Learning Dynamic Robot-to-Robot Object Handover

Yansong Wu [*] **Lingyun Chen** [1] [*] **Ignacio Perez Mahiques** [*]
Zhenshan Bing [**] **Fan Wu** [*] **Alois Knoll** [**] **Sami Haddadin**[1] [*]

[*] *The Chair of Robotics and Systems Intelligence, MIRMI - Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, Germany,* [1] *also with the Centre for Tactile Internet with Human-in-the-Loop (CeTI)*
[**] *The Chair of Robotics, Artificial Intelligence and Real-time Systems, Technical University of Munich, Germany.*

Abstract: Object handover is an essential skill for collaborative robots in both services robotics and manufacturing scenarios. Most previous works were conducted from the perspective of human-robot interaction. The object handover between robots for collaborative task execution, aiming at optimizing time efficiency with human-like smooth behaviour, has not been extensively addressed. In this work, we propose a skill framework based on variable impedance control and dynamic motion primitives to optimize not only the motion trajectories and variable impedance, but also the timing of hand actions during dynamic motion. The effectiveness of the proposed framework is evaluated on a real dual-arm robot system under two handover scenarios with different constraints on the timing of hand actions. The experiment results demonstrate significant time efficiency improvement with reduction of the execution time by 9.4% and 23.7%, compared with accelerating the motion speed of the demonstrated handover. Furthermore, it can be observed that the robot successfully learned dynamic object handover without requiring transfer action to be triggered after both hands stop and remain still. In addition, in the second experiment, it is shown that the object can be transferred even without ensuring firm contact, which indicates that object handover is possible to be realized by throwing-like motion.

Keywords: Intelligent robotics, autonomous robotic systems, robots manipulators, dynamic movement primitive, object handover.

1. INTRODUCTION

Object handover is a crucial interaction skill for robots to provide services in shared environments with humans or perform tasks collaboratively among robot teams. Increasing research efforts have been attracted into investigating this open problem from different perspectives. The field of physical human-robot interaction (pHRI) aims at achieving safe, reliable and ergonomic handover of objects between human-to-robot (H2R) or robot-to-human (R2H). To ensure the safe grasping and passing of objects, the robot, be it a giver or receiver, must be able to detect the grasping status of itself or humans (Konstantinova et al. (2017)), which requires tactile sensors mounted on the end-effector or estimators using joint torque sensing (De Luca et al. (2006)). Grip forces (Chan et al. (2012)) and human mobility (Ardón et al. (2021)) need to be taken into account when designing handover controllers, as well. Human studies on handover forces indicate giver-receiver coupling and suggest designing a handover controller based on the measurement of the load share (Medina et al. (2016)).

Many works have extensively studied H2R and R2H handovers as summarized in Ortenzi et al. (2021). One of the recent examples is the reactive handover controller designed in Costanzo et al. (2021) through combining a human-to-robot and a robot-to-human task strategy to prevent slippage. Nevertheless, robot-to-robot (R2R) handover is less addressed. In this work, we aim at developing a method that can generate highly efficient, generalizable, fast, and natural handover behaviours for robots. Unlike humans that can hand over objects in a seamless manner, naïvely programmed robots have clear transition phases in a sequence of actions. Specifically, during the robot's hand movement, the absence of corresponding smooth arm motion always results in a stiff and unnatural behaviour. For many multi-robot collaboration applications which desire to improve production efficiency, e.g., assembly tasks in robotised factories, equipping robots with fluid handover skills is of critical importance.

To address this problem, in this work, we propose a framework that exploits goal-directed dynamic movements and variable impedance control to enable phase-less and faster handover motion. Moreover, unlike most works assuming that the object must be held in the hands throughout the transfer, we allow the object to be thrown from the giver to the receiver to further improve the time efficiency. The proposed framework is then evaluated by physical experiments performed on a dual-arm robot consisting of two 7-DOF arms. Experimental results show that the handover task completion time can be reduced effectively by combined optimization of motion, timing and impedance shaping. [1]

[⋆] Corresponding Author: Yansong Wu `yansong.wu@tum.de`

[1] The video summarizing the experiments is available via the link: `https://www.youtube.com/watch?v=CT53Oxrusp0`

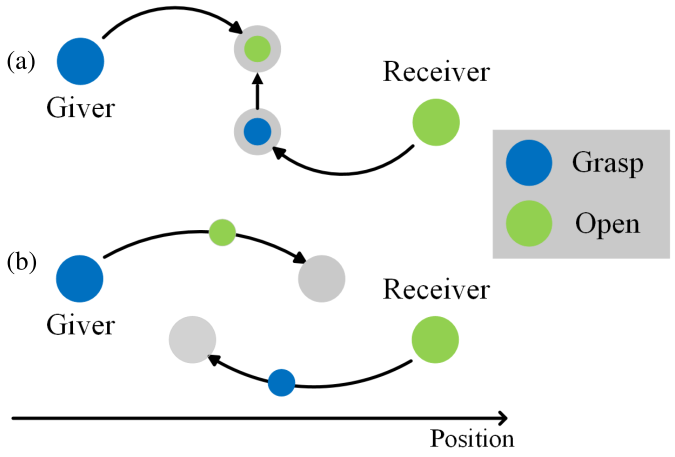

Fig. 1. Conceptual diagram of dynamic robot-to-robot handover. Shown in the figure are (a) handover programmed in a conventional sequential manner, and (b) dynamic handover in which hand actions can be triggered during motion and variable impedance can be exploited.

2. METHOD

As discussed previously, one of the most important features of human handover behaviours is that the passing phase often occurs during motion smoothly without a clear phase transition. Conventionally, a robot handover consists of two phases, *i.e.,* the pre-handover phase and the physical exchange phase (Ortenzi et al. (2021)). The physical exchange phase (object transferring phase) usually begins after the robots come into steady states, as depicted in Fig. 1(a). By doing this, the robots have no way to perform a handover as fluidly and fast as what can be observed in human movements.

A possible way to generate such behaviour is to represent movement using a dynamical system and exploit its goal-directed dynamic behaviour. In particular, as illustrated in Fig. 1(b), the goal positions of both giver and receiver (represented by the gray circles) can deviate from the original position as in case (a) so that the hand actions, *i.e.,* grasping and releasing (represented by the blue and green circles) can be triggered during the motion. Overall, our framework is designed to address the following aspects:

(1) The arms and hands motion is highly coupled. The grasping and releasing motion is allowed to be triggered during dynamic motion without both the giver and receiver stopping and remaining at zero velocity.
(2) The robots are equipped with *compliance skill.* In other words, variable impedance control is used to improve interaction behaviours.
(3) The execution time of the handover is considered a crucial optimization goal. Furthermore, the external force and the robot joint torque are taken into account, as well.

2.1 Variable impedance control

The Franka Emika robot (Haddadin et al. (2022)) is a torque-controlled 7-DOF robot. On the task level, it can be treated as a rigid system, since the elasticity at the joint level is handled at the low-level joint control.

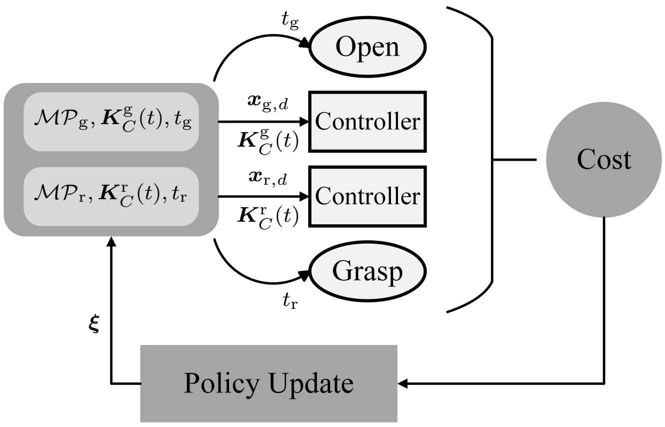
t_g
Open
$\mathcal{MP}_\mathrm{g}, \boldsymbol{K}_C^\mathrm{g}(t), t_\mathrm{g}$
$\boldsymbol{x}_{\mathrm{g},d}$
$\boldsymbol{K}_C^\mathrm{g}(t)$
Controller
$\mathcal{MP}_\mathrm{r}, \boldsymbol{K}_C^\mathrm{r}(t), t_\mathrm{r}$
$\boldsymbol{x}_{\mathrm{r},d}$
$\boldsymbol{K}_C^\mathrm{r}(t)$
Controller
Grasp
t_r
Cost
$\boldsymbol{\xi}$
Policy Update

Fig. 2. Overview of the proposed framework.

The well-known dynamics of a second-order rigid body robot model with n-DoFs is of the form (Ficuciello et al. (2015)):

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}_m + \boldsymbol{\tau}_{ext} \tag{1}$$

where $\boldsymbol{q} \in \mathbb{R}^n$ is the joint state. $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ corresponds to the mass matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is the Coriolis matrix and $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the gravity vector. The motor torque (control input) and external torque are denoted by $\boldsymbol{\tau}_m \in \mathbb{R}^n$ and $\boldsymbol{\tau}_{ext} \in \mathbb{R}^n$, respectively.

To track a trajectory using Cartesian impedance control, the well-known control rule (without inertial shaping) introduced by Albu-Schaffer et al. (2003) is:

$$\begin{aligned} \boldsymbol{\tau}_m =& \boldsymbol{J}^T(\boldsymbol{q})[\boldsymbol{K}_C\tilde{\boldsymbol{x}} + \boldsymbol{D}_C\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{M}_C(\boldsymbol{q})\ddot{\boldsymbol{x}}_d + \\ & \boldsymbol{C}_C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{x}}_d] + \boldsymbol{g}(\boldsymbol{q}), \end{aligned} \tag{2}$$

$$\tilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{f}(\boldsymbol{q}) = \boldsymbol{x}_d - \boldsymbol{x}, \tag{3}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4}$$

where $\boldsymbol{K}_C \in \mathbb{R}^{6\times 6}$ is the desired stiffness matrix, $\boldsymbol{D}_C \in \mathbb{R}^{6\times 6}$ is the damping matrix. The mapping $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^6$ denotes the forward kinematics. $\boldsymbol{M}_C(\boldsymbol{q})$ and $\boldsymbol{C}_C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are the inertial and the Coriolis matrix. $\boldsymbol{J}(\boldsymbol{q})$ refers to the contact Jacobian. For variable impedance control, $\boldsymbol{K}_C$ and $\boldsymbol{D}_C$ represent time dependent function $\boldsymbol{K}_C(t)$ and $\boldsymbol{D}_C(t)$.

2.2 Parameterised movement trajectory and stiffness profile

Trajectory Dynamic Movement Primitive (DMP) is widely-used formalization for encoding trajectories (Ijspeert et al. (2002), Schaal (2006), Ijspeert et al. (2013)), based on the idea of modelling movements using dynamical systems. In the following section, a brief recap of the approach utilized in our framework is provided. Further details in representing a Cartesian space movement primitive ($\mathcal{MP}$) can be found in Ude et al. (2014).

(i) Position DMP:

$$\tau\dot{\boldsymbol{p}} = \boldsymbol{z} \tag{5}$$

$$\tau\dot{\boldsymbol{z}} = \boldsymbol{\alpha}_z(\boldsymbol{\beta}_z(\boldsymbol{g}_p - \boldsymbol{p}) - \boldsymbol{z}) + s\boldsymbol{A}^\top \boldsymbol{f}^p(s) \tag{6}$$

$$\tau\dot{s} = -\alpha_s s \tag{7}$$

$$f_m^p(s) = \frac{\sum_{i=1}^N \psi_i(s)}{\sum_{i=1}^N \psi_i(s)} w_{m,i} \tag{8}$$

$$\psi_i(s) = \exp(-\frac{(s-c_i)^2}{2\sigma_i^2}) \tag{9}$$

where $\boldsymbol{p}$ is the position of the End-Effector (EE). $\tau > 0$ represents the duration and $\boldsymbol{g}_p$ is the goal attractor. The dynamics of $\boldsymbol{p}$ is regulated by a dynamical system which behaves like a mass-spring-damper model, with gains determined by $\boldsymbol{\alpha}_z, \boldsymbol{\beta}_z$. $\boldsymbol{f}^p(s)$ is a forcing term manipulating the shape of the trajectory. It is a function in phase of variable s, whose dynamics makes it asymptotically converge to 0, in a rate controlled by α_s. As a result, the efficacy of forcing term gradually decays to zero. This behaviour is purposely designed by Ijspeert et al. (2002, 2013) to enhance convergence of $\boldsymbol{p}$ to the goal position $\boldsymbol{g}_p$. In addition to s, $\boldsymbol{A}$ is added to the forcing term to scale it according to the movement distance, where the m-th element $a_m = g_{p,m} - p_m$ corresponds to m-th forcing element f^p_m. From (8) and (9), we can see that the forcing term is defined as the weighted sum of a set of N basis functions, of which each is an exponential function defined by centre point c_i and width factor σ_i.

(ii) Orientation DMP:

$$\tau \dot{\boldsymbol{q}} = \frac{1}{2} \boldsymbol{\eta} * \boldsymbol{q} \tag{10}$$

$$\tau \dot{\boldsymbol{\eta}} = \boldsymbol{\alpha}_z(\boldsymbol{\beta}_z 2 \log(\boldsymbol{g}_o * \bar{\boldsymbol{q}}) - \eta) + s\boldsymbol{A}^{\mathsf{T}} \boldsymbol{f}^o(s) \tag{11}$$

where the orientation is represented by unit quaternions $\boldsymbol{q} = v + \boldsymbol{u} \in \mathbb{S}^3$. $\boldsymbol{g}_o$ denotes the goal quaternion orientation. The notion bar denotes the quaternion conjugation, and $*$ denotes the quaternions product.

Note that, in our framework, both shaping parameter $\boldsymbol{w}_f$ and goal $\boldsymbol{g} = [\boldsymbol{g}_p, \boldsymbol{g}_o]$ can be tuned to adjust the shape of DMP. However, since in this paper our evaluation focuses on the handover phase, we intentionally choose only $\boldsymbol{g}$ as policy parameter for comparative experiments, which will be detailed later in the section 3.

Stiffness profile The time-varying stiffness profile is represented via Locally Weighted Regression (LWR), a classic approach to tackle function approximation problem introduced by Atkeson et al. (1997). For a LWR consists of K linear weighted regression functions, it fits a set of $\boldsymbol{A}_k$ to minimize the weighted cost function (16) on the dataset $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{1 \times N}$.

$$\widetilde{\phi}_k(\boldsymbol{x}_n) = \exp(-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)) \tag{12}$$

$$\phi_k(\boldsymbol{x}_n) = \frac{\widetilde{\phi}_k(\boldsymbol{x}_n)}{\sum_{i=1}^K \widetilde{\phi}_i(\boldsymbol{x}_n)} \tag{13}$$

$$\boldsymbol{W}_k = \mathrm{diag}(\phi_k(x_1), \phi_k(x_2), ..., \phi_k(x_N)) \tag{14}$$

$$\hat{\boldsymbol{Y}} = \sum_{k=1}^K \boldsymbol{W}_k \boldsymbol{X} \boldsymbol{A}_k \tag{15}$$

$$J = (\boldsymbol{Y} - \hat{\boldsymbol{Y}})(\boldsymbol{Y} - \hat{\boldsymbol{Y}})^T \tag{16}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the parameters of the k-th radial basis functions (RBF) $\phi_k(\boldsymbol{x}_n)$, and $\hat{\boldsymbol{Y}}$ refers to the estimated output of the LWR. In this framework, the stiffness profile is modified through adjusting the intercepts of the linear regression functions and the RBFs' centers.

2.3 *Policy improvement by evolution strategies*

Exploration and evaluation The exploration phase generates K unconstrained perturbations in policy parameter space for K roll-outs. These perturbations are assumed to obey the multivariate Gaussian distribution $\tilde{\boldsymbol{\xi}}_k \sim \mathcal{N}(\boldsymbol{\xi}, \boldsymbol{\Sigma}_\epsilon), (k = 1, ..., K)$, where $\boldsymbol{\xi}$ donates the centre of the distribution and $\boldsymbol{\Sigma}_\epsilon$ indicates the covariance matrix. Then, the box constraints $\boldsymbol{\xi}_{\max}$ and $\boldsymbol{\xi}_{\min}$ are applied while mapping the perturbation $\tilde{\boldsymbol{\xi}}_k$ to the parameter vector $\boldsymbol{\xi}_k$, which represents the whole policy of the k-th roll-out.

$$\boldsymbol{\xi}_k = \min(\max(\tilde{\boldsymbol{\xi}}_k, \boldsymbol{\xi}_{\min}), \boldsymbol{\xi}_{\max}) \tag{17}$$

where min and max are evaluated element-wise.

In human-robot handovers, two metrics are used to analyze the handover's quality based on the human experience, *i.e.,* objective task performance metrics and subjective metrics. The task performance metrics selected by Ortenzi et al. (2021) focus on the success rate, total handover time, and receiver's task completion time; Similarly, the proposed subjective metrics concern about fluency, trust in the robot, and working alliance. In a robot-to-robot handover, the metrics used are only objective task performance metrics due to the absence of human participation. Other task performance metrics often used are the internal wrench norm, which are the forces and torques that are unnecessary to accomplish the task, and the duration of the passing phase, which is defined as the time while both agents share the object's load (Medina et al. (2016)).

In our framework, the performance of each roll-out is evaluated with the cost function (18), which includes the following aspects: (i) **Task accomplishment**: The bool value Φ_k equals to 0 for a completed handover and 1 for an unsuccessful trial. (ii) **Speed factor**: In the roll-out, the motion of the robot arms is executed s_f times faster than the demonstrated handover. (iii) **Completion time**: t_{ref} and $t_{k,g}$ represent the demonstrated giver's releasing time (reference value) and the real releasing time of the k-th trial. The giver's releasing time is leveraged to mark the end of the handover motion. (iv) **External wrench**: $F_{k,t}$ corresponds to the estimated external wrench, and F_r is the reference value. (v) **Joint torque**: $\boldsymbol{\tau}_{k,t}$ and $\boldsymbol{\tau}_r$ indicate the internal joint torque and its reference value. The variables w_c, w_s, w_t, w_F and w_τ are weighting coefficients, accordingly.

$$\begin{aligned} \boldsymbol{w} &= [w_c, w_s, w_t, w_F, w_\tau] \\ \boldsymbol{c} &= [\Phi_k, e^{1-s_f}, \frac{t_{k,g}}{t_{ref}}, \sum ||\boldsymbol{F}_{t_{k,t}} \oslash \boldsymbol{F}_r||^2, \sum ||\boldsymbol{\tau}_{t_{k,t}} \oslash \boldsymbol{\tau}_r||^2] \\ J_k &= \boldsymbol{w}\boldsymbol{c}^T \end{aligned} \tag{18}$$

where the operator $\oslash$ donates the element-wise division.

Policy update The policy update steps (19)-(23) are based on the PI^{BB} algorithm introduced by Stulp and Sigaud (2012).

$$\tilde{J}_k = \frac{J_k - \min(\{J_k\})}{\max(\{J_k\}) - \min(\{J_k\})} \tag{19}$$

$$P_k = \frac{\exp(-c\tilde{J}_k)}{\sum_{i=1}^K \exp(-c\tilde{J}_i)} \tag{20}$$

$$\boldsymbol{\xi} \leftarrow \sum_{k=1}^K P_k \boldsymbol{\xi}_k \tag{21}$$

$$\boldsymbol{\Sigma}_\epsilon^{temp} = \sum_{k=1}^K P_k(\boldsymbol{\xi}_k - \boldsymbol{\xi})(\boldsymbol{\xi}_k - \boldsymbol{\xi})^T \tag{22}$$

$$\boldsymbol{\Sigma}_\epsilon \leftarrow \boldsymbol{\Sigma}_\epsilon + \gamma(\boldsymbol{\Sigma}_\epsilon^{temp} - \boldsymbol{\Sigma}_\epsilon) \tag{23}$$

First, the cost J_k is normalized according to their maximum and minimum by (19). The normalized cost $\tilde{J}_k$ is used to calculate probability P_k for k-th roll-out according to (20) , where $c > 0$ is a constant. [2] Then, the distribution is updated, according to the weighted averaging rule (21)-(23), where $\gamma \in (0, 1]$ is the applied learning rate while updating the covariance matrix.

Moreover, to achieve faster convergence and improve sample efficiency, another technique employed is sample reuse. After every update, the μ best samples among K roll-outs (at current iteration) are kept for the next update. In other words, from the second episode, only $K - \mu$ perturbations are generated in each sampling phase. The exploration, evaluation and policy update procedures are repeated until $\boldsymbol{\xi}$ converges or the system reaches maximum updates number N_{max}.

The above-mentioned policy updating algorithm has the structure of PI^2 (Stulp and Sigaud (2013)), but uses covariance matrix adaptation as found in the CMA-ES algorithm (Hansen and Ostermeier (2001)). It only uses the total reward received during the execution. Correspondingly, this policy improvement method is converted into a Black-Box Optimization (BBO) method and is proofed as a special case of CMA-ES by Stulp and Sigaud (2012).

2.4 *Policy parameters*

Before ending this section, a summary is provided and illustrated in Fig. 2:

(1) A variable impedance controller is utilized to track the trajectory with time-varying stiffness profile $\boldsymbol{K}_C(t)$ in Cartesian space.
(2) The reference trajectory $\mathcal{MP}$ is symbolized with DMP, whose goal position is tuned as a policy parameter.
(3) The time-varying stiffness profile $\boldsymbol{K}_C(t)$ is represented with a parameterized LWR.
(4) The time-related variables are also taken as policy parameters, including time for the receiver to grasp t_r, giver's releasing time t_g, and speed factor s_f for adjusting the overall trajectory execution.
(5) With the encoding rule introduced in section 3.2, a demonstrated handover is represented as a parameter vector $\boldsymbol{\xi}$. Vice versa, a parameter vector sampled from the distribution can be decoded into a set of continuous control commands for the robot to perform handover roll-out.

3. EXPERIMENTS

3.1 *Experiment setup*

As illustrated in Fig. 3, the experiment setup used in this paper is a dual-arm robot, that consists of two 7-DoF Franka Emika robots (Haddadin et al. (2022)). It is developed based on the humanoid robot *GARMI* (Tröbinger et al. (2021)). At each wrist of the robot, a qb-SoftHand (an anthropomorphic compliant robotic hand) is equipped as end-effector. Due to its well-designed transmission system, it can adapt to the grasped object shape with only one tendon and one motor (Bonilla et al. (2016), Catalano et al. (2014))). As for the software, the robots are controlled based on the Franka Control Interface (FCI) at 1 kHz and the hands' control commands are sent via the RS-485 serial communication interface. The developed control software runs on a computer (Intel Core i5-12600K CPU @ 4.50GHz) installed with Ubuntu 20.04 LTS and real-time kernel.

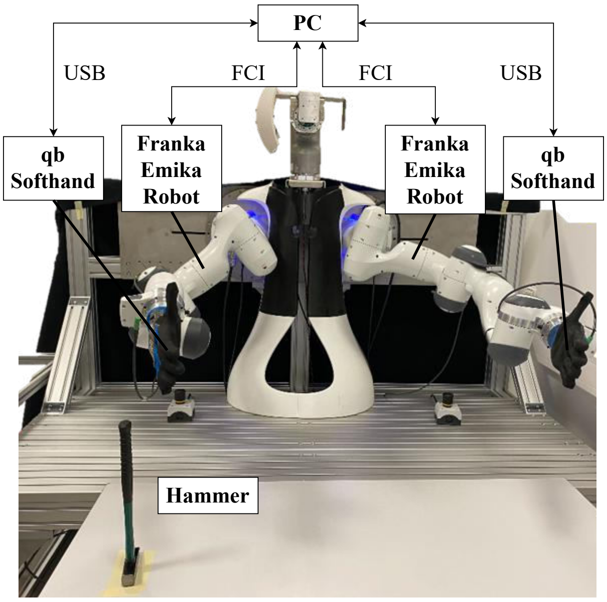

Fig. 3. Overview of experiment setup

3.2 *Experiment design*

To evaluate the effectiveness of the proposed method while completing the robot-to-robot handover task, experiments are designed based on the dual-arm robot mentioned in the section 3.1. During the experiment, the dual-arm robot learns to hand over an engineer's hammer from the right hand to its left hand with the proposed framework.

For each robot arm, the pose sequence of end-effector is fitted with DMP, which contains 10 radial basis function networks (RBFNs). Since we want to keep the shape of the trajectory, for each robot, only 7 parameters are picked to indicate its goal g of the DMP; In order to compress the dimension of parameter space, the Cartesian stiffness matrix of each robot is simplified as a diagonal matrix with $[K_p, K_p, K_p, K_o, K_o, K_o]$ as the diagonal elements. The K_p and K_o are fitted with the function approximator LWR based on the software package DMPBBO (Stulp and Raiola (2019)), respectively. Each approximator consists of 3 basis linear regression functions. The shape of the stiffness profile is modified by tuning the center of the basis functions and offsets of the linear regression functions. Consequently, 12 parameters are utilized to encode the stiffness profile of each robot arm; In addition, 3 extra parameters related to the time are required, in order to depict the movement time point of the hands and the speed factor of the trajectory (a factor to regulate the overall motion speed). As a result, the whole handover policy is represented by a parameter vector $\boldsymbol{\xi}$ with 41 elements. Moreover, the demonstration is preserved and accelerated to work as the baseline while evaluating the performance of the learnt skills, as well.

As for the robot hands' motion, we designed two strategies to achieve dynamic handover behaviour. **S1**: To guarantee safeness and robustness during the handover, the giver is only allowed to release the object after the object is fully grabbed by the receiver. Therefore, a constraint ($t_g \geqslant t_r + \Delta t_h$) is applied while generating samples of timing parameters; **S2**: On the contrary, here we remove the constraint on the timing of the hand motion t_r, t_g to make it possible for the robot to employ a throwing-like motion for object handover.

[2] In our implementation we choose $c = 10$.

Table 1. Parameters used during the experiments.

Property	Unit	Value
t_r, t_g (Demo)	s	6.834, 7.639
Δt_h	s	0.5
K_p, K_o (Demo)	N/m	1000, 100
$w_c, w_s, w_t, w_F, w_\tau$	-	0.35, 0.1, 0.3, 0.2, 0.05
N_{max}, K, μ	-	20, 12, 2
γ	-	0.8

Based on the designed strategies, two groups of experiments are implemented with the same initial conditions. Both experiments share the same initial parameter vectors that are generated from the demonstration. Through fitting the demonstrated trajectory with DMP, the shaping parameters and initial goal g of the DMP are obtained. As for the parameters related to the stiffness profile, they are obtained by representing the K_p and K_o with LWR. The initial receiver's grasping time t_r and giver's releasing time t_g are the same as the demonstrated handover. The initialization parameters used in the experiments are summarized in Table 1.

4. RESULTS

According to the designed strategy, the policy based on the strategy **S1** meets the convergence criteria[3] after 7 iterations and that based on **S2** converges after 8 updates. Then, the final policy is performed on the dual-arm and the recorded motion trajectories and measured external forces are used for evaluation. To evaluate the proposed framework more objectively and fairly, the demonstrated handover is accelerated and added to the comparison group. The original demonstrated trajectory is accelerated to the same speed as that of the policy obtained with strategy **S1**. As soon as the robot arm reaches the handover position, it triggers the grasping action of the receiver. Then, the giver releases the object after the receiver fully grasps it. The accelerated demonstration (Base) is treated as a baseline in the following comparison.

As depicted in Fig. 4, compared to the human demonstrated handover, not only the skill obtained through accelerating the demonstrated trajectory, but also the skills learnt with both proposed strategies have achieved a significant improvement. However, only the policies realized with our proposed framework can handover the object in a dynamic manner. In the following paragraphs, the concrete results will be detailed.

Table 2. Experiment results

	$t_{total}(s)$	$t_r(s)$	$t_g(s)$	$cost_{total}$	$cost_{wrench}$
Demonstration	9.604	6.834	7.639	0.65	0.2
Base	3.876	2,288	3.076	0.3946	0.1759
S1	3.511	1.711	2.336	0.3988	0.1814
S2	2.957	0.957	1.088	0.3559	0.2007

Time As indicated in Table 2, the original demonstrated handover lasts almost 10 s; In contrast, the skill based on the strategy **S1** accelerates this value to 3.5 s; For the baseline, although the movement of the demonstrated trajectory is speeded up to the same level, the overall execution time is dragged down by the releasing motion. (For the robotic hand used in the experiment, it takes around 0.8 s to fully deploy the palm);

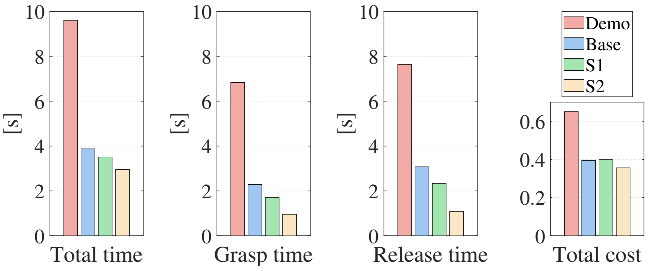

Fig. 4. Overview of the experiment results.

The fastest handover is achieved by the skill based on **S2**, which accomplishes the whole handover process within 3 s.

Moreover, as depicted in Fig. 5, the learnt skills couple the motion of the robot hands and arms effectively and accomplish the handover task faster. For the demonstrated handover, there are two main drawbacks. On the one hand, the motions of the robot arms and hands are not fully decoupled. The robot arms are nearly still when hand motions are triggered. On the other hand, there is an obvious time gap (0.805 s) between the motions of two hands, *i.e.,* the giver begins to release the object after the object is fully grasped; For the baseline, although a shorter completion time is obtained by performing arm movements at a faster rate, the flaws in the demonstration are not effectively eliminated. In contrast, the skill based on the strategy **S1** treats the hands and robot arms as a unity. The receiver begins to grasp the object during the movement of the robot arms, and the gap between the hand motions is narrowed to 0.625 s; For the skill learnt with **S2**, the system makes full use of the object's inertia. The grasping and releasing motions happen when robots operate at a relatively high speed and the grasping and releasing happen almost at the same time (with a 0.131 s gap). As a result, the transferred hammer is in a “flight” state, as illustrated in Fig. 6. During the object transfer process (subfigures a to f), there is an apparent rotation of the hammer, which dues to the combined effect of the hammer's gravity, inertia and the physical interaction between the robot hands.

External force Fig. 7 indicates the estimated external force at the wrist of each robot, and the stiffness configuration when grasping and releasing is marked out on the graph. From top to bottom, the figures represent the external force of the demonstrated handover, the compare baseline (accelerated demonstration), **S1** based policy and **S2** based policy, in turn. Compared to the original demonstration, the curves of the other three skills all show significant jitter at the beginning phase, which is caused by the higher acceleration. In addition, the shape and peak of the external force curves of the demonstration, baseline and skill **S1** are quite similar. In contrast, for the skill acquired with **S2**, the load transfer from the giver to the receiver exhibits distinct behavior in the force graph. Compared to the other handover skills, the hammer is in a ”flight” state in this process. As a result, the giver's maximum force only drops by around 11%, and the receiver's maximum force increases by 8% because of the dynamic impact of the transferred hammer. (The hammer weighs 0.3 kg).

In brief, both strategies based on the proposed framework have improved the handover performance considerably and achieved corresponding forward-looking results. They both enhanced the efficiency with a dynamic handover, under the premise of limiting interaction force. By constraining the hand motion timing, the policy trained with strategy **S1** obtained a solid grasping.

[3] The policy is converged when the range of the cost for roll-outs in an episode is smaller than 0.03.

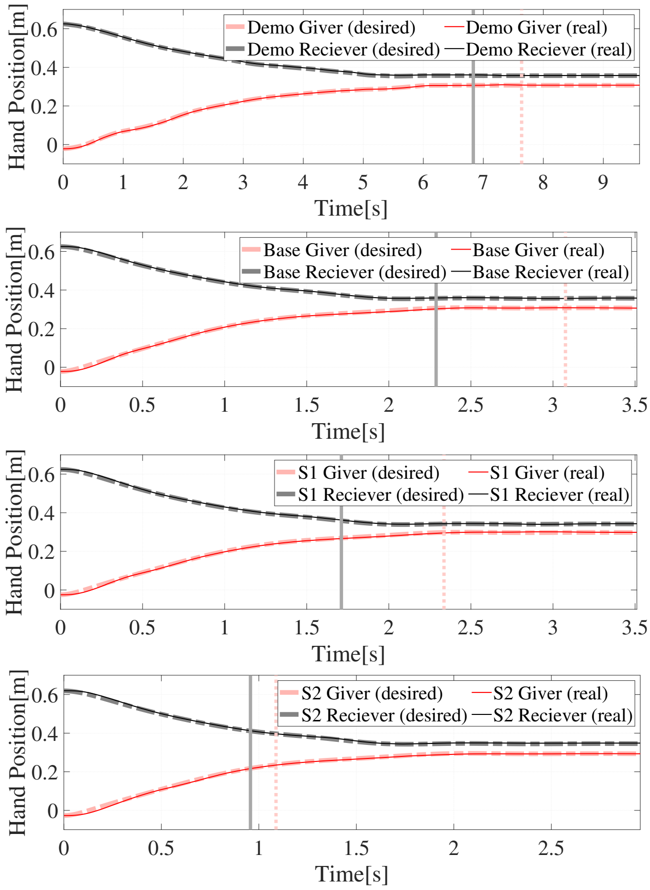

Fig. 5. Hand x-position in the handover process. Since the main motion of the robot arms occurs in the x-axis, only the hand's position in this direction is plotted. The grey vertical line refers to the time to grasp the object, and the pink dotted one indicates the giver's releasing time.

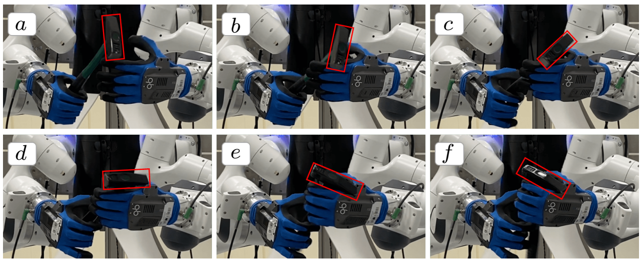

Fig. 6. Object transfer process of the policy learnt with strategy **S2**. The pose of the hammer head is marked with a red rectangle.

The pose of the handover object has a good certainty, and the external force during the handover is effectively lessened; While, for the unrestricted strategy **S2**, the robot made full use of the object's inertia and achieved a faster performance. As a side-effect, the receiver suffered from a slightly increased impact.

5. CONCLUSIONS

This paper proposed a novel framework to formulate the dynamic handover skill based on DMPs and variable impedance control, in which the movement trajectories, stiffness profiles, and timing of hand actions are optimized towards smooth and fast handover behaviour. The effectiveness of the proposed framework was validated on a dual-arm experiment setup in two scenarios, *i.e.,* one focuses on a safe and robust performance, and the other allows the robot to exploit hand motion timing fully. In both cases, the robot successfully learned dynamic handover skill which significantly speeded up the handover motion compared to the baseline (accelerated demonstration). The learnt skill demonstrated effective coupled timing of the hand actions and arm movements in dynamic motion. Furthermore, the latter case used for experimental evaluation showed that the object is possible to be transferred in a "flight" state without ensuring contact with the hands during handover phase. This implies that the object handover can be realized by using throwing-like motion, and worth more research efforts into extending the current framework to incorporate more highly-dynamic movements. In future works, the robot-to-robot handover skill will be trained using more object types to achieve better generalization ability and verified in human-to-robot and robot-to-human scenarios. We will also explore reinforcement learning algorithms to learn handover skills (Bing et al. (2023)).

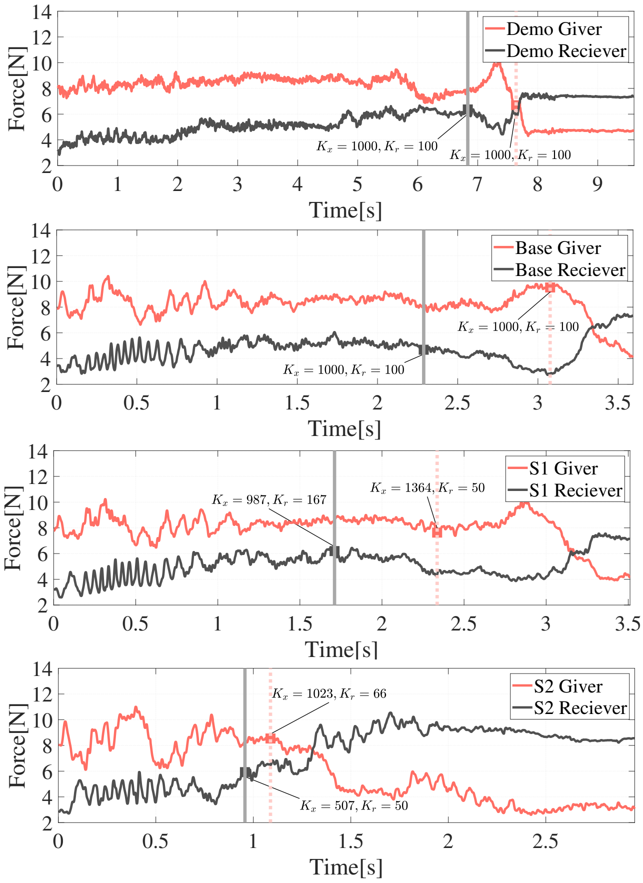

Fig. 7. External Force estimated at the wrist of the robot.

ACKNOWLEDGEMENTS

The authors acknowledge the financial support by the Bavarian State Ministry for Economic Affairs, Regional Development and Energy (StMWi) for the Lighthouse Initiative KI.FABRIK, (Phase 1: Infrastructure as well as the research and development program under, grant no. DIK0249). The work was also funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden. This work was also supported by the European Union's Horizon 2020 research and innovation programme as part of the project euROBIN under grant no. 101070596. Please note that S. Haddadin has a potential conflict of interest as a shareholder of Franka Emika GmbH.

REFERENCES

Albu-Schaffer, A., Ott, C., Frese, U., and Hirzinger, G. (2003). Cartesian impedance control of redundant robots: Recent results with the dlr-light-weight-arms. In *2003 IEEE International conference on robotics and automation (Cat. No. 03CH37422)*, volume 3, 3704–3709. IEEE.

Ardón, P., Cabrera, M.E., Pairet, E., Petrick, R.P., Ramamoorthy, S., Lohan, K.S., and Cakmak, M. (2021). Affordance-aware handovers with human arm mobility constraints. *IEEE Robotics and Automation Letters*, 6(2), 3136–3143.

Atkeson, C.G., Moore, A.W., and Schaal, S. (1997). Locally weighted learning. *Lazy learning*, 11–73.

Bing, Z., Zhou, H., Li, R., Su, X., Morin, F.O., Huang, K., and Knoll, A. (2023). Solving robotic manipulation with sparse reward reinforcement learning via graph-based diversity and proximity. *IEEE Transactions on Industrial Electronics*, 70(3), 2759–2769. doi:10.1109/TIE.2022.3172754.

Bonilla, M., Santina, C.D., Rocchi, A., Luberto, E., Santaera, G., Farnioli, E., Piazza, C., Bonomo, F., Brando, A., Raugi, A., et al. (2016). Advanced grasping with the pisa/iit softhand. In *Robotic Grasping and Manipulation Challenge*, 19–38. Springer.

Catalano, M.G., Grioli, G., Farnioli, E., Serio, A., Piazza, C., and Bicchi, A. (2014). Adaptive synergies for the design and control of the pisa/iit softhand. *The International Journal of Robotics Research*, 33(5), 768–782.

Chan, W.P., Parker, C.A.C., Van der Loos, H.F.M., and Croft, E.A. (2012). Grip forces and load forces in handovers: Implications for designing human-robot handover controllers. In *2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 9–16. doi:10.1145/2157689.2157692.

Costanzo, M., De Maria, G., and Natale, C. (2021). Handover Control for Human-Robot and Robot-Robot Collaboration. *Frontiers in Robotics and AI*, 8. doi:10.3389/frobt.2021.672995.

De Luca, A., Albu-Schaffer, A., Haddadin, S., and Hirzinger, G. (2006). Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1623–1630. IEEE.

Ficuciello, F., Villani, L., and Siciliano, B. (2015). Variable impedance control of redundant manipulators for intuitive human–robot physical interaction. *IEEE Transactions on Robotics*, 31(4), 850–863.

Haddadin, S., Parusel, S., Johannsmeier, L., Golz, S., Gabl, S., Walch, F., Sabaghian, M., Jähne, C., Hausperger, L., and Haddadin, S. (2022). The Franka Emika Robot: A Reference Platform for Robotics Research and Education. *IEEE Robotics & Automation Magazine*, 29(2), 46–64. doi:10.1109/MRA.2021.3138382.

Hansen, N. and Ostermeier, A. (2001). Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2), 159–195. doi:10.1162/106365601750190398.

Ijspeert, A., Nakanishi, J., and Schaal, S. (2002). Movement imitation with nonlinear dynamical systems in humanoid robots. *IEEE International Conference on Robotics and Automation*, (May), 1398–1403. doi:10.1109/ROBOT.2002.1014739.

Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., and Schaal, S. (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2), 328–73. doi:10.1162/NECO{_}a{_}00393.

Konstantinova, J., Krivic, S., Stilli, A., Piater, J., and Althoefer, K. (2017). Autonomous Object Handover Using Wrist Tactile Information. In Y. Gao, S. Fallah, Y. Jin, and C. Lekakou (eds.), *Towards Autonomous Robotic Systems*, volume 10454, 450–463. Springer International Publishing, Cham. doi:10.1007/978-3-319-64107-2_35.

Medina, J.R., Duvallet, F., Karnam, M., and Billard, A. (2016). A human-inspired controller for fluid human-robot handovers. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 324–331. doi:10/gj8kd3.

Ortenzi, V., Cosgun, A., Pardi, T., Chan, W.P., Croft, E., and Kulić, D. (2021). Object Handovers: A Review for Robotics. *IEEE Transactions on Robotics*, 37(6), 1855–1873. doi:10.1109/TRO.2021.3075365.

Schaal, S. (2006). Dynamic Movement Primitives – A Framework for Motor Control in Humans and Humanoid Robotics. *Adaptive Motion of Animals and Machines*, (1), 261–280. doi:10.1007/4-431-31381-8_23.

Stulp, F. and Raiola, G. (2019). Dmpbbo: A versatile python/c++ library for function approximation, dynamical movement primitives, and black-box optimization. *Journal of Open Source Software*. doi:10.21105/joss.01225.

Stulp, F. and Sigaud, O. (2012). Policy improvement methods: Between black-box optimization and episodic reinforcement learning.

Stulp, F. and Sigaud, O. (2013). Robot Skill Learning: From Reinforcement Learning to Evolution Strategies. *Paladyn, Journal of Behavioral Robotics*, 4(1). doi:10.2478/pjbr-2013-0003.

Tröbinger, M., Jähne, C., Qu, Z., Elsner, J., Reindl, A., Getz, S., Goll, T., Loinger, B., Loibl, T., Kugler, C., et al. (2021). Introducing garmi-a service robotics platform to support the elderly at home: Design philosophy, system overview and first results. *IEEE Robotics and Automation Letters*, 6(3), 5857–5864.

Ude, A., Nemec, B., Petrić, T., and Morimoto, J. (2014). Orientation in cartesian space dynamic movement primitives. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2997–3004. IEEE.