

Iterative Learning Control for Human-Robot Collaborative Output Tracking

Jonathan Realmuto, Rahul B. Warrierr, and Santosh Devasia
Dept. of Mech. Eng., U. of Washington, Seattle, WA, USA - 98195

Abstract—This article studies human-robot learning control for collaborative output-tracking tasks. We propose an algorithm to adaptively tune the frequency-dependent iteration gain and apply it to two cases: when the desired output is directly available to the robot and when the robot infers the desired output from human-achieved output. Experiment results are presented to illustrate the application of the proposed approach to a human-robot collaborative output-tracking task. Results show that the error converges to less than the closed-loop robot tracking error, and that the approach can provide varying levels of robot assistance by selecting the desired human-robot collaboration level.

I. INTRODUCTION

At the forefront of robotics is the idea of social robots that are capable of collaboratively working with people to perform tasks. For example, robots are being developed that provide the elderly or the disabled with assistance in performing day-to-day tasks [1]–[4]. A great many of these interactions require the robot to work jointly with the person, such as helping them get up from bed, helping them prepare meals together, or even in performing exercises for post-stroke physical rehabilitation [5]. In other applications, robots are being envisioned as partners in joint human-robot problem-solving teams for search and rescue operations, construction, and space exploration, e.g., [6]. The current article investigates iterative learning control (ILC) [7], [8] of the robot for collaboratively performing a task. The main contribution is the use of an adaptive iteration gain that is tuned during the iterations to avoid potential divergence of the input during the collaborative learning. The approach is applied to an example human-robot collaborative output-tracking task and experimental results show convergence of the robotic tracking task to less than the closed-loop robot tracking error.

Iterative learning control (ILC) [7], [8] is readily applicable to collaborative learning of demonstrated trajectories, and has been well studied as a tool for learning from demonstration, for example in automating surgical sub-tasks [9], and in a self-learning automobile cruise controller [10]. The current research evaluates two aspects of such collaborative learning: (i) varying levels of assistance from the robot to achieve a task and (ii) access to the desired output to the robot controller.

The first aspect of different levels of robot assistance of the human in collaboratively performing a task is important in rehabilitation applications with the use of active orthosis [11], [12] and in robotic prostheses [13]–[16]. The second aspect of access to the desired output to the robot controller is important since conventional applications of ILC require an error signal (that is dependent on the desired output) to iteratively correct the input. Other data-driven supervised learning techniques, such as Reinforcement Learning (RL) work around this lack-of-output access limitation by using an exploration-exploitation strategy to maximize a reward function, which is assumed to be known or is obtained by imitating expert user demonstrations [17]. Such techniques usually require extensive training sessions to obtain reasonable performance, which may not be feasible in collaborative learning applications. In contrast, the current work studies the problem of collaborative learning when the robot controller does not have access to the desired output but needs to provide assistance based on the actions of the human, who can adapt and learn to perform the task at hand. In both cases, with and without direct access to the output, the experimental results with the proposed approach show convergence to the desired level of robotic assistance.

II. PROBLEM FORMULATION AND PROPOSED METHOD

A. Problem setup

1) *The collaborative output*: The goal is to find the robot input u that adjusts the robot output q_r for the collaborative output q to track a desired output q_d , i.e.,

$$q = q_d \quad (1)$$

where collaboratively-controlled output q is given by

$$q = q_r + q_h \quad (2)$$

and q_h is the effect of the human control on the output q and is referred to as the human input. In the following, the output and input are assumed to have two components along the (x, y) coordinate system, i.e.,

$$q = \begin{bmatrix} q_x \\ q_y \end{bmatrix}, q_r = \begin{bmatrix} q_{rx} \\ q_{ry} \end{bmatrix}, q_h = \begin{bmatrix} q_{hx} \\ q_{hy} \end{bmatrix}, q_d = \begin{bmatrix} q_{dx} \\ q_{dy} \end{bmatrix}, u = \begin{bmatrix} u_x \\ u_y \end{bmatrix}. \quad (3)$$

2) *ILC of robot assistance*: Let the level of assistance by the robot to the human for output tracking be specified as the fraction β of the output tracking q_d achieved by the desired robot output q_{rd} , i.e.,

$$q_{rd} = \beta q_d. \quad (4)$$

{realmuto,warrierr,devasia}@uw.edu. J. R. and R.B.W contributed equally.

Then an iterative approach, in the frequency domain, to update the control law u to the robot is given by

$$\begin{aligned} u_{x,k+1}(\omega) &= u_{x,k}(\omega) + \rho_{x,k}(\omega) \hat{G}_x^{-1}(\omega) e_{x,k}(\omega) \\ u_{y,k+1}(\omega) &= u_{y,k}(\omega) + \rho_{y,k}(\omega) \hat{G}_y^{-1}(\omega) e_{y,k}(\omega) \end{aligned} \quad (5)$$

where \hat{G}_x, \hat{G}_y are models of the robot dynamics G_x, G_y , assumed to be decoupled along the x and y axes, the tracking error e_k is given by

$$\begin{aligned} e_{x,k}(\omega) &= \beta q_{dx}(\omega) - q_{rx,k}(\omega) \\ e_{y,k}(\omega) &= \beta q_{dy}(\omega) - q_{ry,k}(\omega) \end{aligned} \quad (6)$$

and the subscript $k > 0$ is an integer representing the iteration step.

3) *ILC convergence*: Convergence is achieved at each frequency, ω with the iterations in Eq. (5), e.g., in the x axis,

$$\lim_{k \rightarrow \infty} u_{x,k}(\omega) = G_x^{-1}(\omega) \beta q_{dx}(\omega), \quad (7)$$

provided the constant iteration gain, $\rho_{x,k}(\omega) = \rho_x(\omega)$ and the phase error, $\Delta_{\theta,x}(\omega)$ in the model $\hat{G}_x(\omega)$ are sufficiently small, i.e., from [18], [19],

$$|\Delta_{\theta,x}(\omega)| < \frac{\pi}{2} \quad (8)$$

$$0 < \rho_x(\omega) < \frac{2 \cos(\Delta_{\theta,x}(\omega))}{\Delta_{m,x}(\omega)} = \rho_x^*(\omega), \quad (9)$$

where $\Delta_x(\omega) = \Delta_{m,x}(\omega) e^{j\Delta_{\theta,x}(\omega)} = \frac{G_x(\omega)}{\hat{G}_x(\omega)}$.

The conditions are similar for the y axis.

4) *Convergence problem*: Since the modeling uncertainties are not known ahead of time, the problem investigated in the following is the selection of the frequency-dependent iteration gains ρ_x, ρ_y to satisfy the iteration conditions, e.g., in Eqs. (8) and (9) for the x axis.

B. Proposed method: ILC with adaptive iteration gain

The proposed adaptive ILC monitors the tracking error e_k at each iteration step k and reduces the iteration gain ρ_k if the tracking error e_k is seen to increase. The approach is described below for the x -axis. The approach for the y -axis is similar and not described due to space limitations.

If the error is decreasing at a specific frequency ω , at iteration step k , i.e.,

$$|e_{x,k}(\omega)| \leq |e_{x,k-1}(\omega)| \quad (10)$$

then the iteration update in Eq. (5) is followed at frequency ω with no change in the iteration gain, i.e.,

$$\rho_{x,k}(\omega) = \rho_{x,k-1}(\omega). \quad (11)$$

If the error increases at iteration step k for a specific frequency $\bar{\omega}$, i.e.,

$$|e_{x,k}(\bar{\omega})| > |e_{x,k-1}(\bar{\omega})| \quad (12)$$

then the iteration gain ρ at that frequency $\bar{\omega}$ is scaled down by $K_x > 1$ till the error e at that frequency $\bar{\omega}$ decreases below the value at iteration step $k-1$, i.e., for integers $i \geq 0$,

$$\rho_{x,k+i}(\bar{\omega}) = \begin{cases} \frac{\rho_{k-1}(\bar{\omega})}{K_x^{i+1}}, & \text{if } e_{x,k+i}(\bar{\omega}) > e_{k-1}(\bar{\omega}) \\ \rho_{k+i-1}(\bar{\omega}), & \text{otherwise.} \end{cases} \quad (13)$$

Moreover, to avoid potential divergence, the input u_x at the frequency $\bar{\omega}$ is updated from the iteration step $k-1$ till the error decreases, i.e., if $e_{x,k+i}(\bar{\omega}) > e_{x,k-1}(\bar{\omega})$, then

$$u_{x,k+i+1}(\bar{\omega}) = u_{x,k-1}(\bar{\omega}) + \rho_{x,k+i}(\bar{\omega}) \hat{G}_x^{-1}(\bar{\omega}) e_{x,k-1}(\bar{\omega}). \quad (14)$$

Lemma 1: If the maximum acceptable iteration gain $\rho_x^*(\omega)$ in Eq. (9) for convergence is non-zero, and the initial iteration gain $\rho_{x,0}(\omega)$ is bounded, then the iterative inputs $u_{x,k}$ converge as in Eq. (7). If the maximum acceptable iteration gain $\rho_x^*(\omega)$ is zero, then the tracking error $e_{x,k}(\omega)$ remains bounded.

Proof: The detailed proof is not provided for brevity. Briefly, when the initial iteration gain is greater than the maximum iteration gain and is non-zero at a frequency ω , i.e.,

$$\rho_{x,0}(\omega) \geq \rho_x^*(\omega) > 0, \quad (15)$$

the iteration gain will become sufficiently small after i^* iterations, since (for a sufficiently large integer i^*)

$$\frac{1}{K_x^{i^*}} \rho_{x,0}(\omega) < \rho_x^*(\omega) \quad (16)$$

The boundedness of the error follows from the design of the iterative input in Eq. (14), which is updated from iteration step $k-1$ before the error started to diverge. ■

Remark 1: After convergence to the noise level, the iteration gain can be reduced with the proposed approach due to potential increase in the tracking error at an iteration step due to noise effects (which should not be corrected).

C. ILC without access to desired output

If the desired output is not directly accessible to the robot controller, and if the human is able to learn the desired output then the error can be estimated using the achieved output q , i.e., the robot-tracking error in Eq. (6) can be replaced by the following error estimate \hat{e}

$$\begin{aligned} e_{x,k}(\omega) &\approx \hat{e}_{x,k}(\omega) = \beta q_{x,k}(\omega) - q_{rx,k}(\omega) \\ e_{y,k}(\omega) &\approx \hat{e}_{y,k}(\omega) = \beta q_{y,k}(\omega) - q_{ry,k}(\omega). \end{aligned} \quad (17)$$

In this case the robot control learning is coupled to the learning of the human in contrast to the case where the robot has access to the desired output. In this case the robot learning is decoupled from the human learning, although the human has to adapt to the changing robot input during the iteration process.

Remark 2: If sufficient time is permitted for the human input to converge and track the desired output (with sufficiently small error) after each iteration update of the robot input, then the estimated tracking error \hat{e} in the robot position is close to the actual error in Eq. (6). Convergence of such intermittent updates has been studied in [20].

III. SYSTEM DESCRIPTION

A. Overview of experimental setup

The collaborative iterative control is applied to a flexible-spring system where both the human and robot effort can be simultaneously used to achieve a desired output as shown in Fig. 1. A flexible structure consisting of a coil spring is

attached rigidly to the end-effector of a MICO2 robot arm, manufactured by Kinova Robotics. A laser attached to the end of the flexible structure projects a red dot on to the screen — the position of the red dot is the output q . An LCD projector projects the desired trajectory q_d , represented by the center of a green circle. A camera is used to track the real-time positions of the laser dot q and the green circle q_d using the image processing tools available in the OpenCV library [21]. The objective of the task is to track the center of the green circle q_d , whose trajectory is specified but not known a priori to the human subject or the machine controller, using the red laser dot q . The robot arm is not compliant and can only be actuated by the robot controller. Additionally, the human subject can deflect the spring to augment the tracking task.

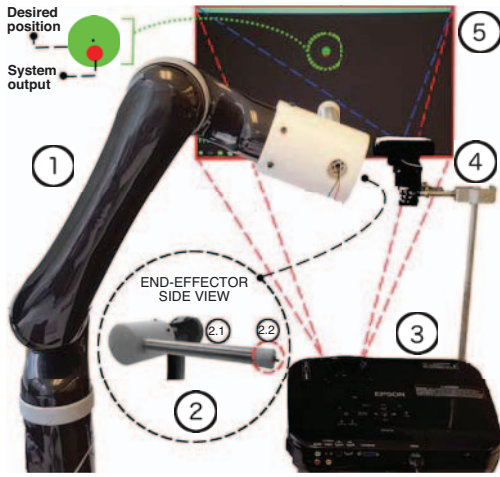


Fig. 1: Experimental setup for the human-robot output-tracking experiment: (1) Kinova Mico 4-dof robot arm; (2) side view of end-effector showing flexible spring structure (2.1) with the attached red laser pointer (2.2); (3) LCD projector to display desired output position (green circle); (4) Logitech C920 web-camera to track the output position q (red laser dot) and the desired position q_d (green circle); and (5) black screen where projector image is displayed.

B. Controlled system: flexible structure

The flexible structure, as shown in Fig. 2, is manipulated by the human operator and the robot end-effector to achieve the final output, q , namely the position of the laser dot on the screen. The robot end-effector is attached to one end of the flexible structure, i.e., end (A) in Fig. 2, while the human operator deflects the structure relative to end (A) by applying a displacement, δ_h at the other end, i.e., end (B) in Fig. 2. Thus, relative to the end (A), we can consider the flexible structure as a cantilever beam with a concentrated load, P acting at end (B) due to the human operator. Then, the displacement of end (B) relative to end (A), i.e., δ_h is related to the slope, θ as, (using the Timoshenko beam theory [22]) $\theta \approx \frac{3\delta_h}{2l}$, where, l is the length of the flexible structure, and $\delta_h = (\delta_{h,x}, \delta_{h,y})$ are deflections of end (B) along each decoupled coordinate (x, y) . The position q of the laser dot on the screen is then obtained as,

$$q = q_r + q_h, \quad \text{where, } q_h = f(\delta_h) \quad (18)$$

for each decoupled coordinate, $q = (q_x, q_y)$, and when the distance to the screen, $L \gg l$ the length of the flexible structure, the relationship between the deflection, δ_h and the screen position, q_h is given by,

$$q_h = f(\delta_h) \approx L \tan \theta = L \tan \left(\frac{3\delta_h}{2l} \right). \quad (19)$$

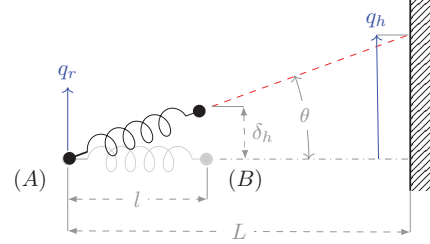


Fig. 2: Collaborative control of the flexible structure. (a) Robot end-effector controlled position, $q_r = (x_r, y_r)$ of end (A), (b) Human operator's deflection with respect to end (A), $\delta_h = (\delta_{x,h}, \delta_{y,h})$, (c) Total position, $q = q_r + q_h$, where $q_h = f(\delta_h)$, as in Eq. (19).

C. Robot position control

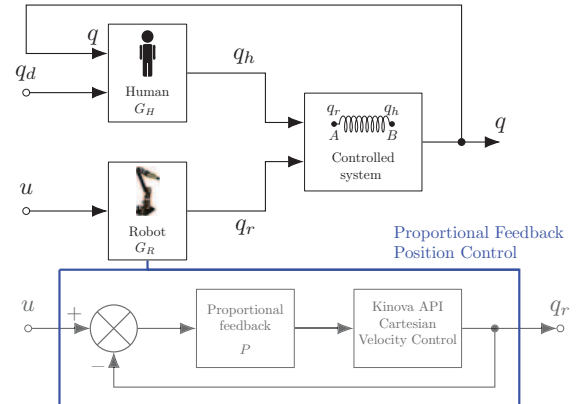


Fig. 3: Block diagram of control architecture of the collaborative output tracking task. Here q_d is the desired output of the collaboratively controlled system, whose output is q ; q_h is the human input to the controlled system; q_r is the robot input to the system; u is the position command to the robot from the machine controller;

The end-effector of the robot arm is directly controlled in Cartesian space using the control scheme depicted in Fig. 3. The Kinova MICO2 is provided with an Application Programming Interface (API), written in C++, that provides basic functions to control the robot arm. The API user manual recommends the use of velocity control in Cartesian end-effector space to achieve smooth trajectories. Moreover, in this mode of operation, the linear speed of the robot is limited to 0.2 m/s, which aids a human operator to safely interact with the robot. An outer proportional feedback loop as shown in Fig. 3 was designed around the commanded position, u to operate the robot with the Kinova velocity controller.

Towards designing the outer position-control loop, the available API velocity-control method was characterized using a step input, u along each axis (x, y) as in Eq. (3). Fig. 4 shows that the step response closely fits a fourth-order linear model, decoupled in each coordinate (x, y), of the form,

$$\frac{d}{dt} \begin{bmatrix} q_x \\ \dot{q}_x \\ q_y \\ \dot{q}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\tau & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\tau \end{bmatrix} \begin{bmatrix} q_x \\ \dot{q}_x \\ q_y \\ \dot{q}_y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \tau & 0 \\ 0 & 0 \\ 0 & \tau \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix},$$

$$y = \begin{bmatrix} q_x \\ q_y \end{bmatrix} = q_r, \quad (20)$$

where, the input $u = v_c$ is the commanded velocity, and the output $y = q_r$ is the position of the end-effector, both in the Cartesian end-effector space. The value of $\tau \approx 8.0$

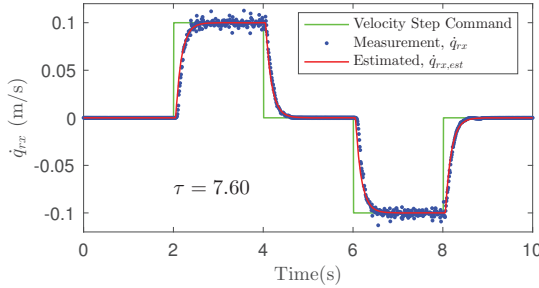


Fig. 4: Step response of the built-in closed loop velocity control method from the Kinova API, compared to the estimate from the model in Eq. (20) in the x -direction.

was found to best fit the step response data to the model in (20), as depicted in Fig. 4 for the x -coordinate direction. The proportional controller was selected as $P = K_p = 2.0$ for a fast rise time with overshoot less than 2%. The resulting closed-loop transfer function, $\hat{G} = [\hat{G}_x, \hat{G}_y]$ is obtained as,

$$\hat{G}_x(s) = \frac{q_{rx}(s)}{u_x(s)} = \frac{K_p \cdot \frac{\tau}{s(s+\tau)}}{1 + K_p \cdot \frac{\tau}{s(s+\tau)}}, \quad (21)$$

$$\hat{G}_y(s) = \frac{q_{ry}(s)}{u_y(s)} = \frac{K_p \cdot \frac{\tau}{s(s+\tau)}}{1 + K_p \cdot \frac{\tau}{s(s+\tau)}}$$

for $s = j\omega$ and $j = \sqrt{-1}$. Also, for low frequencies, $s \rightarrow 0$,

$$\hat{G}_x(s) \Big|_{s \rightarrow 0} = \hat{G}_y(s) \Big|_{s \rightarrow 0} = 1. \quad (22)$$

Thus, in the following, the system models, $\hat{G}_x(\omega), \hat{G}_y(\omega)$ in Eq. (5) are chosen as, $\hat{G}_x(\omega) = \hat{G}_y(\omega) = 1$, for all frequencies, ω .

IV. RESULTS AND DISCUSSION

A. Experimental procedure

Two experimental conditions will be tested: iterative update of the robot input with access to the desired output (access condition) and with no access to the desired output (no access condition). For each condition, four levels of assistance will be used: $\beta = 0.25, 0.5, 0.75, 1$. The learning process evolves over 11 iterations. On the first iterations the robot command is zero e.g., $q_r = 0$ for the extent of the iteration. After each iteration the ILC algorithm is executed and a new open

loop command is used on the next iteration. For all conditions $K_x = K_y = 2$ in Eq. (13).

B. Effect of adaptive learning weights

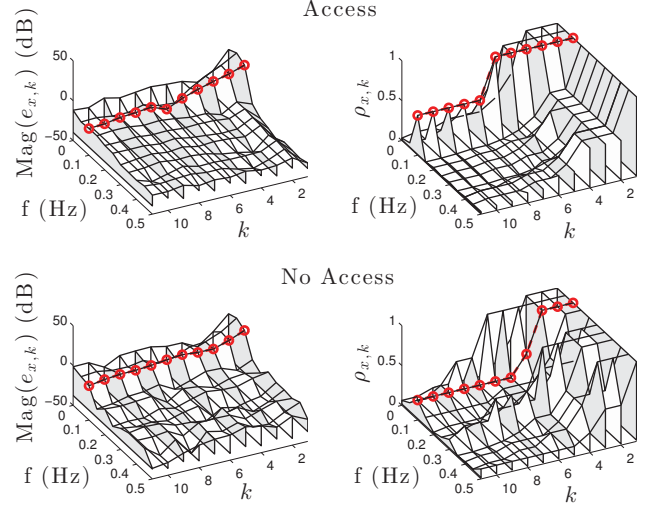


Fig. 5: The evolution of the proposed ILC algorithm, in the x axis, for the access case (top two plots) and no access case (bottom two plots) with level of assistance fraction $\beta = 1$. The first column of plots are the magnitude of error in the x direction $e_{x,k}$ reported in dB. The second column of plots are the learning weights in the x direction $\rho_{x,k}$. The red dots follow the evolution of learning in one frequency, $f = 0.1$ (Hz), across all iterations, $k = [1, 11]$. It can be seen, by following the red dots, that the learning weights decrease when the errors increase.

The proposed approach utilizes adaptive learning weights $\rho(\omega)$ in the frequency domain (see Eq. (13)). Figure 5 illustrates how each weighting component $\rho_{x,k}(\omega)$ were adapted during the learning process for the no access and access condition with $\beta = 1$ (similar results occurred in the y axis). The learning weights were initialized to $\rho_{x,0}(\omega) = \rho_{y,0}(\omega) = 1$ for frequency less than or equal to $f = 0.50$ (Hz) and $\rho_{x,0}(\omega) = \rho_{y,0}(\omega) = 0$ for higher frequencies. A frequency of $f = 0.5$ (Hz) is approaching the bandwidth limit of human visual tracking [23], hence the desired trajectory was limited to lower frequencies. Referring to Fig. 5, it can be seen that as the components of the error $e_{x,k}(\omega)$ increases, the corresponding weighting component $\rho_{x,k}(\omega)$ decreases. It then follows that the proposed approach exhibits some stability properties, e.g., the error growth must be bounded since the learning weights always decrease with increasing error.

Example trajectories achieved by the human-robot collaboration, for all assistance levels β , as well as both access and no access conditions are shown in Fig. 6. Note that the dashed lines represented the desired trajectory of the robot q_{rd} . The plots were generated using data from the last iterations, $k = 11$, for each condition.

C. Convergence analysis

The proposed ILC algorithm resulted in robot performance below what can be achieved with the closed-loop robot con-

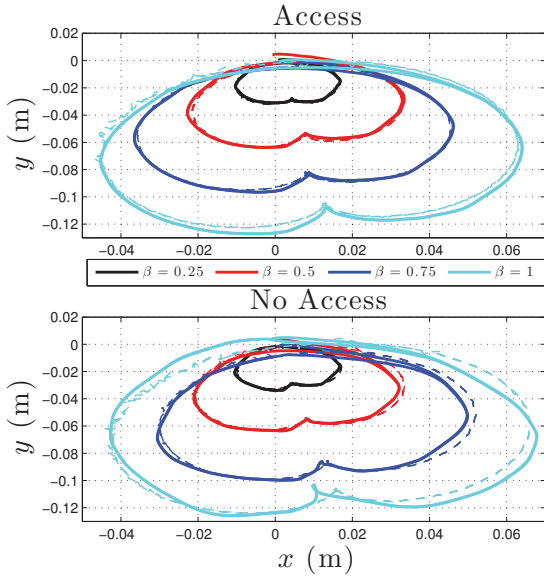


Fig. 6: Tracking achieved by the human-robot collaboration for both access and no access case at all assistance levels. Desired robot output q_{rd} (dashed lines) and the robot output q_r (solid lines) from the last iterations, $k = 11$.

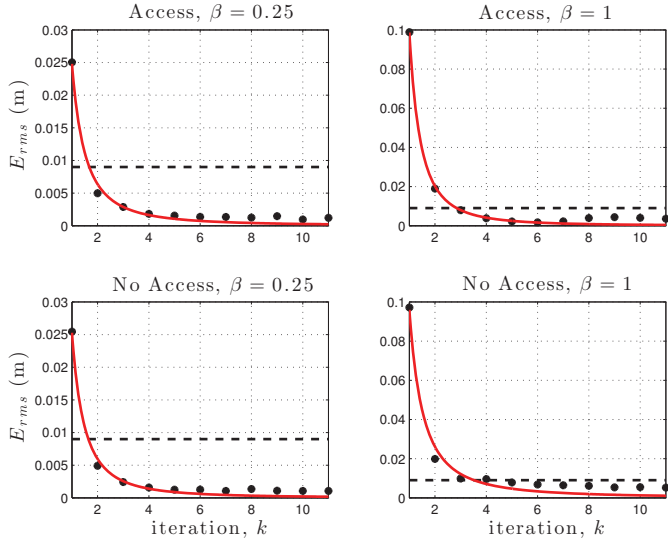


Fig. 7: E_{RMS} in Eq. (24) versus iteration for no access and access case with $\beta = 0.25$ and $\beta = 1$. Power law fits $f(k)$ in Eq. (25) can be seen in red. The black dotted line represents the RMS of the measured closed-loop robot tracking error that includes the sensor (camera) noise.

troller in all conditions. Using the 2-norm as a distance metric the vector valued error e_k can be converted to a scalar error

$$d_E[n] = \|E[n]\|_2 = \sqrt{e_{x,k}[n]^2 + e_{y,k}[n]^2} \quad (23)$$

where n denotes the index of the discrete data points. Next, the root mean square (RMS) can be computed

$$E_{RMS} = \sqrt{\frac{1}{N} \sum_{n=0}^N |d_E[n]|^2} \quad (24)$$

where N is the total number of data points. For the purpose of analyzing the convergence properties of the proposed method, a power law of the form

$$f(k) = Ck^{-r} \quad (25)$$

was fitted to the RMS errors calculated at each iteration of the learning process. In Eq. (25) k is the iteration, C scaling constant, and r is known as the convergence rate. Figure 7 shows the E_{RMS} for no access and access cases with $\beta = 0.25$ and $\beta = 1$. The RMS power law fits in Eq. (25), can be seen as the red line in Fig. 7. An estimate of the measured closed-loop robot tracking error that includes the sensor (camera) noise was computed in order to evaluate the performance of the human-robot collaboration. The calculation consisted of commanding the robot end effector with know trajectories, measuring the laser position with the camera, taking the difference between the known trajectory and the laser position, and calculating the standard deviations at each point in time. The error level was then calculated as the RMS value of the 2-norm of component standard deviations:

$$\Phi = \sqrt{\frac{1}{N} \sum_{n=0}^N |d_{STD}[n]|^2}, \quad d_{STD} = \sqrt{\sigma_x[n]^2 + \sigma_y[n]^2} \quad (26)$$

The computed error level was calculated as $\Phi = 0.0090$ (m), and can be seen in Fig. 7 as the black dotted line. Clearly, the robot converges to below the error level that can be achieved by the closed-loop robot controller (as expected in iterative control approaches) in both cases in Fig. 7.

A smaller value for level of assistance β results in a faster error reduction, as seen in Fig. 7, as expected, since the desired robot output $\beta q_{dx}(\omega)$ is smaller. The iteration error e_k and the level of assistance β for the x axis (the same conditions hold for the y axis) can be related by multiplying Eq. (5) by the dynamics G_x and then subtracting both sides from the desired robot output $\beta q_{dx}(\omega)$ to obtain

$$e_{x,k}(\omega) = \left[\prod_{i=1}^k (1 - \rho_{x,i}(\omega) \Delta_x(\omega)) \right] \beta q_{dx}(\omega), \quad (27)$$

where the initial error is $e_{x,0}(\omega) = \beta q_{dx}(\omega)$. Thus, a larger level of assistance β corresponds to a larger error at iteration step k , although additional iterations can be used to reduce the error further.

D. Effort analysis

The main purpose of the level of assistance β is to adjust how much the robot assists during the desired task. In order to quantify this, effort of the robot F_r and of the human F_h were calculated. First, the vector valued position of the human q_h was calculated as the difference between the laser position q and the robot position q_r :

$$q_h[n] = q[n] - q_r[n] \quad (28)$$

The vector quantity was converted to a scalar measure of distance as $d_h[n] = \|q_h[n]\|_2$, and then the human effort F_h

is defined as the RMS value of the distance d_h and the robot effort F_r given by

$$F_h = \sqrt{\frac{1}{N} \sum_{n=0}^N |d_h[n]|^2}, \quad F_r = \sqrt{\frac{1}{N} \sum_{n=0}^N |d_r[n]|^2} \quad (29)$$

where $d_r[n] = \|q_h[n]\|_2$. Human effort decreased in all cases. Moreover, the results show that the amount of effort needed from the human can be adjusted by changing the level of assistance β as seen in Figure 8, which shows human and robot efforts as a function of iteration. Notice, that with higher values of β less effort is needed from the human.

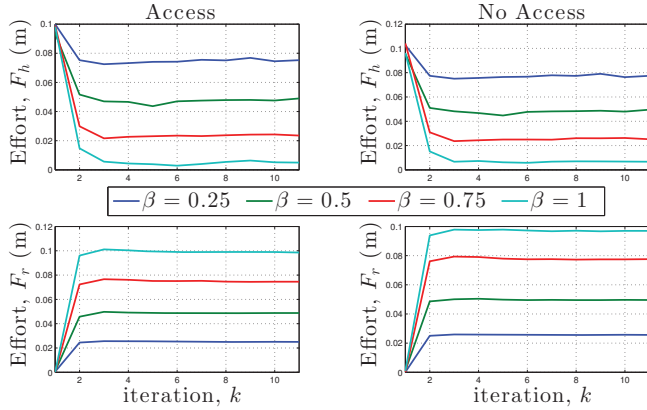


Fig. 8: Human effort F_h and robot effort F_r defined in Eq. (29) as a function of iteration step k .

V. CONCLUSION

A method to adaptively adjust the iterative learning gain of an ILC algorithm was proposed. Convergence of the proposed method was studied in the context of a human-robot collaborative output tracking task. Experimental results show that the error achieved by the human-robot collaboration was less than the closed-loop robot tracking error, and that the approach can be used to provide varying levels of robot assistance by selecting the desired human-robot collaboration level. Future work will include multiple human subject trials and the quantification of the effect of human learning on the convergence of the robot's learning.

REFERENCES

- [1] A. Tapus, M. J. Mataric, and B. Scassellati, "Socially assistive robotics [grand challenges of robotics]," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 1, pp. 35–42, 2007.
- [2] "User-centered design of a dynamic-autonomy remote interaction concept for manipulation-capable robots to assist elderly people in the home," *Journal of Human-Robot Interaction*, vol. 1, no. 1, DOI 10.5898/JHRI.1.1.Mast, pp. 96–118, 2012.
- [3] N. Ezer, A. D. Fisk, and W. Rogers, "More than a servant: Self-reported willingness of younger and older adults to having a robot perform interactive and critical tasks in the home," *Proceedings of the Human Factors and Ergonomics Society 53rd Annual Meeting*, no. DOI 10.1518/107118109X12524441079382, pp. 136–140, 2009.

- [4] C. Martens, O. Prenzel, and A. Graser, "The Rehabilitation Robots FRIEND-I & II: Daily Life Independence through Semi-Autonomous Task-Execution," *Rehabilitation Robotics, Sashi S Kommu (Ed.), ISBN: 978-3-902613-04-2*, pp. 137–162, 2007.
- [5] J. Eriksson, M. J. Mataric, and C. Winstein, "Hands-off assistive robotics for post-stroke arm rehabilitation," in *Proc. IEEE International Conference on Rehabilitation Robotics (ICORR'05)*, 2005, pp. 21–24.
- [6] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: a survey," *Foundations and trends in human-computer interaction*, vol. 1, no. 3, pp. 203–275, 2007.
- [7] H.-S. Ahn, Y. Chen, and K. L. Moore, "Iterative learning control: brief survey and categorization," *IEEE Transactions on Systems Man and Cybernetics part C Applications and Reviews*, vol. 37, no. 6, p. 1099, 2007.
- [8] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic systems*, vol. 1, no. 2, pp. 123–140, 1984.
- [9] J. Van Den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2074–2081.
- [10] J. Wang, X. Xu, D. Liu, Z. Sun, and Q. Chen, "Self-learning cruise control using kernel-based least squares policy iteration," *Control Systems Technology, IEEE Transactions on*, vol. 22, no. 3, pp. 1078–1087, 2014.
- [11] M. B. Wiggin, G. S. Sawicki, and S. H. Collins, "An exoskeleton using controlled energy storage and release to aid ankle propulsion," in *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–5.
- [12] D. Ragonesi, S. Agrawal, W. Sample, and T. Rahman, "Quantifying anti-gravity torques for the design of a powered exoskeleton," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 21, no. 2, pp. 283–288, 2013.
- [13] J. Lai, M. Schoen, A. Perez Gracia, D. Naidu, and S. Leung, "Prosthetic devices: challenges and implications of robotic implants and biological interfaces," *Proc Inst Mech Eng*, vol. 221(2), pp. 173–183, Feb, 2007.
- [14] C. Perez-Maldonado, A. Wexler, and S. Joshi, "Two dimensional cursor-to-target control from single muscle site sEMG signals," *IEEE Transactions of Neural Systems and Rehabilitation Engineering*, vol. 18(2), pp. 203–209, April, 2010.
- [15] J. C. Sanchez, B. Mahmoudi, J. DiGiovanna, and J. C. Principe., "Exploiting co-adaptation for the design of symbiotic neuroprosthetic assistants," *Neural Networks*, vol. 22, pp. 305–315, 2009.
- [16] N. A. Kirsch, N. A. Alibeji, and N. Sharma, "Model predictive control-based dynamic control allocation in a hybrid neuroprosthesis," *Proceedings of the ASME 2014 Dynamic Systems and Control Conference, San Antonio, TX*, pp. 1–8, October 22-24, 2014.
- [17] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, p. 0278364913495721, 2013.
- [18] R. B. Warrier and S. Devasia, "Iterative learning from novice human demonstrations for output tracking," August 2015, in Print.
- [19] S. Tien, Q. Zou, and S. Devasia, "Iterative control of dynamics-coupling-caused errors in piezoscanners during high-speed afm operation," *Control Systems Technology, IEEE Transactions on*, vol. 13, no. 6, pp. 921–931, 2005.
- [20] S. Devasia, "Iterative learning control with time-partitioned update for collaborative output tracking," *Automatica*, vol. 69C, pp. 258–264, 2016.
- [21] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.
- [22] S. Timoshenko, *Strength of materials*. New York, 1930.
- [23] A. T. Bahill and J. D. McDonald, "Smooth pursuit eye movements in response to predictable target motions," *Vision research*, vol. 23, no. 12, pp. 1573–1583, 1983.