Vibration Free Flexible Object Handling with a Robot Manipulator Using Learning Control

Daniele Ronzani1,∗ Shamil Mamedov1,∗ Jan Swevers∗

1 MECO Research Team, Dept. of Mechanical Engineering, KU Leuven, Belgium (e-mail: firstname.lastname@kuleuven.be) and DMMS lab, Flanders Make, Leuven, Belgium.

Abstract: Many industries extensively use flexible materials. Effective approaches for handling flexible objects with a robot manipulator must address residual vibrations. Existing solutions rely on complex models, use additional instrumentation for sensing the vibrations, or do not exploit the repetitive nature of most industrial tasks. This paper develops an iterative learning control approach that jointly learns model parameters and residual dynamics using only the interoceptive sensors of the robot. The learned model is subsequently utilized to design optimal point-to-point (PTP) trajectories that accounts for residual vibration, nonlinear kinematics of the manipulator and joint limits. We experimentally show that the proposed approach reduces the residual vibrations by an order of magnitude compared with optimal vibration suppression using the analytical model and threefold compared with the available state-of-the-art method. These results demonstrate that effective handling of a flexible object does not require neither complex models nor additional instrumentation.

Keywords: Iterative Learning Control, Vibration Suppression, Robotic Manipulation

1. INTRODUCTION

Innovative solutions in many industries require lighter, more durable, and often, consequently, flexible materials (Saadat and Nan, 2002). Applying standard solutions from rigid object manipulation to objects made from novel flexible materials lead to large vibrations. Existing feedback solutions require accurate sensing of the vibrations using an additional sensors and complex analytical or data-driven models. On the other hand, existing feedforward solutions increase the task execution time (Singer and Seering, 1990). Therefore, the industry can substantially benefit from new effective, yet simple solutions for flexible object handling.

In this paper we address the general problem of manipulating a flexible beam with a rigid robot arm (Kapsalas et al., 2018). We focus on solutions that do not use exteroceptive sensors for sensing vibrations of the beam – such as external force-torque sensors at the end-effector or position tracking system – only a joint torque estimator, available in the manipulator software, is used. Recently Mamedov et al. (2022) showed that using simple pendulum approximation of the beam and trajectory optimization, they can handle flexible objects better than existing methods. However, some residual vibration were still present.

Assuming that the beam handling is repetitive, this paper extends the work by Mamedov et al. (2022) and investigates whether vibrations can be further reduced by Iterative Learning Control (ILC). A typical ILC algorithm uses the output error of the current task execution to update the input of the next run (Bristow et al., 2006). Robotic manipulators have been a common application for such learning techniques since its first mention (Arimoto et al., 1984) to more recent advances (Koç et al., 2019). Generating a feasible input for robot manipulators with ILC requires the algorithm to cope with nonlinear dynamics and hard joint constraints. Wang et al. (2018) used a filter-based ILC with linearized model that demands a robust $\mathcal{H}_\infty$ design to account for such approximation. Steinhauser and Swevers (2017) obtained feasible trajectories with an optimization-based ILC formulation where the the nonlinear dynamics and joint constraints were directly accounted for. In this paper we adopt a similar optimization-based strategy. Specifically, the problem at hand requires designing a PTP trajectory for the manipulator which does not result in residual vibrations of the beam. Several ILC techniques for optimizing PTP trajectories are available, e.g. Freeman (2011), Son et al. (2013), however they do not consider residual vibrations after motion. In contrast, Van De Wijdeven and Bosgra (2008) proposed a vibration suppression ILC that is, however, based on a predefined trajectory. Nonetheless, their method accounts for residual vibrations by formulating the problem with a separate control and prediction horizon similar to the proposed ILC.

This paper proposes a vibration suppression ILC for flexible object handling with a robot manipulator. The approach exploits the generic formulation from Volckaert et al. (2013) of an explicit learning and control steps, shown to be equivalent to a norm-optimal ILC. The learning step consists of two estimation problems: the first,
to learn a simple yet effective parametric model that approximates the flexible beam and considers the nonlinear kinematics of the robot manipulator; the second, to learn an equivalent output disturbance to account for the residual dynamics. Finally, in the control step, we formulate a vibration suppression Optimal Control Problem (OCP) for PTP motions that exploits the learned dynamics and accounts for input and joints limits. Namely, we make the following contributions:

- a measurement model for the external torque induced by a flexible object at the end-effector that accounts for the estimation error of the external torque provided by the manipulator software;
- a generalization of OCP formulation from (Mamedov et al., 2022) that leverages the learned residual dynamics and exploit a time-optimal-like formulation to induce zero residual vibration;
- experimental validation of the ILC scheme.

This paper is organized as follows: Section 2 addresses the modeling of the robot arm, beam and external torque sensing. Section 3 discusses the proposed ILC algorithm. Section 4 presents experimental results, followed by a discussion. Section 5 concludes the paper.

2. MODELING

The vibration suppression OCP PTP motion controller requires a system model. Flexible objects are infinite dimensional systems; they are accurately modeled by partial differential equations (PDE) that are computationally demanding to solve and are seldom used in control and trajectory optimization. In robotics, for computationally tractable modeling of flexible objects, researchers make simplifying assumptions to convert PDE to ordinary differential equations (Sakawa et al., 1985; Zhou et al., 2002). The model parameters in the above-mentioned methods are obtained from CAD models because otherwise, in practice, it is difficult to estimate them. Data-driven methods for modeling the beam we adapt the simple lumped modeling approach from Mamedov et al. (2022) and briefly describe it in this section for completeness.

2.1 Manipulator dynamics

For a robot arm with $n_{\text{dof}}$ degrees of freedom (dof), let $\mathbf{q} \in \mathbb{R}^{n_{\text{dof}}}$ be the vector of joint positions and assume that: 

Assumption 1. The robot joint controller can accurately track the given joint reference trajectories.

Then, a double integrator model suffices to accurately describes the manipulators dynamics:

$$\dot{\mathbf{q}} = \mathbf{u},$$

(1)

where $\dot{\mathbf{q}} \in \mathbb{R}^{n_{\text{dof}}}$ is the vector of joint accelerations, and $\mathbf{u} \in \mathbb{R}^{n_{\text{dof}}}$ is the vector of inputs (reference joint accelerations).

2.2 Beam dynamics on the end-effector

For modeling the beam dynamics manipulated by a robot arm, we make another critical and simplifying assumption:

Assumption 2. The beam can be approximated by a simple pendulum of mass $m$ and length $l$ connected to the end-effector of a robot arm through passive revolute joint with stiffness $k$ and damping $c$.

By making such assumption, we consider only the first natural frequency of a beam and only the lateral vibrations.

To derive the pendulum dynamics using the Lagrange formulation (Sciavicco and Siciliano, 2001, Ch. 7), let $\mathbf{p}_m^b \in \mathbb{R}^3$ denote the position of the pendulum mass $m$ in the robot’s base frame

$$\mathbf{p}_m^b(q, \mathbf{\theta}, \dot{\mathbf{\theta}}, \mathbf{\theta}) := \mathbf{R}_b^0 \mathbf{R}_i^b (\mathbf{\theta}) \mathbf{i} + \mathbf{R}_i^b (\mathbf{\theta}) \mathbf{d},$$

(2)

where $[\mathbf{p}_m^b \mathbf{R}_i^b] = \mathbf{f}_b(q)$ are the position and the orientation of the origin of frame $\{b\}$, connected to the end-effector, in base frame $\{0\}$, respectively and are obtained from the forward kinematics of the manipulator, $\mathbf{R}_i^b (\mathbf{\theta}) \in SO(3)$ is a rotation matrix around $z_b$ axis, $\mathbf{\theta} = [\theta_1 \theta_2 \theta_3]^\top$ is the angular position of the pendulum and $\mathbf{i} = [1 0 0]^\top$ is a unit vector. From now on, we drop superscript $(\mathbf{\theta})$ and explicit dependence of variables on joint positions $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ for convenience. Using (2) and its time derivative, it is possible to formulate the Lagrangian. Finally, applying the Lagrange equations leads the final expression for the pendulum dynamics

$$\ddot{\mathbf{\theta}} = f_b(q, \dot{\mathbf{q}}, \mathbf{\theta}, \dot{\mathbf{\theta}}),$$

(3)

where $\omega_b \in \mathbb{R}^3$ and $\dot{\omega}_b \in \mathbb{R}^3$ are the angular velocity and acceleration of frame $\{b\}$ with respect to $\{0\}$ expressed in $\{0\}$ respectively, $\mathbf{S}(\omega_b) := \mathbf{R}_b \mathbf{R}_b^0 \in \mathbb{R}^{3 \times 3}$ is a skew-symmetric matrix and $\mathbf{g} = [0 0 -9.81]^\top$ m/s$^2$ is the gravity acceleration vector.

2.3 External torque sensing

Any control strategy that attempts to improve the manipulation of the flexible beam requires measurements or estimates of the beam motion in response to the control actions. Hence, in this subsection we develop an output model which complements the setup dynamics model from Mamedov et al. (2022).

Fig. 1. Approximation of a beam attached to the end-effector of a robot arm with a simple pendulum of length $l$ and a lumped mass $m$ connected to the end-effector through a passive revolute joint with stiffness $k$ and damping $c$. 

\[Fig. 1. \text{Approximation of a beam attached to the end-effector of a robot arm with a simple pendulum of length } l \text{ and a lumped mass } m \text{ connected to the end-effector through a passive revolute joint with stiffness } k \text{ and damping } c.\]
In absence of exteroceptive sensors, the beam dynamics can be inferred from the torque that its motion generates at frame \( \{ b \} \) along the \( Z_b \) axis (see Fig. 1). Following the pendulum approximation of the beam (3), this reaction torque in frame \( \{ b \} \) along the \( Z_b \) axis is written as:

\[
\tau := \tau_{b,z} = -c\ddot{\theta} - k\theta \tag{4}
\]

The software available in the robot manipulator drive system provides a filtered version of the external joint torque estimates \( \tau_{\text{ext}} \) (Mamedov and Mikhel, 2020; Petrea et al., 2021) that is based on the dynamic model of the robot and torque measurement either at the joint or motor side. Therefore, a filtered version \( \hat{\tau}_{b,z} \) in (4) is retrieved by using \( \tau_{\text{ext}} \) and the robots kinematics to compute the external wrench \( \mathcal{F}_b \) at the \( \{ b \} \):

\[
\mathcal{J}^b_k(q)^T \tau_{\text{ext}} = \mathcal{F}^b_k = \begin{bmatrix} \dot{F}^b_{b,z} \dot{F}^b_{b,y} \dot{z}^b_{b,z} \dot{z}^b_{b,y} \end{bmatrix}^T \tag{5}
\]

where \( \mathcal{J}^b_k \) is the manipulator Jacobian in the \( \{ b \} \) frame. As our variable of interest \( \tau \) (4) can only be retrieved from its filtered version \( \hat{\tau} := \hat{\tau}_{b,z} \), we make the following output modeling assumption:

**Assumption 3.** The available output measurement is the external torque estimate in the frame \( \{ b \} \) along the \( Z_b \) axis, filtered with a first-order low-pass filter:

\[
\dot{\hat{\tau}} = f_\tau(\hat{\tau}, \tau_e) = -a\hat{\tau} + a(\tau + \tau_e) \quad y = \hat{\tau} \tag{6}
\]

where \( a \) is the inverse of the time constant of the filter and \( \tau_e \) is a torque error given by assuming the following:

**Assumption 4.** The external torque estimator might not be correctly initialized but it converges exponentially.

\[
\hat{\tau}_e = -b\tau_e \quad \text{with} \quad \tau_{e,0} = 0 \tag{7}
\]

where \( \tau_{e,0} \) is the unknown initial estimator error.

### 2.4 Setup dynamics

The setup model describes the dynamics later used by the learning algorithm to accomplish the task at hand. For this purpose the model is enhanced with a disturbance \( d \) that affect the reaction torque (4) as

\[
\tau := -c\ddot{\theta} - k\theta + d \tag{8}
\]

in order to capture the residual dynamics. Also, the dependency of the dynamics on a parameter \( p \) is made explicit, resulting in a model of the form \( \dot{x} = f(x, u, p, d) \) and output map \( y = h(x, u, p, d) \). The setup model combines the manipulator dynamics (1), the beam dynamics (3), the reaction torque filtering dynamics (6)-(8):

\[
\dot{\hat{x}} = \begin{bmatrix} \hat{q}^T & \hat{\theta} \end{bmatrix}^T \begin{bmatrix} f_p(\cdot) & f_r(\cdot) \end{bmatrix}^T (9a)
\]

\[
y = \hat{\tau} \tag{9b}
\]

where \( x = [q^T \theta]^T \), \( \hat{q}^T \hat{\theta} \hat{\tau}_{\text{eq}} \). The state of the system is given by the system and \( u \) is the control input as shown in (1). In the rest of this paper, we use discretized the setup dynamics \( x_{k+1} = F(x_k, u_k, p, d) \) obtained from (9a) using a 4th-order Runge-Kutta integrator – and the output map \( y_k = H(x_k, u_k, p, d) := h(\cdot) \) obtained from (9b).

### 3. Iterative Learning Control

This section introduces the overall structure of the proposed ILC algorithm for vibration free handling of a flexible object and subsequently details the two separate steps of the approach. We use the following notation: \((\cdot)_i\) denotes a particular iteration \( i \in \mathbb{N} \) of the ILC; \((\cdot)_k\) denotes a particular time sample \( k \in \mathbb{Z} \) and \((\cdot)_\hat{\cdot}\) indicates that the variable is pre-computed and/or given.

#### 3.1 Algorithm/outline of the approach

**Algorithm 1** Vibration free flexible object handling ILC

1. Require: \( p^0, d^0 \) // prior parameters and disturbance
2. \( u^1 \leftarrow ocp(p^0, d^0) \)
3. \( i \leftarrow 1 \)
4. while \( i \leq i_{\text{max}} \) do
5. \( \tilde{y}^i \leftarrow \text{system_response_measurement}(u^i) \) // Learning step
6. \( p^i \leftarrow \text{parameter_estimation}(	ilde{y}^i, u^i, p^{i-1}) \)
7. \( d^i \leftarrow \text{disturbance_estimation}(\tilde{y}^i, u^i, p^i, d^{i-1}) \) // Control Step
8. \( u^{i+1} \leftarrow ocp(p^i, d^i, u^i) \)
9. \( i \leftarrow i + 1 \)
10. end while

Algorithm 1 shows the general outline of proposed ILC. It start with generating the first control input \( u^1 \) based on the given priors \( p_0 \) and \( d_0 \). Next, the algorithm proceeds by iterating between: collecting the system response measurements \( \tilde{y}^i, u^i \), learning the parametric and residual dynamics \( p^i, d_i \) and computing the next control action \( u^{i+1} \) for vibration free handling.

#### 3.2 Learning step

Traditionally, ILC learns from the tracking error to update the next input. In the proposed approach, the learning is performed by explicitly correcting the model and learning the residual dynamics given the current experiment data \( u^i, \tilde{y}^i \).

In the first learning step, the model parameters \( p^i \) are obtained by solving the following nonlinear least-square estimation problem:

\[
\min_{x, \theta, p} \sum_{k=0}^{N-1} \left[ ||\tilde{y}_k^i - y_k(\theta, p)||^2 + ||p||^2_{V_1} + ||p - p^{i-1}||^2_{V_2} \right] \tag{10a}
\]

s.t \( x_{k+1} = F(x_k, u_k^i, p) \), \( k = 0, \ldots, N - 1 \)  \( \quad y_k = H(x_k, u_k^i, p) \), \( k = 0, \ldots, N - 1 \), \( \quad f_p_{eq}(\hat{q}_0, \theta_0, p) = 0 \), \( \quad x_0 = [q_0^T \theta_0 0^T \tau_{e,0}]^T \), \( \quad p \in \mathcal{P} \quad \tag{10b} \)

where \( \mathcal{P} \) is a feasible sets for the parameters, \( f_p_{eq}(q, \theta, p) := f_p(q, 0, \theta, 0, p) \) and \( \theta_0 \) is the equilibrium position of the pendulum. The main objective is to minimize the prediction error of the parametric model, i.e. (10b) and (10c) refer to the setup model (9) where the disturbance \( d \) is ignored.

In the objective (10a) two regularization terms are added: \( r_{p,1} \), known as Tikhonov or Ridge regression (Boyd and
Vandenbergh, 2004, Chp. 6.3.2), improves the conditioning of the problem but introduces a bias: \( r_{p,2} \) regularizes the change in iteration domain to decrease the learning rate and hence to improve robustness against non-repetitive components, such as noise.

The second step in the learning procedure consists of capturing – as an equivalent disturbance \( d^e \) – the residual dynamics that cannot be described by the parametric model. This is achieved by the following estimation problem where the model parameters are now set to the estimate \( \hat{p} \) from the previous step:

\[
\min_{x, \hat{p}} \sum_{k=0}^{N_p} \left[ \| \tilde{y}_k - y_k^d \|_2^2 + \| d_k \|_w^2 \right] \tag{11a}
\]

\[
+ \sum_{k=0}^{N_p-2} \| d_{k+1} - d_k \|_{w_2}^2 \tag{11b}
\]

s.t. \( x_{k+1} = F(x_k, u_k^d, p^d, d_k), \ k = 0, \ldots, N_p - 1 \), \( y_k = H(x_k, u_k^d, p^d, d_k), \ k = 0, \ldots, N_p - 1 \), \( x_0 = \begin{bmatrix} q_0^T & 0^T & \bar{\theta}_0 & \bar{d}_0 \end{bmatrix}^T \). \tag{11c}

Similar to (10), regularization terms are added to the main objective that minimizes the prediction error. \( r_{d,1} \) penalizes the magnitude of the disturbance i.e. prevents \( d_k \) from becoming too large. \( r_{d,2} \) increases robustness and regulates the learning rate. An additional regularization term \( r_{d,3} \) is added in (11b) to penalize the rate of change in time domain of the disturbance. This term imitates a low-pass filtering effect on the disturbance estimate and increases robustness w.r.t. measurements and process noise (Boyd and Vandenbergh, 2004, Ch. 6.3.2).

### 3.3 Control step

The vibration free flexible object handling task consists of a PTP motion between two resting pose of the flexible beam connected to the end-effector. Such task is defined by the initial rest pose of the setup, determined by \( \bar{q}_0 \) and \( \bar{\theta}_0 \); and final rest pose \( \bar{p}_{b,f}, \bar{R}_{b,f} \), determined by \( \bar{q}_{f} \), with the corresponding equilibrium position of the pendulum \( \bar{\theta}_f \). We compute the feedforward joint acceleration \( u_{k+1} \) by solving the OCP that follows while using the current learned model information \( p_k, \dot{d}_k \):

\[
\min_{x, u} \phi_c(x, u, u^i) + \phi_p(\bar{\theta}, \dot{\bar{\theta}}, \tau) \tag{12a}
\]

s.t. \( x_{k+1} = F(x_k, u_k^d, p^d, \dot{d}_k), \ k = 0, \ldots, N_p - 1 \), \( \tau_k = -k_\theta \bar{\theta}_k - c_\theta \bar{\dot{\theta}}_k + \bar{d}_k \), \( x_0 = \begin{bmatrix} q_0^T & 0^T & \bar{\theta}_0 & \bar{d}_0 \end{bmatrix}^T, u_0 = 0 \), \( p_0(q_{N_p}) = \bar{p}_{b,f}, q_{N_p} = 0 \), \( u_0 = 0 \), \( k = N_p - 1, \ldots, N_p - 1 \), \( \bar{r}_O(\bar{R}_0(q_{N_p}), \bar{R}_{b,f}) = 0_{3x1} \), \( x \in \mathcal{X}, u \in \mathcal{U}, \dot{u} \in \mathcal{J} \). \tag{12b}

where \( \bar{r}_O = r_{d,z,k} + d_k \) is the prediction of the pendulum reaction torque with the equivalent disturbance \( d_k \) of the residual dynamics; \( \bar{r}_O(\cdot) \in \mathbb{R}^3 \) is a function for computing the orientation error between two frames (Sciavicco and Siciliano, 2001, Ch.3); \( \mathcal{X}, \mathcal{U}, \text{ and } \mathcal{J} \) are feasible sets for states, controls, and rate of change of controls. In the problem formulation (12), we consider a control horizon of \( N \) samples in which the motion is executed and for which a control horizon cost term in (12a) is designed to enforce a desirable motion of the robot manipulator:

\[
\phi_c(\cdot) = \sum_{k=0}^{N_p} \| x_k - x_0 \|_Q^2 + \sum_{k=0}^{N_p-1} \| u_k \|_{R_k}^2 + \sum_{k=0}^{N_p-2} \| u_{k+1} - u_k \|_{R_k}^2, \tag{13}
\]

where \( Q \in \mathbb{R}^{n_x \times n_x}, R_0 \in \mathbb{R}^{n_u \times n_u}, R_2 \in \mathbb{R}^{n_u \times n_u} \) are the weights for penalizing deviation of states from the initial state (to avoid excessive movements of the robot), inputs, prior input and jerks, respectively. Additionally, we consider an extended prediction horizon from \( N_c \) to \( N_p \), in which the controls are set to zero (12f), that is used to penalize the residual vibration occurring after the motion by means of the prediction horizon cost in (12a):

\[
\phi_p(\cdot) = \sum_{k=N_c}^{N_p-1} \gamma_k \left( \| \bar{\theta}_k - \bar{\theta}_f \|_1 + \| \rho \bar{\dot{\theta}}_k \|_1 + \rho_3 \| \bar{\tau}_k - \bar{\tau}_f \|_1 \right), \tag{14}
\]

The residual vibration are observed through \( \bar{\theta}_k, \dot{\bar{\theta}}_k \) and \( \bar{\tau}_k \) and hence all three are considered in (14), each with their own weight \( \rho_1, \rho_2 \) and \( \rho_3 \) respectively. The objective terms \( \alpha_1 \) and \( \alpha_2 \) penalize the prediction of the residual vibration by the parametric model. The aim is to keep \( \bar{\theta}_k \) close to the equilibrium position \( \bar{\theta}_f \) and \( \bar{\dot{\theta}}_k \) equal to zero during the time horizon following the robot motion. Additionally, the term \( \alpha_3 \) penalizes the residual vibration as predicted by the torque \( \bar{\tau}_k \) (12c) which includes the residual dynamics given by \( \bar{d}_k \). To achieve vibration suppression \( \bar{\tau}_k \) should be equal to the equilibrium reaction torque given by \( \bar{\tau}_f = -k_\theta \bar{\theta}_f + \bar{r}_O \). Finally, note that all the terms in (14) employ the sparsity promoting \( l_1 \)-norm and are weighted by an exponentially increasing weight \( \gamma > 1 \). This is done with the purpose of promoting zero residual vibration as early as possible after finishing the robot motion, that is as close as possible after reaching time instance \( N_c \). A similar strategy is adopted in (Verschueren et al., 2018) for a time-optimal model predictive control formulation.

### 3.4 Numerical implementation

In this work, we use CasADi (Andersson et al., 2019) to formulate the optimization problems (10), (11) and (12) as nonlinear programs (NLP) following the multiple-shooting method. The NLPs are solved using the nonlinear optimization solver IPOPT (Wächter and Biegler, 2006) which implements an interior-point method. Moreover, we retrieve the computations of velocities and accelerations of the end-effector – i.e., first- and second-order kinematics required in (10), (11) and (12)– from the forward pass of the recursive Newton-Euler algorithm, which exploits the sparsity of the kinematic model unlike algorithmic differentiation. Such efficient functions for kinematics (and their derivatives) are generated by using Pinocchio (Carpentier et al., 2019). The code used in this work is publicly available on a GitHub repository.\(^2\)

---

\(^2\) https://github.com/danielREK3beam_handling_ILC
In this section, we describe the experimental setup, the task, and the ILC settings. Then, we present the experimental validation of the proposed approach and compare it with an existing solution.

4.1 Setup description

The setup used to validate our approach consists of a 7-dof Franka Emika Panda manipulator and a flexible beam with dimensions $60 \times 6 \times 0.1$ cm rigidly attached to the arm's end-effector. The beam is made of stainless steel 316L with $\rho = 6.3$ g/cm$^3$, $EI = 1.267$ N·m$^2$. The actual inputs to the setup are reference joint velocities $\dot{q}_r(t)$ retrieved by integrating the joint accelerations $u$. The given outputs from the setup are joint positions $q(t)$, velocities $\dot{q}(t)$ and estimated filtered external torques $\tau_{ext}(t)$ at 1kHz as detailed in section 2.3.

4.2 Task definitions and ILC settings

4.3 Validation

To quantify the performance of the experiments we define as metric the normalized integral of the absolute value of the zero mean residual vibrations (vibrations that persist after the end of the motion)

$$V = \frac{1}{N_r} \sum_{k=N_r}^{N+N_r} |\dot{\bar{\tau}} - \bar{\tau}|, \quad (15)$$

where $\dot{\bar{\tau}}$ is the average value of $\dot{\tau}$ and $N_r$ are samples of a sufficiently long time horizon such that it contains several of its periods in case of significant vibrations. In this paper, we consider a time window of 5 s in addition to the task motion time.

The proposed ILC algorithm combines a parametric model and a disturbance that represents the residual dynamics. To understand its functioning, we perform Algorithm 1 (ILC) and compare it to the case where the parameter estimation does not include the residual dynamics (ILC-P), i.e., $d^p = 0$. Figure 2 shows that by combining the parametric and the disturbance models, the ILC more accurately predicts the output with respect to ILC-P, especially the residual vibrations. This result motivates the need to learn the residual dynamics and leverage it via the extended prediction horizon cost (14). Figure 3 compares the performance of both ILC, ILC-P and the BASELINE. The top figures show the evolution of the residual vibrations as a function of the ILC iterations. ILC achieves nearly zero residual after a short time interval, especially compared to the first experiment that exploits the analytical model. Despite of that, ILC-P still achieve a considerable reduction of the vibration w.r.t the initial experiment and obtains a comparable vibration suppression to BASELINE. Note that ILC and ILC-P learn the model parameter by exploiting the execution of the task, while BASELINE requires ad-hoc experiments prior to the task. A visual demonstration of the experiments can be found at https://youtu.be/c8v191ND1kg.
5. CONCLUSION

This paper proposes an ILC algorithm for vibration free flexible object handling with a robot manipulator. Assuming that the beam handling is repetitive, this paper extends the work by Mamedov et al. (2022). We present a measurement model for the external torque induced by the flexible object that accounts for the estimation error introduced by the manipulator software. The model enables learning of a parametric model and residual dynamics without relying on any exteroceptive sensors. Unlike other ILC approaches, the proposed algorithm introduces a PTP optimal control strategy that accounts for residual vibration, nonlinear kinematics and physical limits of the manipulator. The approach is experimentally validated and shows a threefold improvement compared with the available state-of-the-art method. This result is mainly due to estimating and exploiting the residual dynamics. This work can provide a solution for learning PTP motion primitives useful for executing more challenging and industrially relevant handling tasks.

REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. Mathematical Programming Computation, 11(1), 1–36. doi: 10.1007/s12532-018-0139-4.

Arimoto, S., Kawamura, S., and Miyazaki, F. (1984). Bettering Operation of Robots by Learning. Journal of Robotic Systems, 1(2), 123–140. doi:10.1002/rob.4620010203.

Boyd, S. and Vandenberghe, L. (2004). Convex Optimization. Cambridge University Press.

Bristow, D.A., Tharayil, M., and Alleyne, A.G. (2006). Survey Of Iterative Learning Control: A Learning-Based Method for High-Performance Tracking Control. IEEE Control Systems, 26(3), 96–114. doi:10.1109/MCS.2006.1636313.

Carpentier, J., Saurel, G., Buondonno, G., Mirabel, J., Lamhaux, F., Stasse, O., and Mansard, N. (2019). The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In IEEE International Symposium on System Integrations (SII).

Freeman, C.T. (2011). Constrained Point-to-Point Iterative Learning Control. IFAC Proceedings Volumes, 44(1), 3611–3616. doi:10.3182/20110828-6-IT-1002.00231.

Kapsalas, C.N., Sakellarion, J.S., Koustoumpardis, P.N., and Aspragathos, N.A. (2018). An ARX-based method for the vibration control of flexible beams manipulated by industrial robots. Robotics and Computer-Integrated Manufacturing, 52, 76–91. doi:10.1016/j.rcim.2017.11.001.

Koe, O., Maeda, G., and Peters, J. (2019). Optimizing the Execution of Dynamic Robot Movements with Learning Control. IEEE Transactions on Robotics, 35(4), 909–924. doi:10.1109/TRO.2019.2906558.

Mamedov, S., Astudillo, A., Ronzani, D., Decré, W., Noël, J.P., and Swevers, J. (2022). An optimal open-loop strategy for handling a flexible beam with a robot manipulator. arXiv preprint arXiv:2210.00578.

Mamedov, S. and Mikheil, S. (2020). Practical aspects of model-based collision detection. Frontiers in Robotics and AI, 7, 571574.

Petrea, R.A.B., Bertoni, M., and Ooboe, R. (2021). On the interaction force sensing accuracy of franka emika panda robot. In IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society, 1–6. IEEE.

Saadat, M. and Nan, P. (2002). Industrial applications of automatic manipulation of flexible materials. Industrial Robot, 29(5), 434–442. doi:10.1108/1043991021040255.

Sakawa, Y., Matsuno, F., and Fukushima, S. (1985). Modeling and feedback control of a flexible arm. Journal of Robotic Systems, 2(4), 453–472. doi:10.1002/rob.4620020409.

Sciacicco, L. and Siciliano, B. (2001). Modelling and control of robot manipulators. Springer Science & Business Media.

Singer, N.C. and Seering, W.P. (1990). Preshaping Command Inputs to Reduce System Vibration. Journal of Dynamic Systems, Measurement, and Control, 112(1), 76–82. doi:10.1115/1.2894142.

Son, T.D., Ahn, H.S., and Moore, K.L. (2013). Iterative learning control in optimal tracking problems with specified data points. Automatica, 49(5), 1465–1472. doi:10.1016/j.automatica.2013.02.008.

Steinhauser, A. and Swevers, J. (2017). Iterative Learning of Feasible Time-optimal Trajectories for Robot Manipulators. IFAC-PapersOnLine, 50(1), 12095–12100. doi:10.1016/j.ifacol.2017.08.2123.

Van De Wijdeven, J. and Bosgra, O. (2008). Residual vibration suppression using Hankel iterative learning control. International Journal of Robust and Nonlinear Control, 18(10), 1034–1051. doi:10.1002/rnc.1228.

Verschueren, R., Ferreau, H.J., Zamarini, A., Mercangoz, M., and Diehl, M. (2018). A stabilizing nonlinear model predictive control scheme for time-optimal point-to-point motions. 2017 IEEE 56th Annual Conference on Decision and Control, CDC 2017, 2018-January, 2525–2530. doi:10.1109/CDC.2017.8264024.

Volckaert, M., Diehl, M., and Swevers, J. (2013). Generalization of norm optimal ILC for nonlinear systems with constraints. Mechanical Systems and Signal Processing, 39(1-2), 280–296. doi:10.1016/j.ymssp.2013.03.009.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical programming, 106(1), 25–57.

Wang, C., Zheng, M., Wang, Z., Peng, C., and Tomizuka, M. (2018). Robust iterative learning control for vibration suppression of industrial robot manipulators. Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, 140(1). doi:10.1115/1.4037265.

Zhou, T., Goldenberg, A.A., and Zu, J.W. (2002). Modal force based input shaper for vibration suppression of flexible payloads. Proceedings - IEEE International Conference on Robotics and Automation, 3(May), 2430–2435. doi:10.1109/robot.2002.1013596.