velocity commands can be safely integrated into positions reachable in a single FRI period and directly transmitted to the KRC.

C. Prensilia IH2 Azzurra component

The IH2 Azzurra anthropomorphic hand by Prensilia embeds internal microprocessors which perform the low-level finger motor control. The communication protocol permits to send through a serial link high-level commands which allow to set position targets for each finger individually or to automatically execute pre-defined grasps. A helper Python library was developed and later used to build a ROS node to interact with the hand using an Actionlib-based interface.

D. DMP based controller implementation

Currently only Cartesian position data is available both in the learning dataset used to train the DMP, and in the output of the *Perception* component. Therefore, we propose to learn position-only trajectories from human demonstration and to encode and reproduce those with the DMP system. Afterward we augment this position trajectory with a desired Cartesian orientation interpolation from the initial orientation of the robot end effector to the target orientation based on a pre-defined per-object delivery orientation.

This component reads the position of the pose output from the *Perception* component and uses this to set the g parameter in the DMP system in (7). Integration of the system for a fixed time step generates new desired Cartesian position and velocity for the robot. Desired orientation and rotational velocity are computed separately with an *ad-hoc* strategy which lets the end-effector move solidarily with the robot's forearm during the initial phases of the motion, and uses Bezier curves to make it converge to the desired orientation during the final approach phase.

In order to combine these position and orientation commands we have implemented a system based on prioritised

task Jacobians, instead of combining them into a single end-effector pose and transforming them to joint space commands using the geometric Jacobian. The benefits of using such a scheme are twofold. On one hand, it allows dealing with close-to-singular configurations by removing the offending constraint while maintaining the feasible sub-tasks. On the other hand, including the generic task support in the inverse kinematics algorithm provides greater flexibility if different types of tasks need to be taken into consideration (e.g. to keep an object upright, which only constrains two of the three rotational degrees of freedom).

Since the tasks are resolved to joint velocities, each task setpoint needs to account for a drift. This is straightforward to formulate in the position case, where the adjusted linear velocity can be computed as:

$$\mathbf{v}_{des} = \mathbf{v}_{dmp} + K_{pos}\left(\mathbf{x}_{dmp} - \mathbf{x}_{rob}\right) \tag{8}$$

where $\mathbf{v}_{des}$ is the velocity for which the instantaneous inverse kinematics will be computed, $\mathbf{x}_{dmp}$ and $\mathbf{v}_{dmp}$ are the target Cartesian position and translational velocity generated by the DMP system, $\mathbf{x}_{rob}$ is the current Cartesian position of the robot, and K_{pos} is a gain which determines how fast the system should try to eliminate the tracking error.

The rotational velocity component is computed in a similar manner:

$$\omega_{des} = \omega_{tgt} + K_{rot} diff(\mathbf{R}_{tgt}, \mathbf{R}_{rob}), \tag{9}$$

where $\mathbf{R}_{tgt}$ and ω_{tgt} are respectively the current desired rotation and angular velocities that are interpolated along time as previously mentioned to converge towards the desired final hand orientation at the exchange location. In this case a $diff$ function provided by KDL is used to compute the rotational velocity which makes an object initially oriented as in $\mathbf{R}_{rob}$ converge to $\mathbf{R}_{tgt}$ in one second.

The following step consists of transforming these new desired linear and rotational velocities of the end-effector to robot joint velocities. In order to do this, the two task Jacobians are directly obtained by dividing the geometric Jacobian of the robot in two submatrices, one relating the joint velocities with the linear end-effector velocity, and the other to the rotational end-effector velocity:

$$\begin{bmatrix} \mathbf{v} \\ \omega \end{bmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_{lin}(\mathbf{q}) \\ \mathbf{J}_{rot}(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} \tag{10}$$

The joint velocity command $\dot{\mathbf{q}}_{lin}$ which satisfies the desired linear Cartesian velocity is computed as:

$$\dot{\mathbf{q}}_{lin} = \mathbf{J}_{lin}^{+}\mathbf{v}_{des} \tag{11}$$

where $\mathbf{J}_{lin}^{+}$ stands for the Moore-Penrose pseudo-inverse of $\mathbf{J}_{lin}$ (dropping the dependency on $\mathbf{q}$ for compactness).

In order to respect the desired linear velocity achieved by $\mathbf{q}_{lin}$, we need to project this second velocity to the null space of the linear velocity Jacobian by using:

$$\dot{\mathbf{q}}_{rot} = \left(\mathbf{J}_{rot}\mathbf{P}_{lin}\right)^{+}\left(\omega_{des} - \mathbf{J}_{rot}\dot{\mathbf{q}}_{lin}\right), \tag{12}$$

where the null space projection matrix $\mathbf{P}_{lin}$ of $\mathbf{J}_{lin}$ is:

$$\mathbf{P}_{lin} = \mathbf{I}_7 - \mathbf{J}_{lin}^{+}\mathbf{J}_{lin} \tag{13}$$

In (12) the term $\mathbf{J}_{rot}\dot{\mathbf{q}}_{lin}$ is inserted to take into consideration the rotational velocity induced by the translational part.

Finally, the joint velocity command that is sent to the *LWR FRI* subsystem is:

$$\dot{\mathbf{q}}_{rob} = \dot{\mathbf{q}}_{lin} + \dot{\mathbf{q}}_{rot}, \tag{14}$$

where $\dot{\mathbf{q}}_{lin}$ and $\dot{\mathbf{q}}_{rot}$ are respectively deduced from eq. (11) and (12). This command is interpolated and integrated to joint positions by the *LWR FRI* component before being transmitted to the robot.

E. CogLab Master

The *Coglab Master* coordinates the behavior of the whole system. It runs a hierarchical state machine (using the SMACH Python library) which at the top level can transition between two idle states, one for the robot resting with an object in its hand, and the other for the robot empty handed. Transitioning between these two idle states is done through a series of states containing sub-state machines which execute different actions: hand over an object from the robot to the human and vice versa, take/put the object from a storage location, and two convenience actions to allow the experimenter to manually put/remove an object from the robot's hand.

The two most relevant states are those dealing with the object handover. For instance, when the robot is required to hand over an object to a person, the state machine first requests the object's relevant data from a database and configures the *DMP Controller* component with the appropriate parameters: primitive to use, motion speed, appropriate delivery pose with respect to the human hand, etc. Then, it simultaneously starts the human hand tracking behavior of the *Perception* component and triggers the start of the DMP-controlled motion towards the human partner's desired handover location. When the robot end-effector gets close to the actual handover location, it monitors the torques on the robot joints to detect when the human has firmly grasped the object. It then commands the *IH2 Azzurra* component to open the hand and release the object. Subsequently, the coordinator commands the *DMP Controller* component to move the robot back to a resting position using a secondary control mode where a motion to a target joint-state position is interpolated and commanded to the robot.

In addition to reacting to the system state, the *CogLab Master* component can also react to a series of events which can be sent from a custom experimenter's GUI built with HTML and a javascript/ROS bridge component. This allows the person running the experiment to trigger specific transitions of the state machine and to configure some of the parameters involved (e.g. even if the torque-based contact detection has been used in some of the experiments, in the trials focused on the approach phase that are described here, the contact trigger was manually provided by the operator).

IV. EXPERIMENTAL VALIDATION

In the context of the CogLaboration project [13], we selected two experimental scenarios in which a robot could

provide assistance to a person: a industrial scenario, and a domestic scenario. In the analysis we present here, we focus on the first scenario, studying the velocity aspects and verbal vs vision based triggering.

A. Experiment description

1) Experiment design: The design of the experiment was inspired by the industrial scenario of a car mechanic with a robot assistant and was implemented with three configurations, with different human postures:

- 'Engine Bay' with the mechanic bent forward over the engine, able to view the handover, generally reaching to the right and slightly backwards with only slight movement restriction (Fig. 3 a).
- 'Hydraulic Lift' with the mechanic under a car on a hydraulic lift, reaching above the head, free to observe the handover, generally reaching to the right side with only slightly impaired freedom of movement (Fig. 3 b).
- 'Lying under car' with the machanic lying on the back under the car, limiting the field of view and the range of arm movement (Fig. 3 c)).

2) Experiment procedure: Each scenario configuration was tested on a separate day to avoid frequent reconfigurations of the setup and minimize participant fatigue. We tested with 7 (4 male, 3 female) naïve participants, recruited from Tecnalia staff, 5 of whom had little or no prior experience of interacting with robots. In each scenario a base case, and variations in two factors were tested: velocity and movement triggering.

To test the effect of speed-accuracy trade-off on the user experience the Robot movements were executed at 5 different speeds ranging between approximately 1 and 1/4 times as fast as an average human reaching movement calculated from an existing data set of a similar setup.

Another manipulation was a switch from the default vision based robot movement triggering (mocked by the operator manually starting the robot motion), used in the speed trials, to verbal triggering in which participants provided a verbal "GO" command.

Each session started with 5 base handovers (medium speed, approximately 1/2 human speed, and visual triggering), to familiarize the participant with the robot and the evaluation protocol. All other trials were performed in a random order with each of the 5 speeds repeated 3 times and the verbal triggering repeated 5 times. Each trial consisted of the following sequence.

1) Delivery from robot to human ($R \rightarrow H$)
   - Human requests object by reaching to robot
   - Robot brings the object to the human
   - Human takes object to task area
   - The participant evaluates the handover
2) Retrieval from human to robot ($H \rightarrow R$)
   - Human holds object out to the robot
   - Robot reaches to the object
   - Robot takes the object back to itself
   - Human evaluates the handover

The evaluation after each handover was provided in four criteria by rating the following statements:

- Q1: It was *easy* to receive the object
- Q2: I was *satisfied* with the interaction
- Q3: The interaction was *comfortable*
- Q4: I felt *safe* during the interaction

The participants were asked to enter an evaluation score between 1 (fully disagree) and 9 (fully agree) using a touch-screen. Once all trials were completed, the participant was interviewed to provide additional feedback and qualitative evaluation of their experiences.

In addition to the qualitative evaluation, we also recorded (i) the location of the human hand as a function of time, providing movement kinematics, (ii) the articular pose of the robot as well as the measured efforts per joint, and (iii) the timing of the events during the handover procedure (e.g. motion start, end).

3) Experiment setup & equipment: The car mechanic scenario was simulated using the setup shown in Fig. 4. The metal frame defined the task area, i.e. the car. The task surface was moved from the waist height ('Engine Bay'), to above the participant's head ('Hydraulic Lift', shown in Fig. 4 left image). For the 'Lying under the car' configuration participants lay on a height adjusted bed, to accommodate the workspace limitations of the robot (Fig. 4 right image). The work frame and bed were positioned such that participants were 100-125cm from the robot. This allowed their outstretched hand to be inside the robot's workspace while their body remained safely beyond reach of the robot. A Liberty magnetic motion tracking system (Polhemus Inc., Vermont, USA) with four magnetic markers, was used to record the position and orientation of the participant's arm at 240Hz during the object handover. The magnetic markers were placed onto the participant's right arm (one on the shoulder, one on the back of the hand, one on the thumb

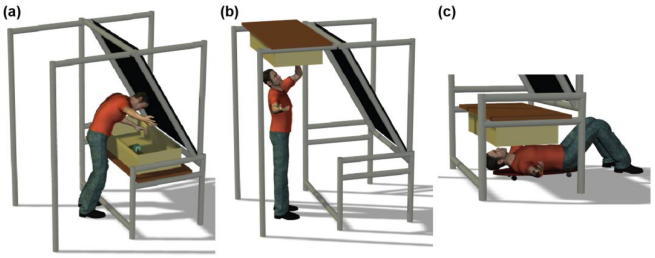

Fig. 3: Human postures in the 3 car mechanic configurations: a) Engine Bay; b) Hydraulic Lift; c) Lying under Car


Fig. 4: Setup: Standing configuration for 'Engine Bay' and 'Hydraulic Lift' (left image) and Lying under the car configuration (right image)

and one on the index finger). This system was used only for off-line analysis, as the control system only used the Kinect sensor for real-time visual input.

The object used for the handovers was a plastic flashlight which was always grasped using a cylindrical power grasp.

B. Main outcomes of the experiments

In order to evaluate the performance of our control system for Human-Robot Interaction, the results of the experiments are evaluated not only in terms of success in achieving object handovers but also in term of the subjective experiences of the participants.

1) Success rate: Table I summarizes the overall object handover success rate. Notably the experiment configuration had only a minor impact on the success rate despite the large differences in the participants sensory and movement capacities in the three configurations.

Success rates are grouped per behavioural conditions on Table II. As it shows, the lowest success rate was observed in the trials with the highest robot velocity (Sp. 1).

Handover failures in the high speed conditions were primarily due to the torque limit of the Kuka LWR triggering the safety stop in the mid-motion. On the other hand, the failures in the verbal triggered trials were caused by participants instructing the robot to start moving before they had moved their own hand into the operating space of the robot.

2) Subjective user experience: Preliminary analysis of the subjective rating responses showed a skewed response distribution with more than 60% of ratings scored above 8 and less than 10% of ratings at 5 or less (Fig. 5). These high ratings show that the participants had a generally positive impression of their interaction with the robot. The skewed response distribution however violates a basic assumption of parametric statistical tests (i.e. normal-distributed data), thus requiring the use of non-parametric tests for statistical evaluation.

Comparison between the subjective rating histograms of the five robot movement speeds (see color coding) revealed that faster movement trials (Sp. 1, same speed as human) had more 9 ratings for Comfort, Satisfaction and Ease than the slower movement trials (Sp. 5, 1/4 speed of a human). For the 'Safe' rating however this pattern was reversed, with more 9 rated trials for slower movements. Comparison between the verbally triggered and the equivalent visually triggered movements (Sp. 3) suggests that participants found the interactions slightly Easier, more Satisfying and more Comfortable when they could explicitly trigger robot movement through verbal commands (higher fraction of 9 ratings) but that visual or

	Total	Eng. Bay	Hydr. Lift	Lying u. Car
$R \rightarrow H$	$495/525 = 94\%$	$163/175$	$167/175$	$165/175$
$H \rightarrow R$	$501/525 = 95\%$	$165/175$	$173/175$	$163/175$

TABLE I: Successful handovers per scenario

	Sp. 1	Sp. 2	Sp. 3	Sp. 4	Sp. 5	Verbal
$R \rightarrow H$	$51/63$	$60/63$	$58/63$	$62/63$	$61/63$	$98/105$
$H \rightarrow R$	$62/63$	$61/63$	$58/63$	$62/63$	$63/63$	$100/105$

TABLE II: Successful handovers per behavioural condition

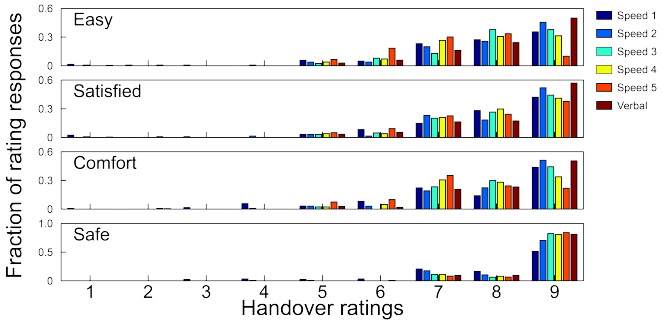

Fig. 5: Histograms showing fraction of trials that received each quality rating. The colors indicate the trial type.

verbal triggering had no impact on their sense of Safety. These results are consistent with the statements participants made during the post-session interviews. Kruskal-Wallis non-parametric ANOVA tests confirmed that effects of movement speed (Table III) and verbal triggering (Table IV) manipulations on subjective experience of the human users were statistically significant. In addition to confirming that the speed manipulation significantly affected Ease, Satisfaction, Comfort and Safety ratings, we note that speed had the least effect in the 'Lying under Car' configuration. A possible reason may have been the reduced ability to see the robot motion and the reduced speed of the human movements in this posture. This illustrates the complex nature of the perceived handover quality, which was affected by more than just pure robot velocity. For the verbally triggered condition, compared to the visually triggered medium speed trials, there was generally a stronger effect on the Ease, Satisfaction and Comfort ratings and a much weaker effect on the Safe rating. In the Engine Bay and Hydraulic Lift configurations, the effect of verbal triggering is much more significant in the $H \rightarrow R$ handovers than in the $R \rightarrow H$ interactions. In contrast to the speed manipulation, the effect of the verbal trigger manipulation is highly significant in the Lying under Car configuration, possibly due to a greater appreciation for the sense of control gained by 'verbal triggering' when participants are restricted by the lying posture.

Kruskal-Wallis p-values	Easy	Satisfied	Comfort.	Safe
Engine Bay $R \rightarrow H$	.015	.205	.001	.000
Hydo. Lift $R \rightarrow H$	.000	.000	.000	.892
Lying u. Car $R \rightarrow H$	.026	.200	.605	.186
Engine Bay $H \rightarrow R$	.014	.722	.001	.000
Hydo. Lift $H \rightarrow R$	.000	.002	.000	.478
Lying u. Car $H \rightarrow R$	.160	.935	.082	.152

TABLE III: Speed manipulations effect on qualitative rating responses (Kruskal-Wallis ANOVA, $df = 4, 16 < N < 21$).

Kruskal-Wallis p-values	Easy	Satisfied	Comfort.	Safe
Engine Bay $R \rightarrow H$	.890	.165	.133	.954
Hydo. Lift $R \rightarrow H$	.035	.133	.193	.088
Lying u. Car $R \rightarrow H$	.000	.000	.009	.909
Engine Bay $H \rightarrow R$	.000	.000	.001	.049
Hydo. Lift $H \rightarrow R$	.030	.055	.519	.137
Lying u. Car $H \rightarrow R$	.000	.000	.000	.178

TABLE IV: Verbal/visual trigger manipulation effect on qualitative rating responses (Kruskal-Wallis ANOVA, $df = 1, 26 < N < 33$).

3) Quantitative properties of handover movement: We considered the following quantitative motion measures.

1) Two temporal aspects of handover motion
   a) End time difference (end time of robot movement minus end time of human reaching)
   b) Peak velocity time difference, (time of robot peak velocity minus time of human peak velocity)
2) Two spatial aspects of handover motion
   a) Movement distance difference (distance between start and end of human reaching minus distance between start and end of the robot movement)
   b) End position error (distance between the robot and the human reach end-points, before the human made an optional final adjustment for picking up/placing the object from/in the robot hand)
3) One spatio-temporal aspect of handover motion
   a) Robot peak-velocity

The means of each of these measures are summarized in Fig. 6, showing that the movement were very similar for both handover directions, $R \rightarrow H$ and $H \rightarrow R$. The experiment configuration, however, did change some of the movement properties, especially the 'End position error'. Fig. 7 provides a more detailed summary of the movement properties by showing the mean changes as a function of the speed or triggering manipulation used on a trial.

4) Quality perception and the Speed-Accuracy trade-off: Based on participant comments that suggested a preference for faster robot movements we analysed the correlation between the qualitative rating changes and quantitative measures of the human and robot movements for the speed manipulation trials in order to identify if changes in perceived interaction quality were primarily due to timing or spatial accuracy aspects of the movements.

The Spearman rank-correlation was computed between each of the four qualitative ratings and the five movement measures for both the $R \rightarrow H$ and the $H \rightarrow R$ handovers. The

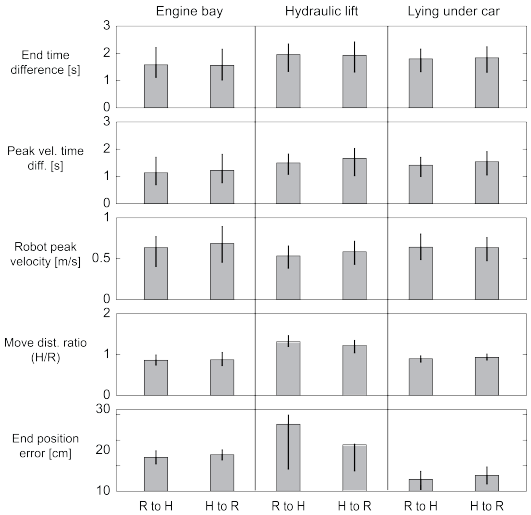

Fig. 6: Mean results for the movement measures, averaged across participants and behaviour manipulations (error-bars indicate 25th and 75th percentiles).

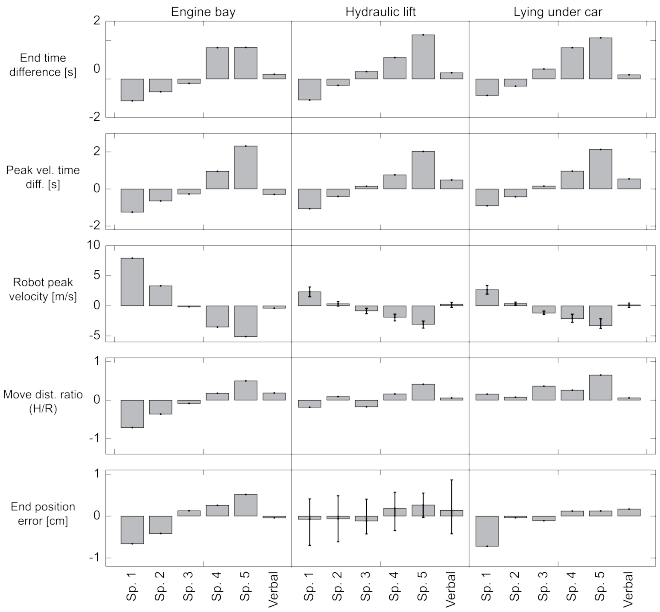

Fig. 7: Mean effect of speed/trigger manipulation, expressed as deviation from the participants' mean values (error-bars indicate 25th and 75th percentiles).

results are summarized in Fig. 8. The three subplots provide the results for the 'Engine Bay' (top), 'Hydraulic Lift' (middle) and 'Lying under the car' (bottom) configurations respectively. The abscissa lists the quantitative measures (e.g. 'End time difference') grouped by measures relating to 'timing performance' on the left (first 2) and measures relating to 'spatial performance' on the right (last 2). The ordinate lists the four qualitative ratings. Colors indicate the value of the Spearman rank-correlation coefficient (ρ), with warm (red) colours for positive correlations and cold (blue) colours for negative correlations (see legend on right). Correlations of $|\rho| < 0.2$ are considered not significantly different from zero ($p > 0.1$). The Safe rating and the Easy-Comfortable-Satisfied ratings have generally opposing relationships to the quantitative behavior properties, i.e. participants were more satisfied but felt less safe for faster robot movements and shorter end-point differences. The 'timing performance'

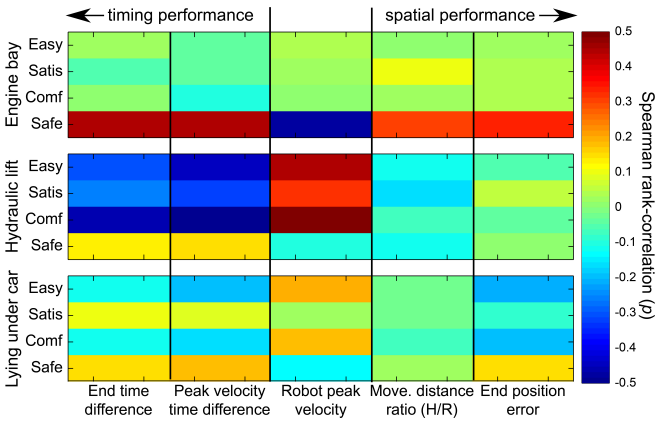

Fig. 8: Correlation analysis between the qualitative and quantitative results for the speed manipulation trials.

related measures were more strongly correlated with the qualitative ratings than the 'spatial performance' measures, indicating a willingness by participants to engage in a speed-accuracy trade-off in favour of rapid interactions. Overall, the strongest correlations between qualitative ratings and movement measures were found in the 'Hydraulic Lift' configuration for the Easy-Comfortable-Satisfied ratings whereas the 'Safe' ratings showed strongest correlations in the 'Engine Bay' configuration. 'Lying on the Bed' yielded the weakest correlations, probably because the reduced visibility and reduced freedom of movement dominated the participants' experiences. 'End position error' had its strongest (negative) correlations in the 'Lying under the car' configuration. Users may prefer a greater weight on speed when they themselves have the space to make compensation movements, while they prefer to wait for more accurate robot behaviour when they themselves are constrained in their movement space.

V. CONCLUSION

In this article we presented the implementation of a robot arm control based on the Dynamic Movement Primitives formalism adapted to human robot object handover. The proposed control law has been implemented onto a LWR robot equipped with the Azzurra hand and experimentally tested through more than 1000 handovers with a range of configurations and behavior manipulations. The results of the behavioral experiments validate the proposed control mode with a success rate of more than 95%. Although the setup of the experiment only covers a specific scenario and object for handover, the proposed method is not limited to any of them, and further experimentation will evaluate the system in a different scenario and with other objects requiring different grasping modes.

The next steps towards fluent object handovers in that framework will consist of increasing the autonomy and adaptability of the control strategy. In that line we would like to extend the situation awareness by permitting the system to automatically detect when the user launches an interaction procedure, or when the same user decides to stop it for any particular reason.

The adaptability will be addressed in terms of object affordance on the one hand and individual preferences in handover location and speed on the other hand. Concerning object afforcances we would like, through perception, to better understand how the user would like to grasp an object or how he would like to deliver it. Concerning individual preference, the experiment showed that the humans implicitly adapted their behavior to the robotic partner, we would like the robotic system to perform a similar effort, taking into account, through the accumulation of exchanges, the preferred handover location or exchange speed of the human.

Finally, some preliminary internal experiments (not evaluated here) on the automatic detection of the contact in between the two agents, through forces analysis at the hand wrist, to trigger the object grasping or release show promising results for possibility of using such data to recognize the handover events.

REFERENCES

[1] R. Bischoff and T. Guhl. The strategic research agenda for robotics in europe [industrial activities]. *Robotics Automation Magazine, IEEE*, 17(1):15–16, 2010.

[2] M. Cakmak, S. Srinivasa, M. Lee, J. Forlizzi, and S. Kiesler. Human preferences for robot-human hand-over configurations. In *IEEE/RSJ IROS*, pages 1986–1993, San Francisco, CA, Sept. 2011.

[3] C. Cipriani, M. Controzzi, and M. Carrozza. The smarthand transradial prosthesis. *Journal of NeuroEngineering and Rehabilitation*, 8(32), May 2011.

[4] C. Daniel, G. Neumann, and J. Peters. Learning concurrent motor skills in versatile solution spaces. In *IEEE/RSJ IROS*, pages 3591–3597, Vilamoura, Portugal, Oct. 2012.

[5] M. Desmurget and S. Grafton. Forward modeling allows feedback control for fast reaching movements. *Trends in cognitive sciences*, 4(11):423–431, Nov. 2000.

[6] B. Gates. A robot in every home. *Scientific American*, 296(1):44–51, 2007.

[7] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal. Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance. In *IEEE ICRA*, pages 2587–2592, Kobe, Japan, May 2009.

[8] M. Huber, C. Lenz, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer. Human preferences in industrial human-robot interactions. In *Int. Workshop on Cognition for Technical Systems*, Munich, Germany, 2008.

[9] M. Huber, H. Radrich, C. Wendt, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer. Evaluation of a novel biologically inspired trajectory generator in human-robot interaction. In *IEEE RO-MAN*, pages 639–644, Toyama, Japan, Sept. 2009.

[10] A. Ijspeert, J. Nakanishi, T. Shibata, and S. Schaal. Nonlinear dynamical systems for imitation with humanoid robots. *IEEE-RAS Int. Conf. on Humanoid Robots*, 2001.

[11] T. Kröger. Opening the Door to New Sensor-based Robot Applications The Reflexxes Motion Libraries. In *IEEE ICRA*, Shangai, China, 2011.

[12] Vincenzo Micelli, Kyle Strabala, and Siddhartha Srinivasa. Perception and control challenges for effective human-robot handoffs. In *RSS, workshop on RGB-D*, Los Angeles, California, June 2011.

[13] G. Pegman. Report on scenarios, tasks and goals. Technical Report Deliverable D2.10, European Project Coglaboration, FP7-ICT-7-2.1 (No. 287888), 2012.

[14] M. Prada and A. Remazeilles. Dynamic Movement Primitives for Human Robot Interaction. In *IEEE/RSJ IROS, workshop on Robot Motion Planning: online, reactive and in Real-time*, Algarve, Portugal, 2012.

[15] M. Prada, A. Remazeilles, A. Koene, and S. Endo. Dynamic movement primitives for human-robot interaction: Comparison with human behavioral observation. In *IEEE/RSJ IROS*, pages 1168–1175, Tokyo, Japan, 2013.

[16] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Y. Ng. ROS: an open-source Robot Operating System. In *IEEE ICRA, Open-Source Software workshop*, Kobe, Japan, May 2009.

[17] G. Schereiber, A. Stemme, and R. Bischoff. The Fast Research Interface for the Kuka Lightweight robot. In *IEEE ICRA, workshop on Innovative Robot Control Architectures for Demanding (research) applications*, pages 15–21, Anchorage, Alaska, May 2010.

[18] E. Sisbot, L. Marin-Urias, R. Alami, and T. Simeon. A human aware mobile robot motion planner. *IEEE Trans. on Robotics*, 23(5), 2007.

[19] E. Sisbot, L. Marin-Urias, X. Broquere, D. Sidobre, and R. Alami. Synthesizing robot motions adapted to human presence. *International Journal of Social Robotics*, 2(3), 2010.

[20] P. Soetens. *A Software Framework for Real-Time and Distributed Robot and Machine Control*. PhD thesis, Katholieke Universiteit Leuven, Belgium, May 2006.

[21] K. Strabala, M. Lee, A. Dragan, J. Forlizzi, S. Srinivasa, M. Cakmak, and V. Micelli. Towards seamless human-robot handovers. *Journal of Human-Robot Interaction*, 2(1), 2013.

[22] F. Stulp. Adaptive exploration for continual reinforcement learning. In *IEEE/RSJ IROS*, pages 1631–1636, Vilamoura, Portugal, Oct. 2012.

[23] K. Yamane, M. Revfi, and T. Asfour. Synthesizing object receiving motions of humanoid robots with human motion database. In *IEEE ICRA*, pages 1629–1636, Karlsruhe, Germany, May 2013.