2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Learning-Based Controller Optimization for Repetitive Robotic Tasks

Xiaocong Li[1], Haiyue Zhu[1], Jun Ma[2], Tat Joo Teo[3], Chek Sing Teo[1], Masayoshi Tomizuka[2], Tong Heng Lee[3]

***Abstract*— Dynamic control for robotic automation tasks is traditionally designed and optimized with a model-based approach, and the performance relies heavily upon accurate system modeling. However, modeling the true dynamics of increasingly complex robotic systems is an extremely challenging task and it often renders the automation system to operate in a non-optimal condition. Notably, many industrial robotic applications involve repetitive motions and constantly generate a large amount of motion data under the non-optimal condition. These motion data contain rich information, and therefore an intelligent automation system should be able to learn from these non-optimal motion data to drive the system to operate optimally in a data-driven manner. In this paper, we propose a learning-based controller optimization algorithm for repetitive robotic tasks. To achieve this, a multi-objective cost function is designed to take into consideration both the trajectory tracking accuracy and smoothness, and then a data-driven approach is developed to estimate the gradient and Hessian based on the motion data for optimization without relying on the dynamic model. Experiments based on a magnetically-levitated nanopositioning system are conducted to demonstrate the effectiveness and practical appeals of the proposed algorithm in repetitive robotic automation tasks.**

I. INTRODUCTION

Dynamics modeling is one of the most fundamental problems in robotics [1]. Based on the dynamic equations which explicitly describe the relationship between the force and motion, various kinds of dynamic control algorithms can be designed [2]–[4], i.e., a model-based approach. However, obtaining the true model of present-day increasingly complex robots, e.g. the quadruped robots [5] and soft robots [6], can be an extremely challenging task. Therefore, the researchers usually have to spend a significant amount of time to fine-tune the controllers to compensate for the modeling inaccuracies.

To address this issue, many learning-based data-driven methods have been developed wherein the dynamic model is not known, yet the optimal policy and parameters can be attained. These methods are generally based on self-learning from past non-optimal control to achieve improved performance in robot control. A promising trend in this line of research is deep reinforcement learning [7], wherein a neural network control policy is learned from the experimental data as well as simulation. Also, neural networks (NN) can be adopted in trajectory tracking tasks e.g., [8], [9]. One of the limitations in NN-based control is the need for a large amount of motion data. Furthermore, the stability issue is another challenge often faced in applications that require safety guarantees. It is also worthwhile to point out that various learning-based kinematic control approaches are available in the literature, such as [10], [11]. Our work, on the other hand, is mainly focused on learning-based dynamic control.

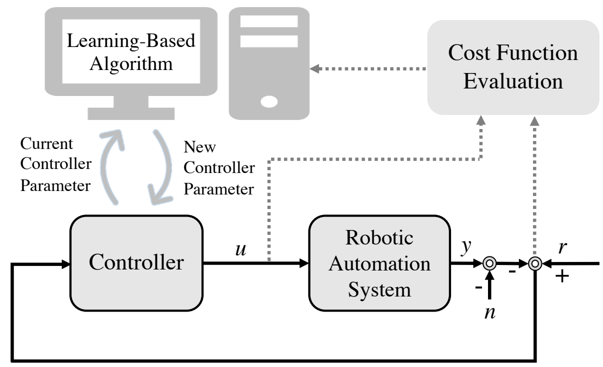

Fig. 1. Learning-based optimization algorithm for repetitive robotic tasks.

Many robotic applications, such as welding, painting, and circuit-board assembly, consist of repetitive motions and hence can adopt less computationally expensive methods. For example, the iterative learning control (ILC) is one of the effective methods that is widely used in robotic arms [12], [13] and soft robots [6], [14]. It is capable of learning from the trajectory tracking data collected in the previous iteration to improve the tracking accuracy in the next iteration. However, the ILC provides only the feedforward control policy and the feedback controller is typically not optimized. Nevertheless, it can serve as a useful add-on module once the feedback controller is optimized. In [15], [16], the authors developed a Gaussian-process-based optimization method applied to quadrotors (potentially applicable to other robotic systems as well). While the algorithm is very effective in safety guarantees, the convergence rate is relatively slow. Several other Gaussian-process-based approaches and applications

This work is supported by Collaborative Research Project U18-R-030SU under SIMTech-NUS Joint Lab on Precision Motion Systems (U12-R-024JL).*(Corresponding author: Haiyue Zhu.)*

[1]Xiaocong Li, Haiyue Zhu, and Chek Sing Teo are with the Mechatronics Group, Singapore Institute of Manufacturing Technology (SIMTech), Agency for Science, Technology and Research (A*STAR), Singapore 138634 {li_xiaocong, zhu_haiyue, csteo}@SIMTech.a-star.edu.sg

[2]Jun Ma and Masayoshi Tomizuka are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720, USA {jun.ma, tomizuka}@berkeley.edu

[3]Tat Joo Teo and Tong Heng Lee are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 {eledttj, eleleeth}@nus.edu.sg

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

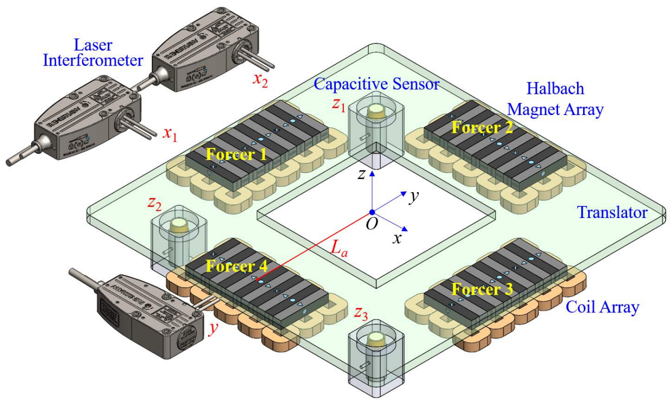

Fig. 2. Schematics of the maglev nanopositioning system.

are also available in [17]–[19]. The iterative feedback tuning (IFT), on the other hand, is one of the fast-converging data-efficient controller optimization algorithms [20]. It uses the actual robot data to provide an estimation of the gradient and Hessian of the cost function without knowing the dynamic model. The Gauss-Newton optimization is then adopted to find the optimal feedback control parameters in an iterative manner. This idea can also be applied to various control structures other than the proportional–integral–derivative (PID) controller, e.g., constrained linear quadratic regulator (LQR) [21], fractional model reference control (FMRC) [22], feedforward control [23], [24] and disturbance observer [25]–[27]. Nevertheless, the majority of the work still focuses on the more widely used PID controller, for instance, the path-tracking control of networked industrial robots in [28] and compliant rehabilitation robots in [29], etc. However, these works focused primarily on high-precision tracking and less on smooth tracking. In fact, in many automation applications such as chips inspection, welding, polishing, etc., both accurate and smooth tracking is required and challenges in guaranteeing both criteria remain; several model-based approaches in this aspect are already available in [30]–[32].

The main contribution of this work is to develop a data-driven feedback control optimization algorithm to provide accurate and smooth trajectory tracking (e.g., S-curve trajectory [33]) in repetitive robotic tasks. The cost function considers the tracking error as well as the control effort variation and the optimization process is fast and efficient. An overview of the algorithm is shown in Fig. 1. We demonstrate the proposed algorithm with a magnetically-levitated (maglev) system as shown in Fig. 2, which is a typical stage for nanopositioning [34]–[36]. Its potential applications include wafer lithography and organic light-emitting diode (OLED) manufacturing [37]–[39]. It is further noted that the proposed data-driven optimization is generic and could potentially be applied to other repetitive robotics tasks as well.

II. Magnetically-Levitated Nanopositioning System

In this section, the design principle of the maglev nanopositioning system is presented which enables a single-input single-output (SISO) control design in Section III. As illustrated in Fig. 2, the 6-DOF positioning (including 3 translational DOF and 3 rotational DOF) is achieved by four forcers, and each forcer consists of a magnet-coil pair. The square coil array is stationary and the currents are grouped into two phases. The Halbach permanent magnet array is mounted with the translator, which generates the 2-DOF sinusoidal magnetic field on the bottom side of the array while canceling the field to near zero on the other side. Each forcer is able to provide coupled 2-DOF actuation forces, i.e., the vertical and horizontal forces. For this design, the planar motion range is unlimited provided there are enough square coils. More details can be found in [40].

The force within one square coil can be expressed based on the relative position (x, z) between the coil and the magnet arrays as

$$\begin{aligned} F_x^c(x,z) &= K_x(x,z)I, \\ F_z^c(x,z) &= K_z(x,z)I. \end{aligned} \tag{1}$$

Here, I is the current, and $K_x(x,z)$ and $K_z(x,z)$ are the force constants for $x-$ and $z-$axes, respectively. The total force generated can be derived as

$$\begin{aligned} \begin{bmatrix} F_x^f(x,z) \\ F_z^f(x,z) \end{bmatrix} &= N \begin{bmatrix} K_x(x,z) & K_x(x+3\tau,z) \\ K_z(x,z) & K_z(x+3\tau,z) \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}, \\ \boldsymbol{F^f}(x,z) &= N\boldsymbol{\Phi_K}(x,\,z)\boldsymbol{I}, \end{aligned} \tag{2}$$

where $\boldsymbol{F^f}(x,z) = [F_x^f(x,z)\ F_x^f(x,z)]^T$, $\boldsymbol{I} = [I_1\ I_2]^T$, I_1 and I_2 are the currents of Phase 1 and 2, respectively. N denotes the number of effective coils in each phase and

$$\boldsymbol{\Phi_K}(x,\,z) = \begin{bmatrix} K_x(x,z) & K_x(x+3\tau,z) \\ K_z(x,z) & K_z(x+3\tau,z) \end{bmatrix}. \tag{3}$$

With the derived force model (2) on each forcer, the total force and torque acting on the translator can be calculated as

$$\begin{aligned} &F_x^t = F_x^{f_1} + F_x^{f_3}, \quad F_y^t = F_x^{f_2} + F_x^{f_4}, \\ &F_z^t = F_z^{f_1} + F_z^{f_2} + F_z^{f_3} + F_z^{f_4}, \\ &T_x^t = (F_z^{f_4} - F_z^{f_2})L_a, \quad T_y^t = (F_z^{f_3} - F_z^{f_1})L_a, \\ &T_z^t = (F_x^{f_1} - F_x^{f_2} - F_x^{f_3} + F_x^{f_4})L_a, \end{aligned} \tag{4}$$

where L_a represents the arm of torque as indicated in Fig. 2.

The 6-DOF motion control is realized through the control of total force and torque vector $[F_x^t\ F_y^t\ F_z^t\ T_x^t\ T_y^t\ T_z^t]$, where the total force and torque will be allocated to each forcer through the inverse relationship of (4) as,

$$\begin{aligned} F_x^{f_1} &= \tfrac{F_x^t}{2} + \tfrac{T_z^t}{4L_a}, & F_z^{f_1} &= \tfrac{F_z^t}{4} - \tfrac{T_y^t}{2L_a}, \\ F_x^{f_2} &= \tfrac{F_y^t}{2} - \tfrac{T_z^t}{4L_a}, & F_z^{f_1} &= \tfrac{F_z^t}{4} - \tfrac{T_x^t}{2L_a}, \\ F_x^{f_3} &= \tfrac{F_x^t}{2} - \tfrac{T_z^t}{4L_a}, & F_z^{f_1} &= \tfrac{F_z^t}{4} + \tfrac{T_y^t}{2L_a}, \\ F_x^{f_4} &= \tfrac{F_y^t}{2} + \tfrac{T_z^t}{4L_a}, & F_z^{f_1} &= \tfrac{F_z^t}{4} + \tfrac{T_x^t}{2L_a}. \end{aligned} \tag{5}$$

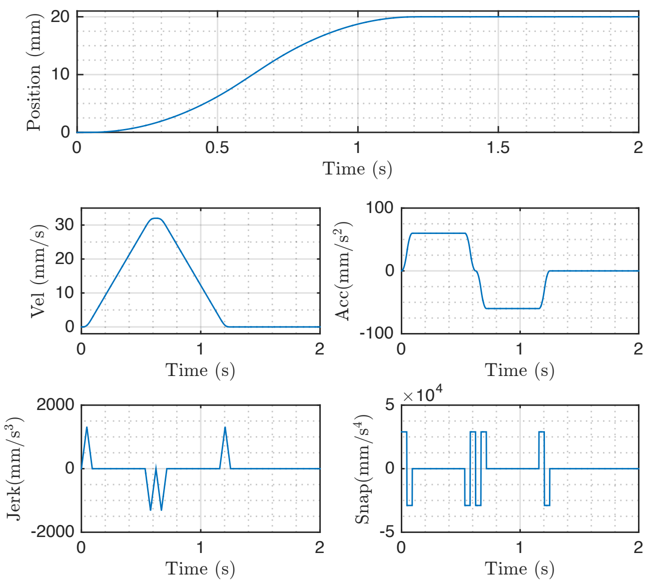

Fig. 3. 4th-order S-curve reference used in repetitive robotic tasks. The trajectory is defined with limited jerk and snap in order to guarantee motion smoothness.

Physically, the local force $\boldsymbol{F}^{\boldsymbol{f_i}}$, $i = 1, 2, 3$, and 4 for each forcer, is generated by energizing the two-phase current $\boldsymbol{I_i} = [I_{i1}\ I_{i2}]^T$ on each forcer. According to (2),

$$\boldsymbol{I_i} = \boldsymbol{\Phi_K}(x, z)^{-1}\boldsymbol{F}^{\boldsymbol{f_i}}/N. \tag{6}$$

The maglev position sensing is done by three laser interferometers (x_1, x_2, y) and three capacitive sensors (z_1, z_2, z_3) as shown in Fig. 2. With the 6-axis real-time measurement, each DOF can be controlled individually as a SISO system.

III. Learning-Based Controller Optimization Algorithm

This section presents the methodology and technical details of the learning-based optimization algorithm. First, to achieve smooth and accurate tracking, the reference profile r should be designed as an S-curve as shown in Fig. 3. Then, the feedback controller need to be designed and optimized to take into account both the control effort variation and the tracking accuracy. Hence, the cost function is defined as:

$$J({}^{\mathbf{i}}\rho) = \underbrace{w_1 e({}^{\mathbf{i}}\rho)^T \cdot e({}^{\mathbf{i}}\rho)}_{J_e} + \underbrace{w_2 \dot{u}({}^{\mathbf{i}}\rho)^T \cdot \dot{u}({}^{\mathbf{i}}\rho)}_{J_{\dot{u}}}, \tag{7}$$

where ${}^{\mathbf{i}}\rho$ is the feedback control parameter in the $\mathbf{i}^{\text{th}}$ iteration, and $J({}^{\mathbf{i}}\rho)$ is the overall cost including both J_e and $J_{\dot{u}}$ with weightings w_1 and w_2. $e({}^{\mathbf{i}}\rho)$ is the tracking error in the $\mathbf{i}^{\text{th}}$ iteration, $u({}^{\mathbf{i}}\rho)$ is the control effort and $\dot{u}({}^{\mathbf{i}}\rho)$ is the change of control effort. As shown in Fig. 1, the robotic automation system $P(s)$ is controlled by

$$C(s, \rho) = \rho^T \bar{C}(s), \tag{8}$$

where s is the Laplace variable, ρ is the control parameter. Assuming an accurate model $P(s)$ is not available, our objective is to use the experimental data collected (e.g. u and y etc.) to obtain the parameters ρ that minimizes $J(\rho)$ (7), i.e., to find

$$\rho^\star = \arg\min_\rho J(\rho). \tag{9}$$

From (7), we can obtain the gradient of $J({}^{\mathbf{i}}\rho)$ as

$$\begin{aligned}\nabla J({}^{\mathbf{i}}\rho) = {} & 2w_1[\nabla\, {}^{\mathbf{i}}e({}^{\mathbf{i}}\rho)]^T \cdot {}^{\mathbf{i}}e({}^{\mathbf{i}}\rho) \\ & + 2w_2[\nabla\, {}^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho)]^T \cdot {}^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho),\end{aligned} \tag{10}$$

and the Hessian is

$$\begin{aligned}\nabla^2 J({}^{\mathbf{i}}\rho) = {} & 2w_1[\nabla\, {}^{\mathbf{i}}e({}^{\mathbf{i}}\rho)]^T \cdot \nabla\, {}^{\mathbf{i}}e({}^{\mathbf{i}}\rho) \\ & + 2w_2[\nabla\, {}^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho)]^T \cdot \nabla\, {}^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho).\end{aligned} \tag{11}$$

$\nabla J({}^{\mathbf{i}}\rho)$ and $\nabla^2 J({}^{\mathbf{i}}\rho)$ enable us to make use of the Gauss-Newton optimization algorithm [41]:

$${}^{\mathbf{i+1}}\rho = {}^{\mathbf{i}}\rho - {}^{\mathbf{i}}\gamma(\nabla^2 J({}^{\mathbf{i}}\rho))^{-1}\, \nabla J({}^{\mathbf{i}}\rho), \tag{12}$$

where ${}^{\mathbf{i+1}}\rho$ is the new parameter and ${}^{\mathbf{i}}\gamma$ is the learning rate. From (10) and (11), we note that $\nabla\, {}^{\mathbf{i}}e({}^{\mathbf{i}}\rho)$, $\nabla\, {}^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho)$, ${}^{\mathbf{i}}e({}^{\mathbf{i}}\rho)$ and ${}^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho)$ are required. While ${}^{\mathbf{i}}e({}^{\mathbf{i}}\rho)$ and ${}^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho)$ can be collected easily from laser interferometer measurement and the controller software, obtaining $\nabla\, {}^{\mathbf{i}}e({}^{\mathbf{i}}\rho)$ and $\nabla\, {}^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho)$ is less straightforward and need to be estimated using the motion data. Similar to the IFT approach, the gradient of e is derived as:

$$\nabla^{\mathbf{i}} e({}^{\mathbf{i}}\rho) = \frac{-P\frac{\partial C({}^{\mathbf{i}}\rho)}{\partial^{\mathbf{i}}\rho}}{[1 + PC({}^{\mathbf{i}}\rho)]^2} \cdot r = -\frac{P\frac{\partial C({}^{\mathbf{i}}\rho)}{\partial^{\mathbf{i}}\rho}}{1 + PC({}^{\mathbf{i}}\rho)} \cdot {}^{\mathbf{i}}e({}^{\mathbf{i}}\rho). \tag{13}$$

$\nabla^{\mathbf{i}} e({}^{\mathbf{i}}\rho)$ can then be obtained by setting ${}^{\mathbf{i}}e({}^{\mathbf{i}}\rho)$ as the new reference signal, and we can derive the following:

$$\nabla^{\mathbf{i}} e({}^{\mathbf{i}}\rho) = -\frac{\partial C({}^{\mathbf{i}}\rho)}{\partial^{\mathbf{i}}\rho} \cdot \frac{1}{C({}^{\mathbf{i}}\rho)} \cdot y_s, \tag{14}$$

where y_s is the position signal collected for this *special experiment*. The gradient of ${}^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho)$ can also be derived as

$$\begin{aligned}\nabla^{\mathbf{i}} \dot{u}({}^{\mathbf{i}}\rho) &= \frac{\frac{\partial C({}^{\mathbf{i}}\rho)}{\partial^{\mathbf{i}}\rho}[1 + PC({}^{\mathbf{i}}\rho)]}{[1 + PC({}^{\mathbf{i}}\rho)]^2} \cdot \dot{r} - \frac{P\frac{\partial C({}^{\mathbf{i}}\rho)}{\partial^{\mathbf{i}}\rho}C({}^{\mathbf{i}}\rho)}{[1 + PC({}^{\mathbf{i}}\rho)]^2} \cdot \dot{r} \\ &= \frac{\partial C({}^{\mathbf{i}}\rho)}{\partial^{\mathbf{i}}\rho} \frac{1}{1 + PC({}^{\mathbf{i}}\rho)} \cdot \dot{e}.\end{aligned} \tag{15}$$

We can obtain $\nabla^{\mathbf{i}} \dot{u}({}^{\mathbf{i}}\rho)$ using the same *special experiment* as

$$\nabla^{\mathbf{i}} \dot{u}({}^{\mathbf{i}}\rho) = \frac{\partial C({}^{\mathbf{i}}\rho)}{\partial^{\mathbf{i}}\rho} \cdot \frac{1}{C({}^{\mathbf{i}}\rho)} \cdot \dot{u}_s, \tag{16}$$

where u_s denotes the control effort of the *special experiment*. Note that $\nabla^{\mathbf{i}} e({}^{\mathbf{i}}\rho)$ and $\nabla^{\mathbf{i}} \dot{u}({}^{\mathbf{i}}\rho)$ are estimated based on the motion data only without using the dynamic model $P(s)$. In addition, ${}^{\mathbf{i}}e({}^{\mathbf{i}}\rho)$ and ${}^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho)$ are available based on the laser interferometer measurement. Hence, $\nabla J({}^{\mathbf{i}}\rho)$ and $\nabla^2 J({}^{\mathbf{i}}\rho)$ could also be estimated according to (10) and (11). It should be noted that another trial with the reference r is needed for obtaining unbiased estimate of the cost function gradient; detailed proof is provided in the extended journal version [42]. A step-by-step guide for implementing the proposed algorithm is described in **Algorithm 1**.

Algorithm 1 Learning-Based Controller Optimization Algorithm

1) Initialize $\mathbf{i} = 0$ and choose the baseline feedback control parameters ${}^{\mathbf{0}}\rho$.
2) Conduct the normal operation with r as the reference and obtain $y^{\mathbf{1}}$ and $e^{\mathbf{1}}$.
3) Calculate $J({}^{\mathbf{i}}\rho)$ and stop if the percentage redudction is not siginificant compared with the previous iteration.
4) Use $e^{\mathbf{1}}$ as the new reference and collect $y^{\mathbf{2}}$.
5) Calculate $\nabla\, {}^{\mathbf{i}}e({}^{\mathbf{i}}\rho)$ and $\nabla^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho)$ using (14) and (16).
6) Perform another trial with r as the reference and measure $e^{\mathbf{3}}$ and $u^{\mathbf{3}}$.
7) Calculate $\nabla J({}^{\mathbf{i}}\rho)$ and $\nabla^2 J({}^{\mathbf{i}}\rho)$ according to (10) and (11), where ${}^{\mathbf{i}}e({}^{\mathbf{i}}\rho)$, ${}^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho)$ are obtained from Step 6 and $\nabla\, {}^{\mathbf{i}}e({}^{\mathbf{i}}\rho)$,$\nabla\, {}^{\mathbf{i}}\dot{u}({}^{\mathbf{i}}\rho)$ are obtained from Step 5.
8) Optimize according to (12), and obtain the new parameter value.
9) Continue with the next iteration starting from Step 2.

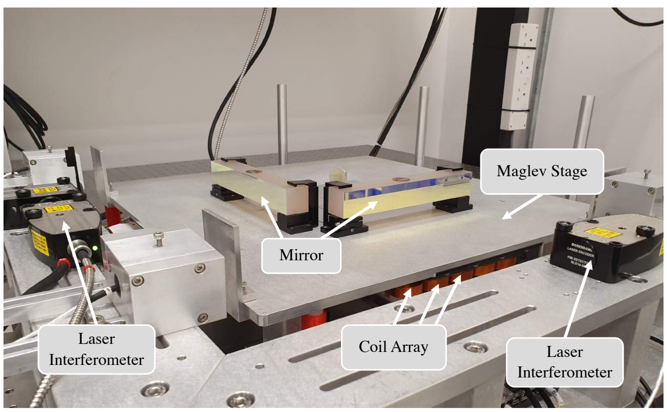

Fig. 4. The 6-DOF maglev stage for experimental validation of the proposed algorithm.

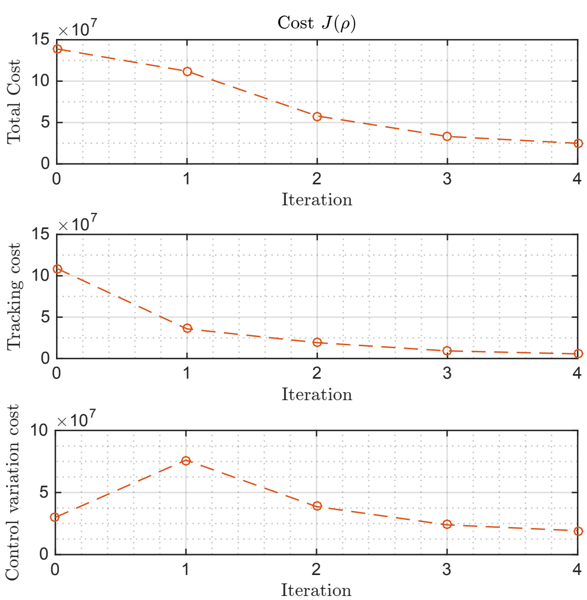

Fig. 5. Cost $J(\rho)$ convergence diagram.

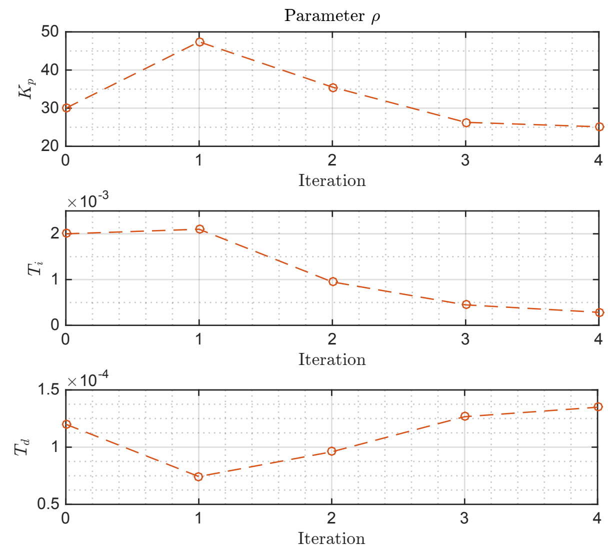

Fig. 6. Controller parameter ρ convergence diagram.

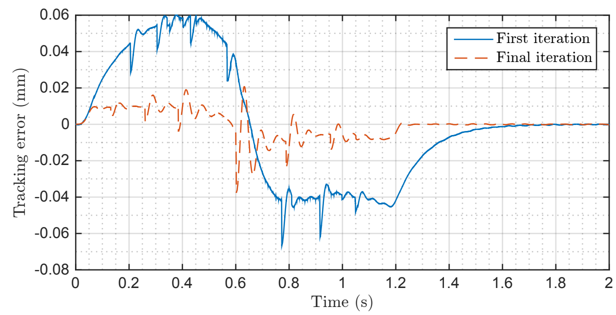

Fig. 7. Tracking performance comparison before and after the the proposed optimization.

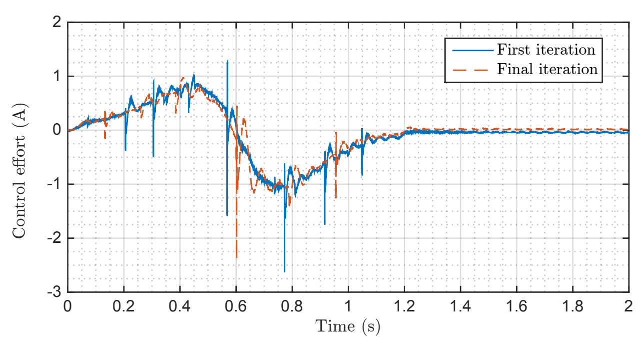

Fig. 8. Control effort comparison before and after the the proposed optimization.

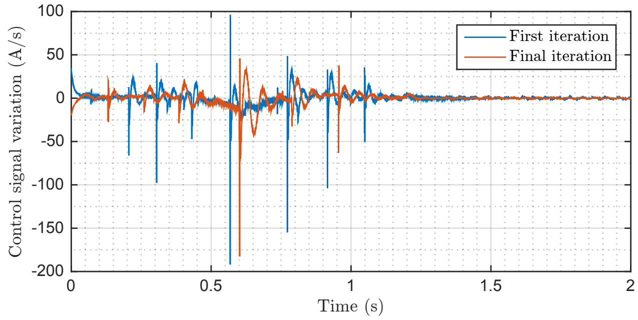

Fig. 9. Control effort variation comparison before and after the the proposed optimization.

IV. Experimental Validation

In this section, a simple single-axis SISO point-to-point tracking control is used to demonstrate the effectiveness of the proposed algorithm. As mentioned earlier, this method is generic and could potentially be applied to other repetitive robotic tasks; therefore, we make the experimental study rather general and less specific to this maglev nanopositioning stage.

In the experiment, the National Instruments (NI) PXI-8110 controller is used with a 5kHz sampling rate. The Renishaw fiber optic laser interferometers and Lion Precision capacitive sensors are used for position measurement with high-precision. The overall setup is shown in Fig. 4. The feedback controller in our experiments adopts a PID control structure used in LabVIEW as well as many other industrial controllers. Nevertheless, the algorithm is applicable to various types of feedback controllers with the form in (8). The control effort $u(t)$ is

$$u(t) = K_p\big(e(t) + \frac{1}{T_i}\int_0^t e(t')dt' + T_d\frac{de(t)}{dt}\big), \quad (17)$$

and it can be expressed using the same form in (8) as

$$C(s,\rho) = \rho^T\bar{C}(s) = \begin{bmatrix}K_p & K_p/T_i & K_pT_d\end{bmatrix}\begin{bmatrix}1\\1/s\\s\end{bmatrix}. \quad (18)$$

TABLE I

Parameters in the maglev experiments

Parameter	Initial	Final
K_p	30	25.1221
T_i	0.002	2.8459×10^{-4}
T_d	0.00012	1.3490×10^{-4}

TABLE II

Cost reduction in the maglev experiments

Cost function	Initial	Final
Overall cost J	1.3892×10^8	2.4833×10^7
Tracking cost J_e	1.0890×10^8	5.6159×10^6
Control variation cost $J_{\dot{u}}$	3.0017×10^7	1.9217×10^7

The initial controller parameter ${}^0\rho$ is shown in Table I, providing a stable and decent trajectory tracking result. The initial controller can be designed by loop shaping if a rough system model is available [43] or tuned manually via trial-and-error. Note that the initial controller must at least be designed or tuned to be a stable controller, as the algorithm is not capable of recovering from an unstable initial control policy.

The goal is to obtain the parameter value that provides a smooth and high-precision tracking of the S-curve point-to-point reference in Fig. 3. Since we are aiming for a data-driven approach, no model information of $P(s)$ is provided to the algorithm. The result of the experiments is shown in Table II, and we can observer both the tracking accuracy and smoothness can be improved significantly. To further show the cost function reduction more clearly, the convergence diagram is plotted in Fig. 5. In addition, the convergence diagram for the feedback controller parameters is shown in Fig. 6. We can therefore see both the parameters and the cost function converge fast and efficiently. Fig. 7 shows the improvement of tracking accuracy before and after the optimization. Here, the small sharp spikes may come from computational delays or laser interferometer signal losses. Fig. 8 and Fig. 9 shows the control effort and it variation, respectively. The improvement in smoothness can also be clearly observed, although it is relatively less significant compared with the improvement in the tracking error. Hence, we conclude that this proposed algorithm is able to provide an accurate and smooth trajectory tracking effectively.

V. Conclusion

In this work, we propose a learning-based data-driven controller optimization algorithm for improving the performance of robots doing repetitive tasks. Without relying upon any *a priori* dynamic model knowledge, the learning-based algorithm is able to provide a fast and efficient feedback controller optimization based on the rich information obtained under the prevailing non-optimal conditions. The cost function in this work only takes into account the smooth and accurate tracking and therefore other criteria deemed necessary (e.g. fast settling, minimum control effort, etc.) for various other applications may be further explored for future works.

Acknowledgements

The authors would like to acknowledge the funding support from the Collaborative Research Project U18-R-030SU under the SIMTech-NUS Joint Lab on Precision Motion Systems (U12-R-024JL).

References

[1] M. W. Spong and M. Vidyasagar, *Robot dynamics and control*. John Wiley & Sons, 2008.

[2] Q. M. Ta and C. C. Cheah, "Stochastic control for orientation and transportation of microscopic objects using multiple optically driven robotic fingertips," *IEEE Transactions on Robotics*, pp. 1–12, 2019.

[3] J. Hu, C. Li, Z. Chen, and B. Yao, "Precision motion control of a 6-dofs industrial robot with accurate payload estimation," *IEEE/ASME Transactions on Mechatronics*, 2020.

[4] X. Lyu, J. Zhou, H. Gu, Z. Li, S. Shen, and F. Zhang, "Disturbance observer based hovering control of quadrotor tail-sitter VTOL UAVs using H_∞ synthesis," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2910–2917, 2018.

[5] T. Boaventura, J. Buchli, C. Semini, and D. G. Caldwell, "Model-based hydraulic impedance control for dynamic robots," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1324–1336, 2015.

[6] F. Angelini, C. Della Santina, M. Garabini, M. Bianchi, G. M. Gasparri, G. Grioli, M. G. Catalano, and A. Bicchi, "Decentralized trajectory tracking control for soft robots interacting with the environment," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 924–935, 2018.

[7] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[8] Q. Li, J. Qian, Z. Zhu, X. Bao, M. K. Helwa, and A. P. Schoellig, "Deep neural networks for improved, impromptu trajectory tracking of quadrotors," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5183–5189.

[9] A. Sarabakha and E. Kayacan, "Online deep learning for improved trajectory tracking of unmanned aerial vehicles using expert knowledge," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7727–7733.

[10] M. Jolaei, A. Hooshiar, J. Dargahi, and M. Packirisamy, "Toward task autonomy in robotic cardiac ablation: Learning-based kinematic control of soft tendon-driven catheters," *Soft Robotics*, 2020.

[11] M. Jolaei, A. Hooshiar, A. Sayadi, J. Dargahi, and M. Packirisamy, "Sensor-free force control of tendon-driven ablation catheters through position control and contact modeling," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2020.

[12] S. Chen, Z. Wang, A. Chakraborty, M. Klecka, G. Saunders, and J. Wen, "Robotic deep rolling with iterative learning motion and force control," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5581–5588, 2020.

[13] O. Koç, G. Maeda, and J. Peters, "Optimizing the execution of dynamic robot movements with learning control," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 909–924, 2019.

[14] A. D. Marchese, R. Tedrake, and D. Rus, "Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator," *The International Journal of Robotics Research*, vol. 35, no. 8, pp. 1000–1019, 2016.

[15] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 491–496.

[16] A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe, "Automatic LQR tuning based on Gaussian process global optimization," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 270–277.

[17] M. Fiducioso, S. Curi, B. Schumacher, M. Gwerder, and A. Krause, "Safe contextual bayesian optimization for sustainable room temperature PID control tuning," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 5850–5856.

[18] M. Neumann-Brosig, A. Marco, D. Schwarzmann, and S. Trimpe, "Data-efficient autotuning with bayesian optimization: An industrial control study," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 730–740, 2019.

[19] M. Khosravi, V. Behrunani, R. S. Smith, A. Rupenyan, and J. Lygeros, "Cascade control: Data-driven tuning approach based on Bayesian optimization," *arXiv preprint arXiv:2005.03970*, 2020.

[20] H. Hjalmarsson, "Iterative feedback tuning-an overview," *International Journal of Adaptive Control and Signal Processing*, vol. 16, no. 5, pp. 373–395, 2002.

[21] J. Ma, Z. Cheng, X. Li, M. Tomizuka, and T. H. Lee, "Data-driven linear quadratic optimization for controller synthesis with structural constraints," *arXiv preprint arXiv:1912.03616*, 2019.

[22] Y. Xie, X. Tang, W. Meng, B. Ye, B. Song, J. Tao, and S. Q. Xie, "Iterative data-driven fractional model reference control of industrial robot for repetitive precise speed tracking," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 1041–1053, 2019.

[23] X. Li, S.-L. Chen, C. S. Teo, and K. K. Tan, "Data-based tuning of reduced-order inverse model in both disturbance observer and feedforward with application to tray indexing," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 7, pp. 5492–5501, 2017.

[24] H. Stearns, S. Mishra, and M. Tomizuka, "Iterative tuning of feedforward controller with force ripple compensation for wafer stage," in *10th IEEE International Workshop on Advanced Motion Control*, 2008, pp. 234–239.

[25] K. K. Tan, X. Li, S.-L. Chen, C. S. Teo, and T. H. Lee, "Disturbance compensation by reference profile alteration with application to tray indexing," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9406–9416, 2019.

[26] X. Li, S.-L. Chen, J. Ma, C. S. Teo, and K. K. Tan, "Data-driven model-free iterative tuning approach for smooth and accurate tracking," in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2018, pp. 593–598.

[27] X. Li, S.-L. Chen, C. S. Teo, and K. K. Tan, "Enhanced sensitivity shaping by data-based tuning of disturbance observer with non-binomial filter," *ISA transactions*, vol. 85, pp. 284–292, 2019.

[28] Y. Xie, J. Jin, X. Tang, B. Ye, and J. Tao, "Robust cascade path-tracking control of networked industrial robot using constrained iterative feedback tuning," *IEEE Access*, vol. 7, pp. 8470–8482, 2019.

[29] W. Meng, S. Q. Xie, Q. Liu, C. Z. Lu, and Q. Ai, "Robust iterative feedback tuning control of a compliant rehabilitation robot for repetitive ankle training," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 173–184, 2017.

[30] J. Ma, S.-L. Chen, C. S. Teo, A. Tay, A. Al Mamun, and K. K. Tan, "Parameter space optimization towards integrated mechatronic design for uncertain systems with generalized feedback constraints," *Automatica*, vol. 105, pp. 149–158, 2019.

[31] J. Ma, X. Li, W. Liang, and K. K. Tan, "Parameter space optimization towards constrained controller design with application to tray indexing," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 7, pp. 5575–5585, 2020.

[32] J. Ma, S.-L. Chen, W. Liang, C. S. Teo, A. Tay, A. Al Mamun, and K. K. Tan, "Robust decentralized controller synthesis in flexure-linked h-gantry by iterative linear programming," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1698–1708, 2018.

[33] P. Lambrechts, M. Boerlage, and M. Steinbuch, "Trajectory planning and feedforward design for electromechanical motion systems," *Control Engineering Practice*, vol. 13, no. 2, pp. 145–157, 2005.

[34] H. Zhu, T. J. Teo, and C. K. Pang, "Flexure-based magnetically levitated dual-stage system for high-bandwidth positioning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4665–4675, 2019.

[35] Q. Xu, "Adaptive integral terminal third-order finite-time sliding-mode strategy for robust nanopositioning control," *IEEE Transactions on Industrial Electronics*, 2020.

[36] J. P. Mishra, Q. Xu, X. Yu, and M. Jalili, "Precision position tracking for piezoelectric-driven motion system using continuous third-order sliding mode control," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 4, pp. 1521–1531, 2018.

[37] H. Zhu, T. J. Teo, and C. K. Pang, "Magnetically levitated parallel actuated dual-stage (maglev-pad) system for six-axis precision positioning," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 4, pp. 1829–1838, 2019.

[38] C.-W. Ha, C.-H. Kim, and J. Lim, "Experimental verification of a magnetic levitation transport system for the OLED display evaporation process under vacuum," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2786–2791, 2018.

[39] M. Bucak, A. F. Bozkurt, K. Erkan, and H. Üvet, "A new design concept of magnetically levitated 4 pole hybrid mover driven by linear motor," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 596–601.

[40] H. Zhu, T. J. Teo, and C. K. Pang, "Design and modeling of a six-degree-of-freedom magnetically levitated positioner using square coils and 1-d halbach arrays," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 440–450, 2016.

[41] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[42] X. Li, H. Zhu, J. Ma, D. T. J. Teo, C. S. Teo, M. Tomizuka, and T. H. Lee, "Data-driven multi-objective controller optimization for a magnetically-levitated nanopositioning system," *IEEE/ASME Transactions on Mechatronics*, 2020.

[43] H. Zhu, C. K. Pang, and T. J. Teo, "Analysis and control of a 6 DOF maglev positioning system with characteristics of end-effects and eddy current damping," *Mechatronics*, vol. 47, pp. 183–194, 2017.