2022 31st IEEE International Conference on Robot and
Human Interactive Communication (RO-MAN)
August 29 - September 2, 2022. Naples, Italy.

2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN) | 978-1-7281-8859-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/RO-MAN53752.2022.9900856

Timing-Specified Controllers with Feedback for Human-Robot Handovers

Alap Kshirsagar†, Rahul Kumar Ravi†, Hadas Kress-Gazit and Guy Hoffman

***Abstract*— We develop and evaluate two human-robot handover controllers that allow end-users to specify timing parameters for the robot reach motion, and that provide feedback if the robot cannot satisfy those constraints. End-user tuning with feedback is a useful controller feature in settings where robots have to be re-programmed for varying task requirements but end-users do not have programming knowledge. The two controllers we propose are both receding-horizon controllers that differ in their objective function, and their user specified parameters, and subsequently their user-interface: One controller uses a minimum cumulative jerk (MCJ) objective function, and the other a minimum cumulative error (MCE) objective function. We implemented the controllers on a collaborative robot and conducted two controlled experiments to compare the user experience and performance of these controllers vis-à-vis a baseline proportional velocity (PV) controller. In each experiment, participants ($n = 30$) interactively tuned the controller parameters, and collaborated with a robot to perform a time-constrained repetitive task. We found that the timing controller with the MCE implementation can provide a better user experience, both while setting the parameters ($p = 0.011$) and performing the handovers with the robot ($p < 0.001$), and fewer failures ($p = 0.016$) compared to the PV controller, however the MCJ implementation did not provide better user experience compared to the PV controller. The MCJ controller also resulted in more failures than the PV controller. These results could inform the design of usable and effective end-user configurable controllers for human-robot interaction.**

***Index Terms*— End-User Programming, Human-Robot Handovers, Model Predictive Control**

I. INTRODUCTION

We present two controllers for human-robot handovers, in which end-users can specify the robot's behavior in terms of reach-to-handover timing parameters, along with two studies evaluating the controllers' performance. Prior studies of human-robot handovers did not consider scenarios where participants need to design the robot's handover controllers for a collaborative task with timing constraints. Such a flexibility is important in industrial settings with dynamically changing task requirements, and it can increase the acceptance of robots as coworkers. The majority of existing handover controllers require tuning non-intuitive controller parameters to specify the robot's behavior, for example, proportional gain [1]–[3] or Dynamic Movement Primitive (DMP) parameters [4], [5]. Some researchers have proposed handover controllers with intuitive parameters, for example timings of different phases of handovers [6] and delay/reach timings [7], but these controllers were not evaluated experimentally in scenarios where end-users can specify these parameters.

† A. Kshirsagar and R. Ravi contributed equally to this work.

All authors are with Sibley School of Mechanical and Aerospace Engineering, Cornell University, USA {ak2458, rr724, hadaskg, hoffman}@cornell.edu

Our proposed controllers not only allow users to specify intuitive timing constraints for the object handover, but also provide feedback to the end-user if they cannot satisfy those constraints, which can be used to better tune the controller. We use Receding Horizon Control (RHC), also known as Model Predictive Control (MPC), to generate the robot's closed-loop motion towards the human's hand from the user specified timing constraints. We implement the controllers on a collaborative robot with two different motion strategies, encoded as objective functions and timing constraints: minimum cumulative jerk (MCJ) and minimum cumulative error (MCE). In the MCJ strategy, the users can specify the maximum reaching time of the robot and the controller generates a minimum jerk trajectory towards the human hand within the specified time limit. In the MCE strategy, the users can specify a stall time along with the maximum reaching time of the robot and the controller generates a trajectory with a minimum cumulative error from the human hand.

We conducted two studies to evaluate the user experience and task performance with each motion strategy. In our studies, users interact with a physical robot to handle time-sensitive "vaccine vials," which have a minimum and a maximum air exposure time. The vials need to be packaged at a variety of locations, making the robot's handover strategy sensitive to the object and the final location. Participants design the robot's handover behavior by specifying the controller parameters for each controller and then have to use the controllers they tuned to perform the collaborative vaccine packaging task.

Each of the two controllers is compared to a baseline proportional velocity (PV) controller, a commonly used closed-loop controller for generating reach-to-handover motions [1], [2]. In the PV controller, the robot's instantaneous cartesian velocity is proportional to the distance from the human's hand, and the users can tune the proportional gain.

We find that the timing controller with the MCE strategy can provide better user experience and fewer failures compared to the PV controller, but we did not find any evidence to support that the timing controller with the MCJ strategy performs better than the baseline PV controller.

Contributions

To the best of our knowledge, this is the first study in which users design a collaborative robot's behavior for an industrial task with timing constraints. Our proposed controllers, unlike the existing robot controllers for object handovers, use intuitive parameters and provide feedback to the user. We find that two different instances of our proposed controller have contrasting effects on the user experience

Authorized licensed use limited to: Cornell University Library. Downloaded on December 27,2022 at 19:38:37 UTC from IEEE Xplore. Restrictions apply.

and task performance, even though they both have intuitive timing parameters and provide failure feedback. Our findings could shed light towards the design of better user-defined controllers for human-robot handovers.

II. Related Work

We situate our work in the context of prior studies of human-robot handovers, and the existing controllers for the reach phase of handovers.

A. Studies of Human-Robot Handovers

HRI researchers have studied human-robot handovers to understand human preferences for robot behaviors in different phases of a handover. This includes the approach [8], reach [5], [7], [9]–[12] and transfer [13], [14] phases of handovers. Some of these studies allowed users to interactively design the robot's handover behavior. Koay et al. [8] studied human's preferences about how the robot should approach, stop and hand over an object, by actively involving the participants in the creation of the robot's handover motion. Rasch et al. [15] asked participants to interactively guide a humanoid robot into different handover configurations and label the configurations as proper or improper. Cakmak et al. [10] conducted a user study to learn human preferences towards handover configurations in robot-to-human handovers by asking participants to configure the robot in what they think is a good or a bad configuration. In Porfirio et al.'s study [16], the users are asked to design a robot behavior from six micro-interactions for a package delivery task while obeying certain social norms. However, none of the prior works studied user-defined handover controllers for an industrial task with timing constraints, which is the focus of our present work.

B. Controllers for Robot's Reach-to-Handover Motion

Existing robot controllers for the reach phase of handovers can be broadly categorized into two groups: open-loop and closed-loop. Open-loop controllers compute the robot's reach-to-handover motion trajectory before the start of the handover and do not update it during the reach phase [9], [10], [12]. These controllers are not suitable for reactive motion planning, and require the human to adapt to the robot's motion. Closed-loop controllers constantly update the robot's reach-to-handover motion trajectory taking into account the observed behavior of the human [1]–[5], [9]. The majority of these closed-loop controllers require tuning non-intuitive controller parameters to generate the robot's behavior. For example, proportional gain in [1]–[3], DMP weights in [4], [5], trapezoidal velocity profile parameters in [9]. Some recent works have proposed handover controllers with intuitive parameters [6], [7] but these controllers do not provide feedback to the end-users to help them tune the controller parameters. We seek to address this gap by investigating closed-loop reach-to-handover controllers which use intuitive timing parameters and provide failure feedback to the user.

III. Handover Controller with Timing Parameters

We hope to understand how to design handover controllers that can be tuned intuitively by end-users. We seek to investigate how user-specified timing constraints should translate into controller parameters in a way that results in a high user experience and an efficient and useful controller. Our controllers in this study focus on the "reach" phase of the robot-to-human handover.

We propose two MPC [17] closed-loop controllers with reach time parameters for generating reach-to-handover motions of a robot. The MPC framework computes control inputs $\mathbf{u}_{t:t+L-1}$ for a given time horizon of length L, so that these controls optimize an objective function J under a set of N constraints $\{h_1, \ldots, h_N\}$. The objective function and constraints are defined over the system state $\mathbf{x}_t$ at time step t and the control input. Formally, MPC solves

$$\min_{\mathbf{u}_{t:t+L-1}} J_{t:t+L-1}(\mathbf{x}_t, \mathbf{u}_{t:t+L-1}), \tag{1a}$$

$$\text{s.t. } h_n(\mathbf{x}_i, \mathbf{u}_i) \leq 0,\ \forall i \in [t, t+L-1],\ \forall n \in [1, N] \tag{1b}$$

We formulate two instances of the MPC optimization problem for a robot's reach-to-handover motion generation. The two instances differ in terms of the objective functions and timing parameters that are specified by the user. We convert these timing parameters into MPC constraints, and the constrained optimization problem is solved at each time step in a receding horizon mode. The control input is given as the robot's end-effector velocity. If there is no feasible solution to the optimization problem at any time step, feedback is provided to the user that the robot is unable to reach the handover location within the specified constraints.

A. Minimum Cumulative Jerk (MCJ) Controller

In our first MPC implementation, users can specify the maximum reaching time of the robot and the controller generates a minimum jerk trajectory towards the human hand. Minimum Jerk is one of the most prominent models of point-to-point human arm movements [18]–[20]. Given a maximum reaching time, this controller should generate a smooth trajectory toward the human's hand. To maintain smoothness (*i.e.,* minimize jerk), this strategy tends to generate a slow robot motion that still stays within the timing constraint. Formally, the MPC objective function and constraints for this strategy are given by:

$$\min \sum_{i=0}^{L-1} \ddot{\mathbf{u}}_{t+i}, \tag{2}$$

s.t.

$$\mathbf{x}_{t+i+1} = \mathbf{x}_{t+i} + \mathbf{u}_{t+i}\Delta t\ \forall i \in [0, L-1], \tag{3a}$$
(System Dynamics)

$$||\mathbf{u}_{t+i}|| \leq u_{max}\ \forall i \in [0, L-1], \tag{3b}$$
(Velocity Limit)

$$\mathbf{x}_0 = [p_x, p_y, p_z], \tag{3c}$$
(Initial End Effector Position)

Authorized licensed use limited to: Cornell University Library. Downloaded on December 27,2022 at 19:38:37 UTC from IEEE Xplore. Restrictions apply.

$$\mathbf{u}_0 = [0,0,0]^T\,, \quad \text{(Zero Initial Velocity)} \tag{3d}$$

$$||\mathbf{x}_{i_{reach}-1} - \mathbf{h}_{i_{reach}-1}|| \leq \epsilon_x, \quad \text{(Get Near Human Hand at Reach Time)} \tag{3e}$$

$$||\mathbf{u}_{i_{reach}-1}|| \leq \epsilon_u. \quad \text{(Reach Velocity near Zero)} \tag{3f}$$

In the above formulation, the current time step is t, the prediction horizon is L, Δt is the time step size, $\mathbf{u}$ is the control input i.e. robot end-effector's velocity, $\mathbf{u_{max}}$ is the maximum permissible speed of the robot's end-effector, $\mathbf{x}$ is the position of the robot's end-effector, $[p_x, p_y, p_z]$ is the initial Cartesian position of the robot's end-effector, i_{reach} is the time step corresponding to the user specified reach time t_r, $\mathbf{h}$ is the position of the user's hand, ϵ_x and ϵ_u are the position and velocity tolerances.

B. Minimum Cumulative Error (MCE) Controller

Our second MPC implementation uses a minimum cumulative error (MCE) optimization strategy, meaning that the robot will try to minimize the total distance from the human hand over the whole trajectory. In contrast to the smooth but slower trajectories produced by the MCJ controller, we expect the MCE controller to generate fast robot trajectories, as these will have the lowest cumulative error. To allow users to also tune the lower bound of the reach time, users can specify a stall time along with the maximum reaching time of the robot and the controller generates a MCE trajectory after stalling for the specified amount of time.

Formally, the MPC objective function for this strategy is represented by,

$$\min \sum_{i=0}^{L-1} ||\mathbf{x}_{t+i} - \mathbf{h}_{t+i}||^2. \tag{4}$$

The constraints are identical to the MCJ formulation, with one exception: Eq. 3d is replaced by,

$$\mathbf{u}_i = [0,0,0]^T \;\; \forall i \in [0, i_{stall}-1]\,, \tag{5}$$

which means that the robot end-effector's velocity is 0 for the user specified time steps i_{stall}.

C. Baseline Controller

To evaluate the user experience and task performance of the two timing controllers, we compare them with a proportional velocity (PV) baseline controller. PV is a widely used closed-loop controller for generating reach-to-handover robot motions [1], [2]. This controller commands the robot's instantaneous Cartesian velocity $\mathbf{u}_t$ to be proportional to the distance between the robot end-effector's position $\mathbf{x}_t$ and the position of the human's hand $\mathbf{h}_t$ *i.e.*,

$$\mathbf{u}_t = k_p||\mathbf{x}_t - \mathbf{h}_t||, \tag{6}$$

where k_p is the user specified proportional gain. The control input is limited by the maximum permissible velocity of the end-effector, *i.e.*, $||\mathbf{u}_t|| \leq u_{max}$.

IV. Study Design

We conducted two studies to compare the user experience and task performance of our proposed timing controllers with the baseline PV controller in an industrial task simulated in a lab setting. The first study evaluated the MCJ controller and the PV controller, and the second study evaluated the MCE controller and the PV controller. In each study, we performed this comparison in two scenarios: first, a "non-optimization" scenario where users need to tune the controller parameters to successfully perform the task, but not necessarily in the shortest possible time, and second, an "optimization" scenario where they need to optimize the controller parameters to reduce the task duration and earn additional money. Our study design was motivated by these research questions: Whether and to what extent do the proposed controllers offer a **better user experience** than a baseline controller? Whether and to what extent do the proposed controllers **improve efficiency** of handovers as compared to a baseline controller?

A. Hypotheses

Since the timing controllers allow users to specify intuitive parameters and provide feedback to the user, we hypothesize that **H1:** *A timing controller will provide a better user experience than a PV controller*. We formulate similar hypotheses for three components of user experience: efficiency, perspicuity and dependability. **H1a:** *A timing controller will have higher efficiency than a PV controller*. **H1b:** *A timing controller will have higher perspicuity than a PV controller*. **H1c:** *A timing controller will have higher dependability than a PV controller*. Since the timing controllers allow users to specify time constraints on the robot's motion, we hypothesize that **H2:** *A timing controller will have lower failures in a time-constrained task than a PV controller*. We do not expect the timing controller to generate slower trajectories than a PV controller. Thus, we hypothesize that **H3:** *A timing controller will not have higher task duration in a time constrained task than a PV controller*. **H4:** *A timing controller will not have higher human idle time in a time constrained task than a PV controller*. **H5:** *A timing controller will not have higher robot idle time in a time constrained task than a PV controller*. We preregistered on a public experimental registry[1]. The experiment protocol was approved by Cornell University's Institutional Review Board for Human Participants.

B. Collaborative Task with Time Constraints

We tested the different robot controllers in a human-robot collaboration task consisting of object handovers from the robot to the human. The task simulates a scenario of vaccine packaging in the pharmaceutical industry: A person and a robot work together to transfer vaccine vials from a storage location to different packages, while ensuring that vaccines do not get over-exposure or under-exposure to air, *i.e.*, the handover time of the vial from the robot to the human falls within acceptable bounds. The experiment setup, as shown

[1] https://aspredicted.org/blind.php?x=5uk95q

Authorized licensed use limited to: Cornell University Library. Downloaded on December 27,2022 at 19:38:37 UTC from IEEE Xplore. Restrictions apply.

in Fig. 2, consists of three types of vaccines, G (green), Y (yellow), and R (red), and three packaging stations at different distances from the storage rack. The robot arm picks up the vials sequentially from the storage rack and hands them over to the person who has to place the vials in the packaging stations. One limitation of the MCE, MCJ and PV controllers is that they do not consider the 3D orientation information, and the robot always has a fixed orientation in our studies. The person is not allowed to enter the refrigeration zone around the storage stack, and if their hand is in this region the system generates a loud beeping sound. The robot starts the handover only if the person's hand is in the reachable work-space of the robot. Each vaccine has a different range of permissible exposure constraints, shown in Fig. 1, therefore requiring different time spans for the handover. For example, the green vaccine needs to be handed over within 1 to 4 seconds.

Vaccine	Minimum Exposure Time (s)	Maximum Exposure Time (s)
G	1	4
Y	2	5
R	3	6

Fig. 1: Graphical Interface showing the time constraints for each vaccine

In each study session, there were two types of rounds for each controller: design and test. Users tuned the controller parameters in the "design" round and performed the vaccine packaging task in the "test" round. In the design round, users could tune the controller parameters by performing handovers with different vaccine vials at the different packaging stations and observing the handover time. Once the design round was over, the controller parameters were locked-in for the subsequent test round. In the test round, users needed to place a set of vaccine vials, handed over by the robot, in three different packaging stations following a predefined sequence. In each study session, there were two types of test rounds: non-optimization and optimization. In the design round before the non-optimization test round, users tuned the controller parameters so that the vials are handed over within the timing constraints, but not necessarily in the shortest possible time. In the design round before the optimization test round, they re-tuned those controller parameters to perform the vaccine packaging task in the shortest time.

C. Monetary Incentive Scheme

For more external validity, we designed a monetary reward system to incentivize participants in the optimization test round: If they performed the task faster than a specified time limit, participants would get \$1 per second less than the specified time limit. For each second spent by the user in the unsafe refrigeration zone, they were penalized by \$1. Each failed handover incurred a penalty of \$2.

D. Robot and User Interface

We implemented the MCE, MCJ and PV controllers in the Python programming language, and programmed a Kinova Gen 3 robot using the Robot Operating System (ROS) to

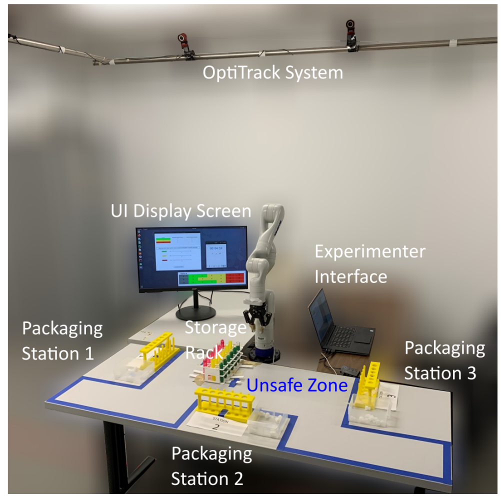

Fig. 2: The experiment setup consisted of a stack of vaccine vials, three packaging stations, a Kinova Gen3 robot arm, a screen to display UIs, and an OptiTrack system. The participants were penalized if they entered the area inside the blue tape *i.e.* the unsafe zone.

perform the vaccine packaging task autonomously with a human. We used an OptiTrack system to track the human hand and the robot's gripper. The robot released the vial if the human's hand was within a certain distance from the robot's gripper. We developed graphical interfaces in Qt for each controller. The timing constraints and the controller parameters of each vaccine are shown in Fig. 3 (PV controller), Fig. 4 (MCJ controller), and Fig. 5 (MCE controller). During the design rounds, the UI display screen showed the controller interface, current handover timer, design round's countdown timer, packaging sequence and time constraints for each vaccine. In the test rounds, only the controller interface, vaccine packaging sequence and time constraints for each vaccine were shown.

E. Human Motion Model

The MCE and MCJ controllers can utilize a prediction model of the human hand's motion in the reach phase of a handover. In our pilot trials, we observed that the users reached the handover location much faster than the robot, and waited for the robot at the handover location. Therefore in our studies, we used a static model of the human hand's position $\mathbf{h}$ over the prediction horizon L, given by,

$$\mathbf{h}_{t+i} = \mathbf{h}_t \forall i \in [0, L-1]. \tag{7}$$

The human hand's position is updated at each time-step using the observations from the OptiTrack system.

F. Experiment Procedure

Participants signed a consent form, read study instructions, and then performed an unrecorded practice session consisting of a design and a test round with a dummy controller to get familiarized with the experiment setup. The main part of the experiment consisted of two pairs of design and test rounds, one for each controller. The first test round was the non-optimization round and the second test round was the optimization round, as described in Section IV-B

Authorized licensed use limited to: Cornell University Library. Downloaded on December 27,2022 at 19:38:37 UTC from IEEE Xplore. Restrictions apply.

and Section IV-C. The participants did not know about the monetary incentive a-priori and an experimenter explained the reward system only after the participants finished the non-optimization rounds with each of the two controllers. The order of controllers was randomized and counterbalanced to avoid recency effects.

In each design round, the participants were given 5 minutes to tune the robot's controller parameters. Participants specified the controller parameters verbally to the experimenter, who activated the the UIs shown in Fig. 3 (PV), Fig. 4 (MCJ), and Fig. 5 (MCE). Participants would see the UI screen from their workspace position, as shown in Fig. 2. This choice was made to save participants the time to move back and forth between the workspace area and the keyboard and mouse area of the UI computer. Participants could test their specified parameters by verbally requesting the robot to hand over a vaccine vial, and updating the parameters after observing the handover time for that vaccine. They could do so as many times as they wanted within the 5 minute design round. The participants' verbal requests were relayed to the robot by the experimenter, to avoid speech processing errors.

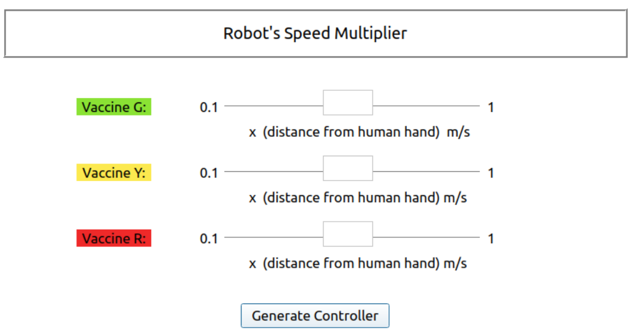

Fig. 3: Graphical Interface for the PV controller

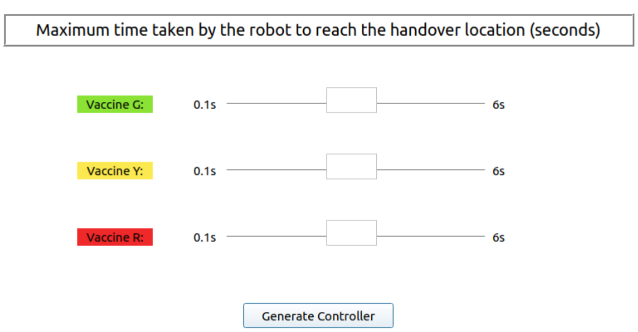

Fig. 4: Graphical Interface for the MCJ controller

Each design round was followed by a test round in which the participants had to perform 18 sequential handovers with the robot, which used the final controller parameters from the design round. The robot autonomously handed over the vials, one by one, in the following order: six green vaccines, six yellow vaccines and six red vaccines. The participants had to receive the vials and place them in a pre-defined

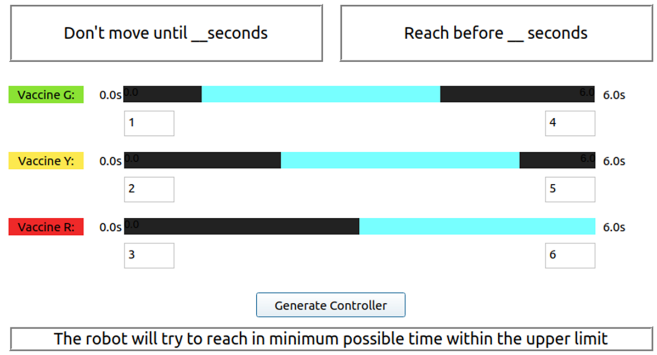

Fig. 5: Graphical Interface for the MCE controller

packaging sequence shown in Fig. 6. The participants were asked to move to the next packaging station before receiving the vaccine vial. This rule forced participants to perform handovers at different packaging stations, and the pre-defined packaging sequence ensured that the three vaccine types were handed over at each of the three stations.

Station #						
1	G	G	Y	Y	R	R
2	G	Y	Y	Y	R	R
3	G	G	G	Y	R	R

Fig. 6: Sequence of vaccines to be placed at each packaging station

After finishing each of the non-optimization and optimization test rounds with both controllers, participants reported their experience of working with each controller. We used Laugwitz et al.'s [21] user experience questionnaire to elicit the efficiency, perspicuity, and dependability ratings of each of the controllers. In Study 1, we also collected responses to these measures of workload: mental demand, performance, effort, and frustration, using the NASA TLX questionnaire, but in Study 2 we did not measure these variables (see: Section IV-G). Additionally, at the end of the experiment participants were asked to provide a written response to the prompt: "Please write a few sentences comparing the two controllers based on your experience of designing and using them in this experimental task".

We recorded the duration of the test rounds, the robot's and the human's idle time during the test rounds, and the number of handover failures in the test rounds. In the first study, failures include handovers with timing constraint violations, and—for the timing controller—also cases in which the controller could not find a feasible solution. In the second study, we also counted instances of vials falling from users' hand during the handover as failures (see: Section IV-G).

G. *Design Differences Between Study 1 and Study 2*

The experiment design of both of our studies was similar except for the following differences:.

- In Study 1, we asked participants to report their experience of "utilizing this controller for performing this task". We realized that this question combines two different user experiences: one of specifying the controller

Authorized licensed use limited to: Cornell University Library. Downloaded on December 27,2022 at 19:38:37 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Summary of Quantitative Results of Study 1 and Study 2

Variable	Study 1 : Non-optimization MCJ	Study 1 : Non-optimization PV	Study 1 : Optimization MCJ	Study 1 : Optimization PV	Type	Study 2 : Non-optimization MCE	Study 2 : Non-optimization PV	Study 2 : Optimization MCE	Study 2 : Optimization PV
Efficiency	4.72 (1.05)	4.64 (1.09)	4.73 (0.99)	5.08 (1.01)	P	5.38(1.21)	4.53(1.17)	5.06(1.31)	4.65(1.37)
					C	5.45(1.15)	4.52(1.14)	5.23(1.37)	4.70(1.44)
Perspicuity	4.79 (1.34)	4.82 (1.42)	4.39 (1.37)	4.83 (1.45)	P	5.45(1.37)	4.89(1.44)	5.06(1.97)	4.60(1.48)
					C	5.95(1.08)	5.11(1.23)	5.45(1.72)	4.49(1.54)
Dependability	4.92 (1.01)	4.97 (1.24)	4.58 (1.21)	5.22 (1.00)	P	5.56(1.25)	4.79(1.27)	5.08(1.58)	4.48(1.26)
					C	5.77(1.11)	4.73(1.04)	5.42(1.31)	4.43(1.42)
Overall UE	4.81 (0.98)	4.81 (1.16)	4.56 (1.02)	5.04 (1.01)	P	5.46(1.21)	4.74(1.19)	5.02(1.71)	4.57(1.31)
					C	5.72(1.01)	4.78(1.03)	5.42(1.30)	4.43(1.60)
Failures	2.77(2.67)	1.87(1.83)	2.43(2.57)	1.43(1.48)	-	1.93(3.93)	2.97(2.82)	1.63(3.22)	3.10(3.06)
Task Duration	286.07(15.00)	250.00(6.48)	281.08(9.96)	245.52(5.25)	-	268.25(28.25)	246.42(10.87)	259.63(12.08)	240.08(10.04)
HIT	198.29(21.38)	163.71(20.32)	200.85(22.13)	168.88(17.25)	-	178.28(34.96)	156.44(21.42)	180.80(19.51)	159.34(24.63)
RIT	165.21(12.58)	142.56(7.70)	162.67(10.77)	138.08(5.77)	-	166.40(34.36)	151.83(15.60)	159.81(17.52)	148.91(14.35)

P: Programming, C: Collaborating, HIT: Human Idle Time, RIT: Robot Idle Time

parameters and one of physically performing the handovers with the robot. To measure these two variables separately, in Study 2, we asked participants to report their experience of "programming the robot (setting the parameters) with this controller", and additionally, their experience of "physically collaborating with the robot (handling the vaccines) with this controller".

- In Study 1, we collected responses to the NASA TLX questionnaire to measure user workload. We found that these scores were correlated with the user experience (UE) scores.Therefore, in Study 2 we did not use the NASA TLX questionnaire.
- In Study 1, the feedback from the MCJ timing controller was "Couldn't reach your hand in time" and the experimenter verbally told participants that they need to increase the reach time. In Study 2, we changed the feedback from the MCE controller to a more comprehensible prompt: "I will not be able to reach in the specified time limit", and to maintain uniformity across participants, the experimenter did not provide any additional prompts.
- In Study 1, we noticed some instances of vials falling from the participants' hands during the handover, but we could not record these instances. In Study 2, we decided to record these instances since such falls are a mode of failure of the robot's handover controller.
- In the optimization test rounds of Study 1, the time limit for earning the monetary rewards described in Section IV-C was: 279.19 seconds (SD=3.96) for the MCJ controller and 245.16 seconds (SD=3.34) for the PV controller. In Study 2, this time limit was: 260 seconds (SD=0.0) for the MCE controller and 245 seconds (SD=0.0) for the PV controller, because the MCE controller resulted in faster handovers, on average, as compared to the MCJ controller.

H. *Participants*

31 participants (12 engineering, 19 non-engineering) participated in Study 1, and a different set of 30 participants (2 engineering, 28 non-engineering) participated in Study 2. We did not use the same participants in both the studies to avoid learning effects. In Study 1, there was an error in the data logging system for one participant, and we do not include this participant in the analysis of task duration and handover failures. Additionally, another participant did not fill out one of the user experience questionnaires, and we do not include this participant in the analysis of user experience. Thus we have $n = 30$ observations for all analyses. Each study session took approximately 90 minutes, and the participants were compensated a minimum of $15, and additional rewards based on performance in the optimization round. The average payout was $18.42 for Study 1 and $19.57 for Study 2.

V. Results

A. *Quantitative Results*

Table I shows the summary of quantitative results of both studies, where the mean and standard deviation of the dependent variables in the non-optimization and optimization rounds are shown. To test the hypotheses **H1–H5**, we conducted paired sample Student's t-tests on the measured differences. In case of significant deviations from normality, checked with the Shapiro-Wilk Normality test, we used the Wilcoxon's signed-rank test. The results of these tests are shown in Table II.

B. *Qualitative Feedback*

At the end of the study session, we had asked participants to provide qualitative feedback, with the prompt "Please write a few sentences comparing the two controllers based on your experience of designing and using them in this experimental task". Two coders independently analyzed the comments to find which controller each participant preferred in terms of efficiency, perspicuity, dependability and handover quality. The inter-coder reliability was 87.1% in Study 1 and 84.2% in Study 2.

The majority of the participants who wrote comments related to perspicuity preferred the perspicuity of the timing controllers over that of the baseline PV controller: 18 out of 28 in Study 1 and 17 out of 26 in Study 2.

Study 1, P22: *"The speed multiplier [PV] controller was a little more difficult to learn because I did not have a great idea of what the base speed was or what speed a number would produce until after practicing with the controller."*

Authorized licensed use limited to: Cornell University Library. Downloaded on December 27,2022 at 19:38:37 UTC from IEEE Xplore. Restrictions apply.

TABLE II: Results of Paired Sample Tests for Hypotheses **H1–H5** for the optimization and non-optimization rounds

				Non-optimization Round				Optimization Round			
Hypothesis	**Variable**	**C1**	**C2**	**Test**	**Statistic**	**p-Value**	**Effect Size**	**Test**	**Statistic**	**p-Value**	**Effect Size**
H1a	Efficiency (P)	MCE	PV	S	-2.658	0.006**	-0.485	S	-1.307	0.101	-0.239
H1a	Efficiency (C)	MCE	PV	S	-3.102	0.002**	-0.566	S	-1.859	0.037*	-0.339
H1a	Efficiency	MCJ	PV	S	-0.256	0.4	-0.047	S	1.696	0.950	0.310
H1b	Perspicuity (P)	MCE	PV	S	-1.556	0.065	-0.284	S	-1.002	0.162	-0.183
H1b	Perspicuity (C)	MCE	PV	S	-2.844	0.004**	-0.519	S	-3.245	0.001**	-0.592
H1b	Perspicuity	MCJ	PV	S	0.065	0.526	0.012	S	1.144	0.869	0.209
H1c	Dependability (P)	MCE	PV	S	-2.776	0.005*	-0.507	S	-1.635	0.056	-0.298
H1c	Dependability (C)	MCE	PV	S	-3.679	< 0.001***	-0.672	S	-3.452	< 0.001***	-0.630
H1c	Dependability	MCJ	PV	S	0.170	0.567	0.031	S	2.402	0.989	0.439
H1	Overall UE (P)	MCE	PV	S	-2.426	0.011*	-0.443	S	-1.197	0.120	-0.219
H1	Overall UE (C)	MCE	PV	S	-3.547	< 0.001***	-0.648	S	-3.307	0.001**	-0.604
H1	Overall UE	MCJ	PV	S	0.0	0.5	0.000	S	1.848	0.963	0.337
H2	Failures	MCE	PV	W	278.5	0.016*	0.474	S	2.068	0.024*	0.378
H2	Failures	MCJ	PV	S	-1.451	0.921	-0.265	S	-1.912	0.967	-0.349
H3	Task Duration	MCE	PV	W	13.0	1.000	-0.944	W	6.0	1.000	-0.974
H3	Task Duration	MCJ	PV	W	0.0	1.000	-1.000	S	-18.260	1.000	-3.334
H4	Human Idle Time	MCE	PV	W	38.0	1.000	-0.837	S	-6.291	1.000	-1.149
H4	Human Idle Time	MCJ	PV	W	2.0	1.000	-0.991	S	-15.455	1.000	-2.822
H5	Robot Idle Time	MCE	PV	W	46.0	1.000	-0.789	S	-3.053	0.998	-0.557
H5	Robot Idle Time	MCJ	PV	S	-11.187	1.000	-2.042	S	-12.236	1.000	-2.234

C1: Controller 1, C2: Controller 2, S: Student's t-test, W: Wilcoxon's signed-rank test, (P): Programming, (C): Collaborating
* $p < 0.05$, ** $p < 0.005$, *** $p < 0.001$

Study 2, P28: *"Personally, I like the time [MCE] controller compared to the second one [PV] because the multiplier [PV] is kind of confusing. Although there is pattern and relation between multiplier and speed of the robot reaching my hand, it takes me some time to think about what it means and how it is related to the robot's speed."*

Study 2, P04: *"[PV] was much easier to use since it was more straight forward and I only had to worry about one parameter instead of 2."*

The majority of the participants who wrote comments related to the dependability of the controllers felt that the PV controller was more dependable than the MCJ timing controller (10 out of 12 participants), but the MCE timing controller was more dependable than the PV controller (4 out of 5 participants).

Study 1, P26: *"With the time controller [MCJ], I faced a problem of the controller not being able to meet the set time which felt like it needed more tuning of parameters than it seems on paper."*

Study 2, P10: *"With the time [MCE], I could select a concrete amount of time and expect the robot arm not to move until that set amount of time had passed."*

The majority of the participants who wrote comments related to efficiency, preferred the baseline PV controller over the timing controllers: 4/4 in Study 1 and 5/7 in Study 2.

Study 1, P01: *"Since the three stations were at different distances from the robot, I felt that the multiplier [PV] worked better."*

Study 2, P18: *"But, the multiplier [PV] may be more efficient since you can change the speeds to deliver the vials faster."*

VI. Discussion

We developed handover controllers that allow users to specify a robot's behavior in terms of intuitive timing parameters and provides failure feedback. Our hypothesis was that—compared to a baseline PV controller—the timing controllers would result in a better user experience and lower failure rates, while not negatively affecting the handover timing efficiency of the robot.

Our results suggest that "the devil is in the details." Two instances of a timing-with-feedback controller can result in very different, and even opposite, user experience and task performance. The instances developed for this study differ in terms of the motion strategies of the robot, in turn encoded as objective functions and constraints of the MPC problem.

Our findings indicate that the MCJ strategy, which we initially thought would be preferred by users, results in a lower user experience compared to a PV controller, more prominently when the users have to optimize the controller parameters for faster handovers, whereas the MCE strategy results in a higher user experience compared to the baseline. One possible cause for this is that the robot's reach time, when using the MCJ controller, is more sensitive to target locations due to a gradual starting motion. This leads to a higher number of instances of infeasible solution of the MPC problem for MCJ (mean: 2.03, SD: 1.86) than MCE (mean: 1.20, SD: 2.11). Another possible cause is that the MCJ controller produces slower robot motions than the MCE and PV controllers. Finally, the prompts of the controller parameters and the feedback could have also affected the user experience of designing the controllers.

We found that the MCE controller resulted in lower vial falls from the human hand during the handover than the PV controller[2]. This could be attributed to the fact that the PV controller slows down the robot as it approaches the human hand, and sometimes stops short of the human hand due to a steady state error. Thus, users find it difficult to gauge exactly

[2]MCE mean falls: 0.07, PV mean falls: 0.4, $W = 105.5$, $p = 0.005$, $r = 0.758$

Authorized licensed use limited to: Cornell University Library. Downloaded on December 27,2022 at 19:38:37 UTC from IEEE Xplore. Restrictions apply.

when the object will be released by the gripper. In contrast, the MCE controller often overshoots the target position and thus thrust the object into the human hand, which may be preferred by users. In Study 2, P14 commented, for example: *"I like how it got really close to my hand and basically put the tubes into my hand."*

Both instances of our timing controller resulted in higher test-round task duration, robot idle time, and human idle time compared to the baseline controller. One possible cause is that users were more conservative in setting the parameters for the timing controller. For example, in case of the MCE controller even though the optimal stall time for the green vaccine was 0 seconds, 13 participants set a non-zero value in the optimization round. Second, there is an inherent sub-optimality with both instances of the timing controller for our setup shown in Fig. 2. A reach time that is optimal for handovers near the packaging station 1 might be infeasible for handovers near the packaging station 3 because of the longer travel distance from the storage rack. Thus, the user has to select a higher reach time to have successful handovers at all the packaging stations. Additionally the MPC controller has a higher sample time increasing it's response time by approximately 0.2 seconds per handover.

One surprising result was that the participants rated the MCE controller to be more efficient for physically collaborating with the robot, compared to the PV controller, even though the MCE controller resulted in higher task duration than the PV controller. Based on the qualitative feedback of some participants this could be attributed to the faster robot motion produced by the MCE controller. In Study 2, P30 commented, for example: *"I also found it [PV] the most annoying because it felt like it was moving so slowly. I actually like the second one [MCE] better because it feels like you are going faster."*

A few participants had more than 50% failures in the test rounds. In Study 1, one participant had 11 failures with the MCJ controller in the first test round. In Study 2, three participants had more than 10 failures with PV, and four participants had more than 10 failures with the MCE controller. In their qualitative feedback, some participants attributed these failures to their lack of experience with robots. Study 2, P9: *"I was a little slow at understanding how the concept applied to the programming. I am not familiar with robotics or programming them. Once I got the idea, it made logical sense. But it took me a while to get there."*

We hope that our timing-with-feedback controllers will be useful in scenarios where end-users need to design a robot's handover behavior. We investigated one such task in this paper, and our studies revealed the advantages and disadvantages of our proposed controllers vis-à-vis a baseline controller. Our work can motivate further research towards the design of better user-defined controllers for human-robot handovers. This includes investigating other motion strategies and constraints, designing corrective feedback, and testing in real industrial settings.

REFERENCES

[1] V. Micelli, K. Strabala, and S. Srinivasa, "Perception and control challenges for effective human-robot handoffs," in *Robotics: Science and Systems (RSS) Workshop on RGB-D Cameras*, 2011.

[2] M. Melchiorre, L. S. Scimmi, S. Mauro, and S. Pastorelli, "Influence of human limb motion speed in a collaborative hand-over task," in *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics*, 2018.

[3] J. R. Medina, F. Duvallet, M. Karnam, and A. Billard, "A human-inspired controller for fluid human-robot handovers," in *IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 324–331.

[4] M. Prada, A. Remazeilles, A. Koene, and S. Endo, "Dynamic movement primitives for human-robot interaction: comparison with human behavioral observation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1168–1175.

[5] A. Koene, A. Remazeilles, M. Prada, A. Garzo, M. Puerto, S. Endo, and A. M. Wing, "Relative importance of spatial and temporal precision for user satisfaction in human-robot object handover interactions," in *International Symposium on New Frontiers in Human-Robot Interaction*, 2014.

[6] A. Kshirsagar, H. Kress-Gazit, and G. Hoffman, "Specifying and synthesizing human-robot handovers," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 5930–5936.

[7] M. K. Pan, E. Knoop, M. Bächer, and G. Niemeyer, "Fast handovers with a robot character: Small sensorimotor delays improve perceived qualities," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[8] K. Koay, E. Sisbot, D. Syrdal, M. Walters, K. Dautenhahn, and R. Alami, "Exploratory study of a robot approaching a person in the context of handing over an object." in *AAAI Spring Symposium*, 2007.

[9] M. Huber, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer, "Human-robot interaction in handing-over tasks," in *IEEE International Symposium on Robot and Human Interactive Communication*, 2008.

[10] M. Cakmak, S. S. Srinivasa, M. K. Lee, J. Forlizzi, and S. Kiesler, "Human preferences for robot-human hand-over configurations," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 1986–1993.

[11] C.-M. Huang, M. Cakmak, and B. Mutlu, "Adaptive coordination strategies for human-robot handovers," in *Robotics: Science and Systems*, Rome, Italy, 2015.

[12] A. Kshirsagar, M. Lim, S. Christian, and G. Hoffman, "Robot gaze behaviors in human-to-robot handovers," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6552–6558, 2020.

[13] M. K. X. J. Pan, E. A. Croft, and G. Niemeyer, "Exploration of geometry and forces occurring within human-to-robot handovers," in *IEEE Haptics Symposium*, Mar. 2018.

[14] Z. Han and H. Yanco, "The effects of proactive release behaviors during human-robot handovers," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2019, pp. 440–448.

[15] R. Rasch, S. Wachsmuth, and M. König, "An evaluation of robot-to-human handover configurations for commercial robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 7588–7595.

[16] D. Porfirio, A. Sauppé, A. Albarghouthi, and B. Mutlu, "Authoring and verifying human-robot interactions," in *Annual ACM Symposium on User Interface Software and Technology*, Oct. 2018.

[17] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer science & business media, 2013.

[18] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[19] S. Shibata, K. Tanaka, and A. Shimizu, "Experimental analysis of handing over," in *IEEE International Workshop on Robot and Human Communication*, 1995, pp. 53–58.

[20] C. Wang, L. Peng, Z.-G. Hou, L. Luo, S. Chen, and W. Wang, "Experimental validation of minimum-jerk principle in physical human-robot interaction," in *International Conference on Neural Information Processing*, 2018, pp. 499–509.

[21] B. Laugwitz, T. Held, and M. Schrepp, "Construction and evaluation of a user experience questionnaire," in *Lecture Notes in Computer Science*, 2008, pp. 63–76.

Authorized licensed use limited to: Cornell University Library. Downloaded on December 27,2022 at 19:38:37 UTC from IEEE Xplore. Restrictions apply.