OPTIMAL AND LEARNING-BASED OUTPUT TRACKING WITH NON-PERIODIC TRACKING-TRANSITION: APPLICATION TO HIGH-SPEED PROBE-BASED NANOFABRICATION

BY JIANGBO LIU

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Mechanical and Aerospace Engineering

Written under the direction of

Qingze Zou

and approved by

____

____

____

____

New Brunswick, New Jersey

May, 2017

ABSTRACT OF THE DISSERTATION

Optimal and Learning-based Output Tracking with Non-periodic Tracking-Transition: Application to High-speed Probe-based Nanofabrication

by Jiangbo Liu

Dissertation Director: Qingze Zou

High-speed precision tracking is needed in a wide variety of motion control applications ranging from high-speed AFM(Atomic Force Microscope) operation, high-throughput manufacturing, to robotic operations. Challenges still exist in high-speed precision control of systems such as smart actuators with coupled hysteresis and dynamics. Although output tracking has been well-studied for linear systems, tracking with non-periodic tracking-transition switching for non-minimum phase linear systems still remains challenging, especially when multiple control objectives need to be achieved, including smooth transition from one output tracking session to the next one without inducing post-transition oscillations, input energy minimization without saturation under input amplitude constraint, and furthermore, minimization of the overall transition time. Moreover, further difficulties also arise in exploring the advantages of iterative learning control (ILC) in achieving precision but robust output tracking at high-speed to non-repetitive applications with online-generated desired trajectory, particularly for systems with complicated input-output behavior such as Hammerstein systems. The ILC framework can be extended, although for linear systems, to non-periodic output tracking via the superposition principle (SP), where the system response (output) to a linear combination of inputs equals to the same linear combination of the outputs to each individual inputs. Exploring the notion of the SP beyond linear systems is largely limited, as the nonlinearities are difficult to be modeled effectively and accurately. Therefore, challenges still exist in high-speed precision output tracking in emerging applications.

In this dissertation, a multi-objective optimal tracking/transition under input constraints for non-periodic tracking/transition switching problem, and a learning-based approach for tracking control of Hammerstein systems are proposed. An approach to extend the previous work on smooth output transition and smooth tracking/transition switching to further achieve minimization of both input-energy and transition time under input amplitude constraints is proposed. The constrained input optimization problem is converted to an unconstrained input minimization problem, and then solved by utilizing an improved conjugate gradient method. The total transition time is further minimized via one dimensional search. The almost superposition of Hammerstein systems (ASHS) is developed, and then exploited for precision control of hysteresis-Hammerstein systems. We showed that for Hammerstein operator satisfying a Lipschitz condition, a weak form of the ASHS—the linear combination of outputs approaches to the response to the linear combination of the corresponding inputs with a different set of combination coefficients—exists when there are enough output elements. The strict form of the ASHS—the coefficients of the output and the input combination match to each other exactly—holds for a certain choice of the inputs. The number of outputs in the ASHS is further quantified for hysteresis-Hammerstein system. We then present the realization of the ASHS for hysteresis-Hammerstein systems in the learning-based output tracking applications, based on the uniform B-splines for decomposition and the inverse Preisach modeling for superposition, where two optimizations of the ASHS for practical implementation are proposed. Moreover, for the trajectory decomposition problem arose in the ASHS, we further develop an asymptotic online trajectory decomposition (where the trajectory is only partially known at the decomposition instants) by only using one type of basis functions without truncation. The problem of trajectory approximation using only one basis function (along with its time-shifted copies) is addressed via a least-square minimization approach. The issue of truncating basis function at the boundaries is resolved via a zero-period extension (i.e., adding a zero-period to its beginning and end). It is shown that the coefficients of the basis functions at the initial portion of the extension period approach to zero as the length of the extension period increases. A sectional interactive decomposition algorithm is proposed for online trajectory decomposition through a trajectory redesign scheme.

Implementation of the frequency-domain iterative learning control (FD-ILC) in real-time for high-speed nanofabrication and AFM imaging becomes an issue as the FD-ILC involves multiple FFT/IFFTs that demand intensive online computation. An algorithm of optimal time-distributed fast Fourier transform and time-distributed inverse fast Fourier transform (OTD-FFT/TD-IFFT) is proposed to optimally distribute the FFT computation of a real-time sampled data sequence to minimize the per-sampling computational complexity without increasing the total computational complexity, and to obtain the entire Fourier transform sequence without latency. The proposed approach is extended to real-time IFFT computation, and then combined and applied to real-time FD-ILC implementation. The computational complexity analysis shows that the per-sampling computational complexity is substantially reduced by using the proposed approach.

The proposed optimal and learning-based output tracking and tracking-transition techniques can effectively achieve accurate high-speed nanofabrication. The effectiveness of the proposed techniques was demonstrated by simulation and experimental examples in accurate high-speed nanomanipulation utilizing an AFM probe driven by a piezoactuator. The optimal transition design and tracking approach was demonstrated by a simulation example in the control of the z-piezoactuator in an AFM system. It showed that the proposed approach can achieve minimal output oscillation and minimal input energy, and fast tracking under given input constraints. The proposed offline-learning based control technique was implemented to compensate for both hysteresis and dynamics in a piezoactuator system. It showed that high-speed, large-range precision tracking of hysteresis-dynamics systems, and considerable accuracy improvement when compared to PID control at different tracking speeds are achieved. A simulation example to decompose a trajectory sequentially online has demonstrated the effectiveness of the proposed online asymptotic trajectory decomposition technique. Finally, real-time implementation of FD-ILC based on an optimal time-distributed FFT was demonstrated through high-speed trajectory tracking on a piezoelectric actuator in experiments.

Dedication

For my grandma, mother, and my family

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Precision and high-speed trajectory tracking is needed in many advanced applications, such as Atomic Force Microscope(AFM) imaging [1], nanomanipulation [2], nanodeposition [3], robotic operation [4], and hard-disk control [5]. In these applications, high-speed and precision tracking is needed, for example, in the AFM imaging process, high-resolution and fast-speed sample topography imaging is required. In some of the applications, such as nanofabrication and nanopatterning, there exists application-specified trajectory to be tracked and transition between two tracking trajectory sessions to be properly designed. Nonlinearities such as hysteresis, typically involved in the piezoactuators for high-resolution control, should be well-compensated to achieve an outstanding tracking performance. Although plenty of techniques have been proposed to solve these problems, challenges still exist due to the complexity of the system dynamics (limited bandwidth and non-minimum phase), involvement of the hysteresis nonlinearities, and the online specified desired trajectory (resulting in complicate online computation). The problem of optimal trajectory design and tracking with non-periodic tracking-transition switching, when taking the input threshold constraints into consideration, becomes a constrained multi-objective optimization problem, which significantly complicates the design process, especially for non-minimum phase linear systems and nonlinear systems; The hysteresis-nonlinearity further poses challenges in high-speed and large-range tracking. In this dissertation, the optimal tracking/transition with non-periodic tracking-transition switching is proposed for non-minimum phase linear systems, and offline-learning technique based on superposition principle is developed to extend the capability of the iterative learning control for the control of hysteresis-Hammerstein systems.

These problems in trajectory-tracking/transition for linear and nonlinear system, including optimal tracking/transition for non-periodic tracking/transition switching, and high-speed and precision output tracking of systems with complicated dynamics and hysteresis, have attracted a great attention due to their importance in a wide range of applications. Many techniques are proposed to resolve the multi-objective tracking/transition with non-periodic tracking/transition switching for non-minimum phase systems. The optimal boundary states of the transition can be obtained through the optimal output transition (OOT) technique [6], which has been further extended to solve the output tracking problem with periodic tracking-transition switching [7]. Although the input energy is minimized in the OOT approach, the optimal transition trajectory obtained can be highly oscillatory for systems of lightly-damped dynamics [8], and this method is not suitable to output tracking with non-periodic tracking-transition switching. These two issues have been addressed in the recently developed inversion-based optimal output transition technique [9, 10], all these approaches cannot guarantee that the input amplitude is not saturated while the input energy is minimized, and the transition time is small when the above smooth-output, minimal-input requirements are satisfied. Therefore, there exists a need to develop techniques to further account for the input and time constraints in output tracking/transition with non-periodic tracking-transition switching.

Simultaneous hysteresis and dynamics compensation still remains challenging in the tracking control of hysteresis systems. Many limitations exist in the current techniques to compensate for hysteresis. For example, it has been proposed to use the inverse Preisach model in the controller design [11, 12], or by updating the inverse of Preisach model through an adaptive scheme [13]. These techniques, however, are limited to compensating for hysteresis in quasi-static case—accounting of both dynamics and hysteresis effects is not adequately addressed. It has been proposed to modify the inverse Preisach model to account for the rate-dependent hysteresis effect (i.e., the combined hysteresis and dynamics effects) [14]. However, the compensation of the dynamics effect is rather constrained since only a handful of the excitation frequencies are considered in the inverse-Preisach-model based controller design [14]. To improve the dynamics compensation, techniques based on an extension of neural network [15] [16] and the inverse rate-dependent Prandtl-Ishlinskii (PI) model [17] have also been proposed. In these existing approaches, however, heavy online computation and a complex modeling process—prone to modeling errors—are required. The performance-robustness trade-off involved may not be efficient in accounting for the largely quasi-static uncertainty/variation usually found in hysteresis systems such as smart actuators. These complexities (in modeling and control) and robustness related issues can be largely alleviated in the iterative learning control framework [18, 19]—as an example, the high tracking performance obtained via the ILC-based techniques can be largely maintained against model uncertainties and/or system variations through a couple of iterations. Since iterative learning control are largely limited to the repetitive operations. Therefore, there is still a need to develop an offline learning-based technique to compensate for the hysteresis and dynamics online simultaneously.

Online trajectory decomposition problem arises in the offline learning-based technique developed for the hysteresis systems. However, limitations exist in current techniques developed for online trajectory decomposition. For example, basis functions constructed by biarc-spline functions are employed to decompose the trajectory in curve machining [20]. Since only smoothness is considered for the boundary condition, the basis functions are not compactly supported to begin and end at a constant, thereby, not suitable for control applications. The limitation can be alleviated by using uniform B-splines as the basis functions in the decomposition [21, 22]. These general decomposition frameworks do not account for the objectives such as tracking precision, and high-order smoothness as needed in control applications. These control-related objectives can be mostly taken into account by extending the decomposition technique to ILCs. For example, a task-based basis function method is employed to extend ILC to micro-robotic deposition [3]. This technique is limited to specific trajectories with a small amount of repeated patterns. Rational basis functions are introduced to approximate the system transfer function and the system input so that the ILC can have the extrapolation ability [23]. Multiple basis functions are identified through ILC from different basic states of a linear time variant system with slow changes, and the interpolation of these basic inputs are used to generate inputs for other states [24]. However, considerable different basis functions are needed in the decomposition, resulting in numerous learning and computation. A decomposition technique with only a few simple decomposition basis functions is utilized to decompose the input and output trajectories in ILC, and the decomposition coefficients are updated online to accommodate for the possible change of the desired trajectory [25], and an overlapped B-splines based trajectory decomposition method is proposed to extend the implementation of ILC to more general applications [26]. However, this overlapped decomposition technique suffers from element truncation and involves multiple basis elements, which increases the complexity for control applications. Therefore, there is still a need to develop a decomposition method with minimal untruncated basis elements for online control applications.

As implementation of the frequency-domain iterative learning control (FD-ILC) in real-time for high-speed nanofabrication and AFM imaging involves multiple FFT/IFFTs, the intensive online computation complexity becomes another important issue. Limitations exist in current techniques developed for online implementation of FFT/IFFT. For example, a pipeline processor based on very large scale integration (VLSI) technology has been proposed for computing FFT of fixed-length [27] [28]. The limitations in FFT computation for fixed data lengths might be addressed through hardware development based on field-programmable gate array (FPGA) and digital signal processor (DSP) technologies, such as a radix-2 single-path delay feedback pipelined FFT/IFFT processor based on FPGA [29], and a compact online FFT algorithm based on DSP [30]. Although these FPGA/DSP-based techniques improved the flexibility in online FFT computation, specially-designed hardware is needed. Alternatively, algorithms have been proposed to improve online computation of FFT on general purpose microprocessors (e.g., Intel's x86 microprocessors), including a distributed FFT algorithm to process a large block of incoming data [31] that spreads the computation across smaller blocks of the incoming data, and a multi-rate controller based on a new discrete lifted system model to save computation by reducing the length of input data [32]. In these efforts, however, the per-sampling-period complexity is still significant, and latency is induced when the transformed (output) sequence is obtained. Therefore, there are needs to develop techniques to further improve the real time implementation of FFT/IFFT, particularly, on general-purpose microprocessors.

The dissertation is organized as follows,
In Chapter 2, an approach that achieves multi-objective optimal output tracking/transition under the input and time constraints with non-periodic tracking/transition switching for non-minimum phase linear systems is developed. The proposed approach extends the technique developed in [9] that attains an optimal smooth output tracking/transition in non-periodic tracking/transition switching. Particularly, the input-energy minimization under amplitude constraint is further obtained by optimizing a solution set obtained by the previous method [9]. We first show that such a constrained minimization can be converted into an unconstrained minimization problem. Then, the optimal solution are obtained based on an improved conjugate gradient technique [33] that guarantees the global optimization without convexity requirement for the underlined cost function (instead, a mild Lipschitz condition is needed). Then, the transition time is further minimized for the obtained optimal output transition trajectory. The proposed approach is illustrated by a simulation example of nanomanipulation utilizing a piezoactuator.
In Chapter 3, an offline-learning technique based on an almost superposition of Hammerstein systems (ASHS) is proposed for the tracking control of hysteresis systems. We show that for a Hammerstein operator satisfying the Lipschitz condition, a weak form of the ASHS exists, where the linear combination of outputs matches the response to that of the corresponding inputs with a different set of combination coefficients, provided that the outputs are Lipschitz continuous and the number of outputs are large enough. Furthermore, the strict form of the ASHS—the output and the input combination coefficients in the linear combinations match to each other exactly—holds for a certain choice of the inputs. For hysteresis-Hammerstein systems, the number of outputs needed in the ASHS can be quantified. We further present the realization of the ASHS for hysteresis-Hammerstein systems in the previewed output tracking based on the uniform B-splines for output decomposition and the inverse Preisach modeling

for input synthesis, where an optimization of the outputs selection and an online optimization of the combination coefficients of ASHS for practical implementation are also proposed. This ASHS not only is a powerful tool for Hammerstein system analysis, but can also be explored for control of these systems. As an example, we illustrate its application to simultaneous hysteresis-dynamics compensation in preview-based precision output tracking, by integrating the B-spline-based trajectory-decomposition technique for linear systems [26, 34] with the inverse Preisach model for superposition. The efficacy of the proposed approach is demonstrated by an experimental implementation on a piezoelectric actuator.

In Chapter 4, a single-basis online asymptotic trajectory approximation is further developed to resolve the remaining trajectory decomposition issue that arose in Chapter 3. The problem of trajectory approximation using only one type of basis function is addressed via a least-square minimization approach, and it is shown that arbitrary decomposition precision can be achieved through increasing the number of basis functions involved. The truncation issue of basis function in approximation is resolved via a zero-period extension (i.e., adding a zero-period to its beginning). It is shown that the coefficients of the basis functions at the initial portion of the extension period approach to zero as the length of the extension period increases. We proposed also a sectional interactive decomposition algorithm for online trajectory decomposition (where the trajectory is only partially known) through a trajectory redesign scheme. Simulation examples by using B-splines as the basis functions are employed to demonstrate the proposed decomposition method.

In Chapter 5, an OTD-FFT/TD-IFFT technique with direct application to online FD-ILC is developed. Without increasing the total computational complexity, the proposed OTD-FFT/TD-IFFT algorithm exploits the butterfly structure of FFT/IFFT to optimally distribute the FFT computation of an online sampled data sequence to each of the sampling period, such that the per-sampling-period computational complexity is minimized, and the entire Fourier transform sequence is obtained without latency.

Specifically, through the proposed OTD-FFT, the maximal per-sampling-period computational complexity for a data sequence of length 2^N is reduced from $2^N \log 2^N$ multiplications and $2^N \log 2^N$ additions to $2^{N+1}-2$ multiplications and $2^{N+1}-2$ additions, respectively. Similar idea is extended to the IFFT computation in the proposed TD-IFFT algorithm, where, with the frequency domain data of length 2^N already known, the butterfly computation of IFFT is sequentially distributed to the first $2N$ sampling periods. The proposed OTD-FFT/TD-IFFT algorithm is applied to online implementation of FD-ILC, and the computational complexity of the proposed OTD-FFT/TD-IFFT and the FD-ILC is analyzed. The efficacy of the proposed approach in online control applications is demonstrated through experimental implementation of a recently-developed FD-ILC technique [35] to high-speed precision trajectory tracking on a nanopositioning system.

In Chapter 6, this dissertation is concluded.

Chapter 2

Multi-Objective Optimal Trajectory Design and Tracking with Non-periodic Tracking-Transition Switching for Non-minimum Phase Linear Systems

Abstract

In this chapter, the problem of trajectory design and tracking of non-periodic tracking-transition switching for non-minimum phase linear systems is considered. Such a problem exists in various applications where the output trajectory consists of application-dependent tracking sessions and to-be-designed transition sessions. The challenge arises when multiple control objectives are considered, including the smooth transition from one output tracking session to the next one without large oscillations during the transition, smooth tracking-transition switching without inducing pre- and/or post- switching oscillations, input energy minimization without saturation under amplitude constraint, and furthermore, minimization of the overall transition time. The proposed approach extends the previous work that attained smooth output transition and smooth tracking-transition switching to further achieve amplitude-constrained input-energy minimization and transition time minimization. The constrained input optimization problem is converted to an unconstrained input minimization problem. The optimal output and input are obtained by using an improved conjugate gradient method. The total transition time is further minimized via one dimensional search. The proposed approach is illustrated through a simulation example in probe-based nanomanipulation utilizing a piezoelectric actuator.

2.1 Introduction

The problem of optimal trajectory design and tracking with non-periodic tracking-transition switching under input and transition-time constraints is considered for non-minimum phase linear systems. Such a trajectory design and tracking problem arise in applications such as nanomanipulation [2], robotic operation [4] and hard-disk control [5], where the entire trajectory consists of the tracking portions and the to-be-designed transition portions. The output-transition related control problem has attracted a great attention due to its importance in a wide range of applications. For example, conventionally the optimal state transition (OST) method (e.g., [36] has been applied. The solution, however, not only tends to be non-optimal (for output tracking), but also fails to account for post-transition needs such as maintaining the output at a constant, or tracking a desired trajectory. Although maintaining a constant output can be achieved via the input shaping technique [37], the choice of boundary condition is ad hoc and tends to be non-optimal, and the issue of multiple transitions and/or post-transition tracking are still not considered. The optimal boundary states of the transition can be obtained through the optimal output transition (OOT) technique [6], which has been further extended to solve the output tracking problem with periodic tracking-transition switching [7]. Although the input energy is minimized in the OOT approach, the optimal transition trajectory obtained can be highly oscillatory for systems of lightly-damped dynamics [8], and this method is not suitable to output tracking with non-periodic tracking-transition switching. Although these two issues have been addressed in the recently developed inversion-based optimal output transition technique [9, 10], all these approaches cannot guarantee that the input amplitude is not saturated while the input energy is minimized, and the transition time is small when the above smooth-output, minimal-input requirements are satisfied. Therefore, there exists a need to develop techniques to further account for the input and time constraints in output tracking with non-periodic tracking-transition switching.

Challenge exists in accounting for the input and transition time constraints in non-periodic tracking-transition switching for non-minimum phase systems. These two objectives, transition output and switching smoothness, and input energy minimization under amplitude constraints being coupled together—particularly for non-minimum phase systems [38, 39], impose a constrained multi-objective optimization problem challenging to solve directly due to the strong coupling and non-guaranteed convexity. Minimizing the input energy under amplitude constraint is crucial as input saturation leads to not only tracking performance deterioration, but also slowed response and poor robustness [40]. Moreover, it is highly desirable that the transition time can be further minimized when the above output and input requirements are satisfied.

The main contribution of this chapter is the development of an approach that achieves the optimal transition output trajectory design and tracking under the input and time constraints in the non-periodic tracking-transition switching with preview for non-minimum phase linear systems. The proposed approach extends the technique developed in [9] that attains an optimal smooth output tracking in non-periodic transition switching. Particularly, the input-energy minimization under amplitude constraint is obtained by optimizing the design variables on which the optimal output transition trajectory depends. We first show that such a constrained minimization can be converted into an unconstrained minimization problem. Then, the optimal design variables are obtained based on an improved conjugate gradient technique [33] that guarantees the global optimization without convexity requirement for the underline cost function (instead, a mild Lipschitz condition is needed). Then, the transition time is further minimized for the obtained optimal output transition trajectory. The proposed approach is illustrated by a simulation example of nanomanipulation using a piezoelectric actuator.

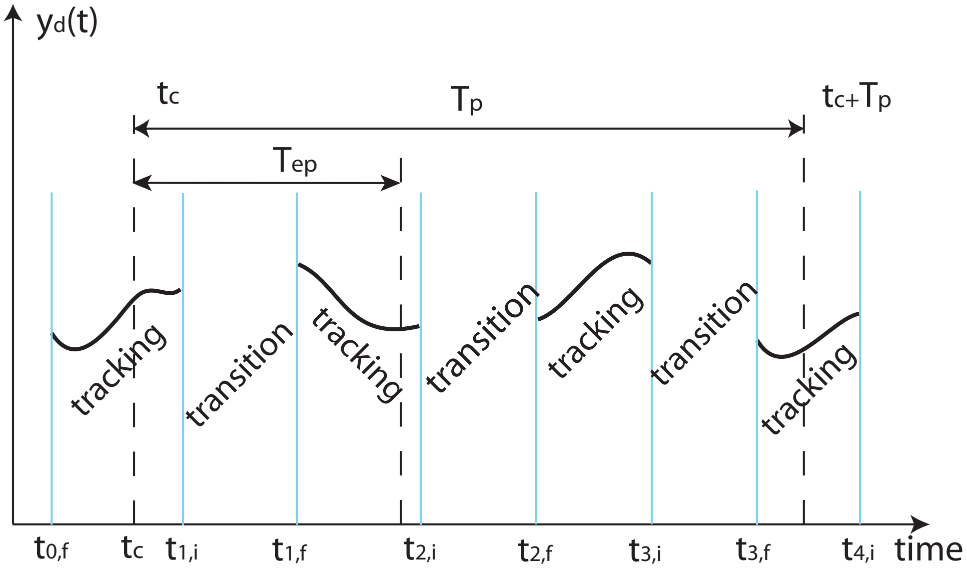

Figure 2.1: The desired output trajectory including tracking and transition sections, where T_p is the total preview time, and T_{ep} is the effective preview time.

2.2 Problem Formulation

Consider the following square linear time invariant system:

$$\dot{x} = Ax + Bu, \quad y = Cx, \tag{2.1}$$

where $x(\cdot) \in \Re^n$ and $u(\cdot),\ y(\cdot) \in \Re^p$. The square system is considered below for ease of presentation–the proposed approach below can be extended to right invertible systems [41, 42] with minor changes.

Assumption 1. *System (2.1) is controllable, observable, and hyperbolic, with a well defined relative degree* $r = [r_1,\ r_2, \ldots,\ r_p]^\mathsf{T}$.

The above hyperbolic assumption (no zeros on the imaginary axis) of the system, although might be alleviated, for example, via minor perturbation to the internal dynamics [43], or equivalently, the desired trajectory [10], is employed here for ease of presentation. We consider output tracking with a finite preview involving non-periodic tracking-transition switching, where the tracking portions of the output trajectory are given, while the transition portions of the trajectory are not specified and need to be designed. As shown in Fig. 1, the entire desired trajectory can be partitioned as

$$y_d(\cdot) = (\cup_q y_{q,dtr}(\cdot)) \cup (\cup_m y_{m,dtn}(\cdot)), \ m, \ q \in \mathbb{N} \tag{2.2}$$

where $y_{q,dtr}(\cdot)$ is the desired output trajectory in the q^{th} tracking session, $y_{m,dtn}(\cdot)$ is the to-be-designed output trajectory in the m^{th} transition session, and $\cup$ denotes union. We define the time interval for the q^{th} tracking trajectory, I_q, and the time interval for the m^{th} transition trajectory, S_m, to be open and closed, respectively, i.e.,

$$I_q = (t_{q,f}, t_{q+1,i}), \quad S_m = [t_{m,i}, t_{m,f}]. \tag{2.3}$$

Correspondingly, the control input could be given as

$$u_{inv}(\cdot) = (\cup_q u_{q,tr}(\cdot)) \cup (\cup_m u_{m,tn}(\cdot)), \tag{2.4}$$

where $u_{q,tr}(\cdot)$ and $u_{m,tn}(\cdot)$ are, respectively, the control inputs for the tracking and transition intervals.

Assumption 2. *During any given q^{th} tracking session, the desired tracking trajectories $y_{q,dtr}(\cdot)$ of the k^{th} output channel are differentiable at least to the r_k^{th} order for all $t \in I_q$, and can be described by a finite-order polynomial of time t.*

The finite-order polynomial description of the desired output in Assumption 2, from practical viewpoint, does not further constraint the output. By the development of the preview-based stable-inversion [39, 44], we define the effective preview time T_{ep} utilized to calculate the inverse input for tracking as the part of preview time T_p so that the tracking error is negligible for the given tracking precision. Moreover, we assume that

Assumption 3. *At any given time instant t_c, the available preview time T_p is at least twice of the effective preview time T_{ep}, i.e., $T_p \geq 2T_{ep}$.*

The above assumption is reasonable as the amount of preview time can be adjusted, for example, through choosing the time scale of the desired trajectory and the transition time. Moreover, a long enough preview time also ensures that the minimization of the input energy under amplitude constraint (the Objective $\mathcal{O}_3$ below) is well-defined as the inputs for the tracking section and the transition section, due to the pre-actuation and post-actuation time [43], are coupled together. We realize that the preview time, although might be substantially reduced and/or avoided through the design of the desired trajectory in set-point regulation [45], is needed in general output tracking where the desired trajectory is given.

Under the above Assumptions 1-3, we set out to design the optimal transition output trajectory and obtain the corresponding input such that:

$\mathcal{O}_1$: The entire output desired trajectory $y_d(\cdot)$ is smooth up to r^{th} (the relative degree of the system) order derivative, i.e., the r^{th} order derivative of the desired output trajectory is continuous during the entire tracking course.

$\mathcal{O}_2$: During each transition session within the previewed time, the energy of the desired output along with its derivatives is minimized,

$$\min_{y_{dtn}(\cdot)}(E_{o,m}) = \min_{y_{dtn}(\cdot)} \int_{t_{m,i}}^{t_{m,f}} (H_{\mathbb{Y}}\mathbb{Y}_{dtn}(\tau))^{\mathsf{T}} R_{\gamma}(H_{\mathbb{Y}}\mathbb{Y}_{dtn}(\tau))d\tau \tag{2.5}$$

where $R_\gamma \in \Re^{p\times p}$ is a positive semidefinite matrix, $t_{m,i},\ t_{m,f}$ are the beginning and end time of the m^{th} transition, respectively, $\mathbb{Y}_{dtn}(\cdot)$ is the vector of designed desired output transition trajectory and its derivatives in that transition, i.e.,

$$\mathbb{Y}_{dtn}(t) = [\xi_{1,dtn}(t)^{\mathsf{T}},\ y_{1,dtn}^{(r_1)}(t),\ \xi_{2,dtn}(t)^{\mathsf{T}}, y_{2,dtn}^{(r_2)}(t), \dots,\ \xi_{p,dtn}(t)^{\mathsf{T}},\ y_{p,dtn}^{(r_p)}(t)] \tag{2.6}$$

with

$$\xi_{k,dtn}(t) = \left[y_{k,dtn}(t),\ \dot{y}_{k,dtn}(t), \dots,\ \frac{d^{r_k-1}y_{k,dtn}(t)}{dt^{r_k-1}} \right]^{\mathsf{T}}, \tag{2.7}$$

and $H_{\mathbb{Y}}$ is a given block diagonal matrix,

$$H_{\mathbb{Y}} = \text{diag}([H_{Y,1},\ H_{Y,2}, \dots,\ H_{Y,p}]), \tag{2.8}$$

with $H_{Y,k} \in \Re^{r_k+1}$ vectors comprised of the coefficients of a monic Hurwitz polynomial in increasing order.

Under the condition that Objectives $\mathcal{O}_1$ and $\mathcal{O}_2$ are satisfied, achieve the following Objectives $\mathcal{O}_3$ and $\mathcal{O}_4$:

$\mathcal{O}_3$: At any given time instant $t_c \in I_q$, the energy of the preview-based inverse input (specified later in Sec. 3) within the effective preview time, $u_{inv}(t)$ for $t \in [t_c, t_c+T_{ep}]$, is minimized under the condition that the input amplitude is within the given threshold, i.e.,

$$\min \sum_{j=1}^{p} \int_{t_c}^{t_c+T_{ep}} u_{j,inv}^2(t)dt, \tag{2.9}$$

subject to

$$\|u_{j,inv}(\cdot)\|_{p,\infty} \leq \mathbf{M}_{j,u}, \quad \text{for} \quad j = 1, 2, \ldots, p, \tag{2.10}$$

where $\|u_{j,inv}(\cdot)\|_{p,\infty} \triangleq \sup |u_{j,inv}(\cdot)|$, and $\mathbf{M}_{j,u}$ is the given threshold for the j^{th} input-output channel.

$\mathcal{O}_4$: Furthermore, the total transition period in the previewed time is minimized,

$$\min \sum_{m=1}^{s} T_m, \tag{2.11}$$

where $T_m = t_{m,f} - t_{m,i}$ is the transition time duration for the m^{th} transition section, and s is the total number of transitions in the effective preview time interval $[t_c,\ t_c + T_{ep}]$. Objective $\mathcal{O}_1$ ensures that the corresponding input–by the stable-inversion theory [38]– is continuous, which is preferred over inputs with “jumps” and/or “spikes” in practice. Moreover, as revealed later (Sec. 3) the weight matrix $H_{\mathbb{Y}}$ specified in Objective $\mathcal{O}_2$ serves as the design parameters to achieve Objective $\mathcal{O}_3$.

2.3 Optimal Output Transition with Non-periodic Tracking-Transition Switching

The proposed approach comprises two parts. First, the candidates of the transition trajectories and the corresponding inputs are designed to meet Objectives $\mathcal{O}_1$ and $\mathcal{O}_2$. Second, the optimal desired transition trajectory is sought from these candidates to meet Objectives $\mathcal{O}_3$ and $\mathcal{O}_4$.

2.3.1 Design of Candidate Desired Transition Trajectories for Objectives $\mathcal{O}_1$ and $\mathcal{O}_2$

The candidate desired trajectories that meet Objectives $\mathcal{O}_1$ and $\mathcal{O}_2$ can be readily obtained through the optimal transition trajectory design and tracking (OTTDT) approach proposed in [9], summarized below. Under Assumptions 1 and 2, system (2.1) can be transformed into the following output tracking form,

$$\begin{cases} \dot{\xi}(t) = I_{up}\xi(t) + B_\xi y^{(r)}(t), \\ \dot{\eta_s}(t) = A_{s1}\eta_s(t) + A_{s2}\xi(t) + B_s y^{(r)}(t), \\ \dot{\eta_u}(t) = A_{u1}\eta_u(t) + A_{u2}\xi(t) + B_u y^{(r)}(t), \end{cases} \tag{2.12}$$

where

$$y^{(r)}(t) = [y_1^{(r)}(t),\ y_2^{(r)}(t), \ldots,\ y_p^{(r)}(t)]^\mathsf{T}, \tag{2.13}$$

$$\xi(t) = [\xi_1(t)^\mathsf{T},\ \xi_2(t)^\mathsf{T}, \ldots,\ \xi_p(t)^\mathsf{T}]^\mathsf{T}, \tag{2.14}$$

with $\xi_k(t)$ given by (2.7), and

$$I_{up} = \mathrm{diag}([I_{up,1},\ I_{up,2}, \ldots,\ I_{up,r_p}]), \tag{2.15}$$

with $I_{up,k} = [\mathbf{0},\ I_k;\ 0,\ \mathbf{0}]_{r_k\times r_k}$ and I_k a $(r_k-1)\times(r_k-1)$ identity matrix, such that for any given desired trajectory $y_d(\cdot)$, exact output tracking can be obtained by using the following inverse input

$$u_{inv}(t) = M_\xi\xi(t) + M_s\eta_s(t) + M_u\eta_u(t) + M_r y^{(r)}(t). \tag{2.16}$$

In (2.12), η_s and η_u are the stable and unstable subspaces of the internal dynamics with eigenvalues of A_{s1} and A_{u1} all in the open left and open right half plane, respectively. Readers are referred to [39, 44] for details (e.g., the expressions for matrices $M_\xi,\ M_s,\ M_u,\ M_r,\ B_s,\ B_u$).

By using a state feedback controller H_ξ,

$$y^{(r)}(t) = H_\xi\xi(t) + \gamma(t), \quad \text{with} \tag{2.17}$$

$$H_\xi = \mathrm{diag}([H_{\xi,1},\ H_{\xi,2}, \ldots,\ H_{\xi,p}]), \quad \text{and} \tag{2.18}$$

$$H_{\xi,k} = [-h_{k,1}, -h_{k,2}, \ldots, -h_{k,r_k}], \tag{2.19}$$

the output subdynamics in (2.12) is stabilized as

$$\dot{\xi}(t) = \hat{A}_\xi\xi(t) + B_\xi\gamma(t), \quad \text{where,} \tag{2.20}$$

$$\hat{A}_\xi = \mathrm{diag}([\hat{A}_{\xi,1},\ \hat{A}_{\xi,2}, \ldots, \hat{A}_{\xi,p}]), \quad \text{and} \tag{2.21}$$

$$B_\xi = \mathrm{diag}([B_{\xi,1},\ B_{\xi,2}, \ldots,\ B_{\xi,p}]), \tag{2.22}$$

$$\begin{aligned} \hat{A}_{\xi,k} &= \begin{pmatrix} \mathbf{0} & I_k \\ & H_{\xi,k} \end{pmatrix}_{r_k \times r_k}, \\ B_{\xi,k} &= \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}_{r_k \times 1}. \end{aligned} \tag{2.23}$$

Objectives $\mathcal{O}_1$ and $\mathcal{O}_2$ are satisfied by solving the following optimal state transition problem [36] for the stabilized output subdynamics (2.20), which, by (2.6, 2.17), can be rewritten in the form of (2.5) in Objective $\mathcal{O}_2$,

$$\begin{aligned} &\min_{\gamma(\cdot)} \int_{t_{m,i}}^{t_{m,f}} \gamma^{\mathsf{T}}(t) R_\gamma \gamma(t) dt \\ &= \min_{\gamma(\cdot)} \int_{t_{m,i}}^{t_{m,f}} (-H_\xi \xi(t) + y^{(r)}(t))^{\mathsf{T}} R_\gamma (-H_\xi \xi(t) + y^{(r)}(t)) dt \\ &= \min_{\gamma(\cdot)} \int_{t_{m,i}}^{t_{m,f}} (H_{\mathbb{Y}} \mathbb{Y}_{dtn}(t))^{\mathsf{T}} R_\gamma (H_{\mathbb{Y}} \mathbb{Y}_{dtn}(t)) dt, \end{aligned} \tag{2.24}$$

with $H_{Y,k} = [-H_\xi,\ \mathbf{1}]$. The solution to (2.24) is given as

$$\gamma^*(t) = R_\gamma^{-1} B_\xi^\top e^{\hat{A}_\xi^{\mathsf{T}}(t_{m,f}-t)} \mathcal{G}^{-1}(T_m) \mathcal{L}(t_{m,i}, t_{m,f}), \tag{2.25}$$

where

$$\begin{aligned} \mathcal{G}(T_m) &= \int_0^{T_m} e^{\hat{A}_\xi (T_m - \tau)} B_\xi R_\gamma^{-1} B_\xi^{\mathsf{T}} e^{\hat{A}_\xi^{\mathsf{T}}(T_m-\tau)} d\tau, \\ \mathcal{L}(t_{m,i}, t_{m,f}) &= \xi_{dtn}(t_{m,f}) - e^{\hat{A}_\xi (t_{m,f} - t_{m,i})} \xi_{dtn}(t_{m,i}). \end{aligned} \tag{2.26}$$

With (2.25), the desired output is given by,

$$y_{dtn}^{(r)}(t) = H_\xi \xi_{dtn}(t) + \gamma^*(t), \quad \text{for } t \in \Re^+, \tag{2.27}$$

where $y_{dtn}^{(r)}(\cdot)$ and $\xi_{dtn}(\cdot)$ are as defined in (2.13, 2.14) corresponding to the desired output, respectively.

Note that the solution $\gamma^*(t)$ to the optimization problem of (2.24) becomes trivial, i.e., $\gamma^*(\cdot) = 0$, only when $\mathcal{L}(t_{m,i}, t_{m,f})$ in (2.25) is zero, i.e., only when the final output state (the desired output value at the end of the transition) falls on the initial-condition state curve. Such a case virtually does not exist in practices or can be easily avoided via the design of the stabilized feedback gain H_ξ and a small perturbation to the transition time T_m otherwise. Thus, without loss of generality, we assume

Assumption 4. *The state transition $\mathcal{L}(t_{m,i}, t_{m,f})$ in (2.25) is non-zero, and the optimal solution $\gamma^*(t)$ to the optimization problem in (2.24) is non-zero.*

For the given transition boundary condition and the transition time, the only design variable of the optimal transition input-output is the weighting matrix $H_{\mathbb{Y}}$ (see (2.12, 2.16, 2.25-2.27)). Next, we achieve Objective $\mathcal{O}_3$ through the design of the weighting matrix $H_{\mathbb{Y}}$.

2.3.2 Objective $\mathcal{O}_3$: Optimal Transition Trajectory with Minimal Input Energy under Amplitude Constraint

The optimal transition trajectory that meets Objective $\mathcal{O}_3$ is obtained in two steps: First, the constrained optimization problem of $\mathcal{O}_3$ is transformed to an unconstrained optimization for ease of solving. Second, the solution is obtained through an improved conjugate gradient method.

An Equivalent Unconstrained Input Optimization Problem

To convert the amplitude-constrained input energy minimization problem (2.9, 2.10) to a practically equivalent unconstrained minimization problem, we note that for the k^{th} output with $k = 1,\ 2, \cdots,\ p$, the state-feedback gain $H_{\xi,k}$ is nothing but the coefficients of the characteristic polynomial of the stabilized output subsystem subsystem (see (2.19)), i.e.,

$$\begin{aligned}\mathcal{P}_k(s) &= s^{r_k} + h_{k,r_k} s^{r_k-1} + h_{k,r_k-1} s^{r_k-2} + \cdots + h_{k,1} \\ &= (s-\lambda_{k,1})(s-\lambda_{k,2})\ldots(s-\lambda_{k,r_k})\end{aligned} \tag{2.28}$$

where $\lambda_{k,1},\ \lambda_{k,2},\ \ldots,\ \lambda_{k,r_k}$ are the eigenvalues of $\hat{A}_{\xi,k}$. Thus, the design of the state-feedback gain $H_{\xi,k}$ for the k^{th} output ($k = 1, 2, \cdots, p$) is equivalent to designing the eigenvalues of the stabilized output subsystem state matrix $\hat{A}_{\xi,k}$, $\lambda_{k,1},\ \lambda_{k,2},\ \ldots,\ \lambda_{k,r_k}$. Since oscillations of the desired output can be completely eliminated when all the eigenvalues $\lambda_{k,1},\ \lambda_{k,2},\ \ldots,\ \lambda_{k,r_k}$ have negative real, we consider, without loss of generality, the eigenvalues in designing the output trajectory for the m^{th} transition section from a set of negative real finite numbers,

$$\mathcal{V}_{\lambda,m} \triangleq [\lambda_{1,1}^m,\ \lambda_{1,2}^m,\ldots,\ \lambda_{1,r_1}^m,\ \lambda_{2,1}^m,\ldots,\ \lambda_{p,1}^m,\ldots,\ \lambda_{p,r_p}^m], \quad m = 1, 2, \cdots, s, \quad \text{and} \tag{2.29}$$

$$\mathcal{V}_{\lambda,m} \in D_m \triangleq \{(S_1,\ S_2,\ldots,S_{N_\lambda}) : S_v \in \Re \text{ and } S_v < 0, \text{ for all } v = 1,\ 2,\ldots,\ N_\lambda\}, \tag{2.30}$$

where $N_\lambda = \sum_{k=1}^{p} r_k$. It is challenging, however, to directly solve the amplitude-constrained input energy minimization problem in Objective $\mathcal{O}_3$, as the given threshold must be satisfied at every time instant over the entire transition period. Thus, the following unconstrained optimization problem is solved instead:

$$\min_{\mathcal{V}_\lambda} \mathcal{J}(\mathcal{V}_\lambda) = \min_{\mathcal{V}_\lambda} \sum_{j=1}^{p} \int_{t_c}^{t_c+T_{ep}} u_{j,inv}^2(t) + \mathbf{M}_{j,u}^2 \exp\left(\frac{c(u_{j,inv}^2(t) - \mathbf{M}_{j,u}^2)}{\mathbf{M}_{j,u}^2} \right) dt, \tag{2.31}$$

where $c > 0$ is a large enough constant (specified later), and

$$\mathcal{V}_\lambda = [\mathcal{V}_{\lambda,1},\ \mathcal{V}_{\lambda,2},\ldots,\ \mathcal{V}_{\lambda,s}] \in D \triangleq D_1 \times D_2 \ldots \times D_s \tag{2.32}$$

is the vector of eigenvalues used in designing the output trajectories of all the transition sections during the entire effective preview time at time instant t_c, $[t_c,\ t_c + T_{ep}]$, and D is the the corresponding set of eigenvalue vectors, $\mathcal{V}_\lambda$, with $D_i \times D_j = \{(x,\ y) : x \in D_i,\ y \in D_j\}$, and $D_1 \times D_2 \cdots \times D_s$ defined similarly.

We next show that in practice, the constrained minimization problem $\mathcal{O}_3$ is virtually equivalent to the above unconstrained minimization problem, provided that the inverse input does not contain any flat period at the threshold (i.e., the corresponding input is not stationary at the threshold in any finite-length time interval). The following Lemma 5 shows that such a condition on the inverse input does hold.

Lemma 5. *Let Assumptions 1 to 4 be satisfied, and let the desired output trajectory within the effective preview time, $y_d(t)$ for $t \in [t_c,\ t_c + T_{ep}]$, be given by (2.2), with the portions in the transition periods satisfying Objectives $\mathcal{O}_1$ and $\mathcal{O}_2$, then for any positive finite $\mathbf{M} < \infty$ the inverse input $u_{inv}(t)$ in (2.16) is not stationary during any finite time interval, i.e., $m(\{t : |u_{j,inv}(\mathcal{V}_\lambda, t)| = \mathbf{M}, \forall \mathcal{V}_\lambda \in D\}) = 0$ ($m(A)$ is the Lebesgue measure of set A).*

Proof. We proceed by contradiction. First we show the inverse input (2.16) can be represented as a finite summation of exponents and polynomials of time t. Consider the case that the system does not have repeated zeros. By Assumption 2, $y(t) = \sum_{i=0}^{l} a_i t^i, \quad t \in \Re^+,\ l \in \mathbb{N}$, where $a_i \in \Re$ and a_is are not all zero, thus the j^{th} component of the stable state $\eta_s(t)$, $\eta_{s,j}(t)$, can be represented as

$$\begin{aligned}
\eta_{s,j}(t) &= e^{\lambda_j t}\eta_{s,j}(t_0) + \int_{t_0}^{t} e^{\lambda_j (t-\tau)} \left(\sum_{p=1}^{r} A_{s2,\ jp} y^{(p)}(\tau) + B_{s,j} y^{(r+1)}(\tau) \right) d\tau \\
&= e^{\lambda_j t}\eta_{s,j}(t_0) + \int_{t_0}^{t} e^{\lambda_j (t-\tau)} \left(\sum_{p=1}^{r} A_{s2,\ jp} \left(\sum_{i=0}^{l} a_i \tau^i \right)^{(p)} + B_{s,j} \left(\sum_{i=0}^{l} a_i \tau^i \right)^{(r+1)} \right) d\tau \\
&= e^{\lambda_j t}\eta_{s,j}(t_0) + \sum_{i=0}^{l} \int_{t_0}^{t} e^{\lambda_j (t-\tau)} a_i' \tau^i d\tau, \\
&= e^{\lambda_j t}(\eta_{s,j}(t_0) + b_j) + \sum_{i=0}^{l} F_{ji} t^i,
\end{aligned} \tag{2.33}$$

where $i \in \mathbb{N},\ j \in \mathbb{N},\ p \in \mathbb{N},\ \lambda_j \in \mathbb{C}$ are the minimum-phase zeros of the system, $b_j \in \Re$ and $F = (F_{ji}) \in \Re^{n_s \times (l+1)}$ is a coefficient matrix with the j^{th} row and i^{th} column element F_{ji} depend on a_i and λ_j, with n_s the number of stable states.

Similarly, when there are no repeated non-minimum phase zeros, the unstable dynamics $\eta_u(t)$ can also be represented as a summation of exponents and polynomials as in (2.33). Thus by (2.16),

$$u_{k,inv}(t) = \sum_{j=1}^{n-r} c_{kj} e^{\lambda_j t} + \sum_{i=0}^{l} d_{ki} t^i, \tag{2.34}$$

where

$$c_{kj} = G_{kj}(\eta_j(t_0) + b_j) \tag{2.35}$$

with $\eta = [\eta_s^\intercal\ \eta_u^\intercal]^\intercal$, and $G = [M_s,\ M_u]$ of full row rank, and d_{ki} depend on $M_\xi,\ M_r,\ A_s,\ A_u,\ y(t)$. As the set $\Omega_f = \{f_i(t) = t^{k_i} e^{\lambda_i t} : k_i \in \mathbb{Z}^+,\ \lambda_i \in \mathbb{C},\ i = 1,\ 2, \cdots, N_i,\ N_i \in \mathbb{N}$, and $N_i < \infty\}$ is linearly independent for distinct k_is or distinct λ_is in any finite-length time interval [46], $u_{k,inv}(t)$ equaling to a constant in a finite time interval implies that $c_{kj} = 0$ for each $j = 1,\ 2, \ldots,\ n-r$ and $d_{ki} = 0$ for each $i = 1,\ 2, \ldots,\ l$, which by (2.35), in turn, forces $G_k j = 0$ for each $j = 1,\ 2, \ldots,\ n-r$—as $\eta_j(t_0) + b_j$, being initial condition dependent, cannot be always zero. As a result, G does not have a full row rank—a contradiction.

For the case where the system has repeated zeros, there are repeated poles in the inverse system (2.12), and the term $e^{A_{u1}t}$ in computing $\eta_u(t)$ or the term $e^{A_{s1}t}$ in computing $\eta_s(t)$ generates extraneous terms $t^k e^{\lambda t}$ in addition to terms $e^{\lambda t}$. By replacing those terms of $e^{\lambda t}$ in (2.34) with the terms of $t^k e^{\lambda t}$, for finite $k \in \mathbb{Z}^+$, the same argument above can be applied to show that the same conclusion holds.

During the transition sections, $\gamma(t)$ in (2.25), thereby $\xi(t)$, $y^{(r)}(t)$ in (2.20, 2.27), can be expressed as a summation of finite elements in the set $\Omega_f = \{t^{k_i} e^{\lambda_i t}\}$, then the same procedure can be applied to represent the inverse input as a finite sum of exponents and polynomials of t. Although the expressions of $\xi(t)$, $y^{(r)}(t)$ involve exponents of t, rather than only polynomial as for the tracking trajectory, the set of exponents corresponding to each internal state, i.e., each component of $\eta_s(t)$ and $\eta_u(t)$, are still distinct to each other, thereby the same argument above applies. This completes the proof. □

Lemma 6. *Let the finite constant $M > 0$ be given, and the function $f(\boldsymbol{x}, t) \in L_\infty$: $f : D \times [a, b] \to \Re$, with $0 \le a < b < \infty$, be continuous in $\boldsymbol{x}$ and piecewise continuous in t, and the Lebesgue measure of the set $V = \{t : |f(\boldsymbol{x}, \cdot)| = \mathbf{M}, \forall \boldsymbol{x} \in D\}$, $m(V) = 0$, then for any given $\epsilon > 0$, there exists a finite constant $c > 0$, such that the solution $\boldsymbol{x}^*$ that minimizes the cost function,*

$$J(\boldsymbol{x}) = \int_a^b f^2(\boldsymbol{x}, t) + \mathbf{M}^2 \exp\left(c\left(\frac{f^2(\boldsymbol{x}, t)}{\mathbf{M}^2} - 1\right)\right) dt$$

has the following properties:

- $\|f(\boldsymbol{x}^*, \cdot)\|_\infty \le \mathbf{M}$ *, where $\|f(\boldsymbol{x}, \cdot)\|_\infty$ is the infinity norm w.r.t time t.*
- *For any $\boldsymbol{x}$ satisfying $\|f(\boldsymbol{x}, \cdot)\|_\infty \le \mathbf{M}$,*

$$J(\boldsymbol{x}^*) \le \int_a^b f^2(\boldsymbol{x}, t) dt + \epsilon.$$

Proof. Let $B_1 = \{t : |f(\mathbf{x}, t)| < \mathbf{M}\} \cap [a, b]$, $B_2 = \{t : |f(\mathbf{x}, t)| = \mathbf{M}\} \cap [a, b]$, and $B_3 = \{t : |f(\mathbf{x}, t)| > \mathbf{M}\} \cap [a, b]$. Since $m(B_2) = 0$, then the cost function in the Lebesgue integral form can be represented as

$$\begin{aligned} J(\mathbf{x}) &= \int_{[a,b]} f^2 + \mathbf{M}^2 \left(\int_{B_1} h + \int_{B_2} h + \int_{B_3} h \right) \\ &= \int_{[a,b]} f^2 + \mathbf{M}^2 \left(\int_{B_1} h + \int_{B_3} h \right), \end{aligned}$$

where $h(\mathbf{x}, t) \triangleq \exp(c(\frac{f^2(\mathbf{x},t)}{\mathbf{M}^2} - 1))$.

Let $D_1 = \{\mathbf{x} : \|f(\mathbf{x}, \cdot)\|_\infty \leq \mathbf{M}\} \cap D$ and $D_2 = \{\mathbf{x} : \|f(\mathbf{x}, \cdot)\|_\infty > \mathbf{M}\} \cap D$. Then for any $\mathbf{x} \in D_1$, $m(B_1) = b - a,\ m(B_3) = 0$, and

$$J(\mathbf{x} \in D_1) = \int_{[a,b]} f^2 + \mathbf{M}^2 \int_{B_1} h,$$

As for all $\mathbf{x} \in D_1,\ h(\cdot, \cdot) \to 0$ when $c \to \infty$, there exists a finite C_1 such that $h(\cdot, \cdot) \leq \frac{\epsilon}{\mathbf{M}^2(b-a)}$ for any given $\epsilon > 0$, and any $c > C_1$. Hence for any $\mathbf{x} \in D_1$,

$$J(\mathbf{x} \in D_1) \leq \int_{[a,b]} f^2 + \epsilon \leq \mathbf{M}^2(b-a) + \epsilon.$$

Similarly, for any $\mathbf{x} \in D_2$, $m(B_1) > 0,\ m(B_3) > 0$, and $h(\cdot, \cdot) \to \infty$ when $c \to \infty$, there exists a finite C_2 such that $h(\cdot, \cdot) > (b-a)/m(B_3) + \epsilon$ for all $c > C_2$. Thereby for any $\mathbf{x} \in D_2$,

$$J(\mathbf{x} \in D_2) > \int_{[a,b]} f^2 + \mathbf{M}^2(b-a) + \epsilon.$$

By setting $c = \max(C_1, C_2)$, we have $J(\mathbf{x} \in D_1) < J(\mathbf{x} \in D_2)$, and the two claimed properties follow. This completes the proof. □

The above Lemmas 5 and Lemma 6 imply that the original optimization problem in (2.9, 2.10) can be solved with arbitrary precision as the unconstrained optimization problem (2.31). The constant c can be chosen a priori as both the threshold value $M_{j,u}$ and the transition time period are known. Moreover, note that the threshold value needs to be appropriately chosen: for the j^{th} input, the threshold value $M_{j,u} \geq \sum_{i=1}^{p} \|G_{ji}^{-1}(\cdot)\|_\infty \|y_{i,tr}(t)\|_\infty$, with G_{ji} the linear operator from the j^{th} input to the i^{th} output, and $y_{i,tr}(t)$ the i^{th} channel output trajectory. Such a condition generally holds and can be checked in practice.

Optimal Solution based on the Improved Conjugate Gradient Method

We propose an approach based on the improved conjugated gradient method [33] to obtain the optimal input that meets Objective $\mathcal{O}_3$ through minimization of the cost function in (2.31). As the inverse input $u_{inv}(\cdot,\cdot)$ implicitly depends on the eigenvalues of $\hat{A}_\xi$, we utilize the concept of variation calculus to obtain the gradient of $u_{inv}(\cdot,\cdot)$ w.r.t. a given eigenvalue $\lambda_{k,l}^m$. The gradient of the cost function (2.31) with respect to eigenvalue $\lambda_{k,l}^m$ is

$$\frac{\partial \mathcal{J}(\mathcal{V}_\lambda)}{\partial \lambda_{k,l}^m} = 2\sum_{j=1}^{p}\int_{t_c}^{t_c+T_{ep}}\left(c\exp\left(\frac{c(u_{j,inv}^2-\mathbf{M}_{j,u}^2)}{\mathbf{M}_{j,u}^2}\right)+1\right)u_{j,inv}\frac{\partial u_{j,inv}}{\partial \lambda_{k,l}^m}dt. \tag{2.36}$$

To calculate the derivative of $u_{j,inv}$ with respect to $\lambda_{k,l}^m$, consider the perturbed dynamics due to the variation of the eigenvalues of $\hat{A}_\xi$, δH_ξ, during the m^{th} transition section. By (2.12, 2.16),

$$\delta\dot{\xi}(t) = \hat{A}_\xi\delta\xi(t) + B_\xi\delta H_\xi\xi(t) + B_\xi\delta\gamma(t). \tag{2.37}$$

Similarly, the corresponding perturbed dynamics of the internal dynamics can be obtained as

$$\begin{cases} \delta\dot{\eta_s}(t) = A_{s1}\delta\eta_s(t) + A_{s2}\delta\xi(t) + B_s\delta y^{(r)}(t), \\ \delta\dot{\eta_u}(t) = A_{u1}\delta\eta_u(t) + A_{u2}\delta\xi(t) + B_u\delta y^{(r)}(t), \end{cases} \tag{2.38}$$

and the corresponding variation of the inverse input becomes

$$\delta u_{inv}(t) = M_\xi\delta\xi(t) + M_s\delta\eta_s(t) + M_u\delta\eta_u(t) + M_r\delta y^{(r)}(t). \tag{2.39}$$

The initial conditions for the above variational dynamics (2.37, 2.38) are zero, then the variation of the inverse input can be readily represented as a linear function of the variation of the output dynamics, δH_ξ, and that of the external input, $\delta\gamma(t)$ (both caused by the eigenvalue variation). The variation of the output dynamics during the transition period can be obtained as

$$\begin{aligned} \delta\xi(t) &= \int_{t_c}^{t} e^{\hat{A}_\xi(t-\tau)}B_\xi(\delta H_\xi\xi(\tau)+\delta\gamma(\tau))d\tau, \\ \delta y^{(r)}(t) &= \delta H_\xi\xi(t) + H_\xi\delta\xi(t) + B_\xi\delta\gamma(t), \end{aligned} \tag{2.40}$$

and $\delta\xi(t) = 0,\ \delta y^{(r)}(t) = 0$ during the tracking sections. The variation of the internal dynamics can be readily obtained as

$$\begin{aligned} \delta\eta_s(t) &= \int_{t_c}^{t} e^{A_{s1}(t-\tau)}[A_{s2}\delta\xi(\tau) + B_s\delta y^{(r)}(\tau)]d\tau, \\ \delta\eta_u(t) &= e^{-A_{u1}(t_c+T_{ep})}\delta\eta_u(T_{ep}) - \int_{t}^{t_c+T_{ep}} e^{-A_{u1}(\tau-t)}[A_{u2}\delta\xi(\tau) + B_u\delta y^{(r)}(\tau)]d\tau. \end{aligned} \tag{2.41}$$

The variation of the internal dynamics contains a stable and an unstable portion, where the variation of the portion $\delta\eta_u(t)$ is calculated with the effective preview time T_{ep}. The unknown future boundary condition, $\delta\eta_u(T_{ep})$, is set to zero with negligible truncation-caused error (see Assumption 4).

Since

$$\delta u_{inv}(t) = \sum_{k=1}^{p}\sum_{l=1}^{r_k} \frac{\partial u_{inv}}{\partial\lambda_{k,l}^m}\delta\lambda_{k,l}^m, \tag{2.42}$$

$$\delta H_\xi(t) = \sum_{k=1}^{p}\sum_{l=1}^{r_k} \frac{\partial H_\xi}{\partial\lambda_{k,l}^m}\delta\lambda_{k,l}^m, \tag{2.43}$$

$$\delta\gamma(t)(t) = \sum_{k=1}^{p}\sum_{l=1}^{r_k} \frac{\partial \gamma}{\partial\lambda_{k,l}^m}\delta\lambda_{k,l}^m, \tag{2.44}$$

the gradient of the inverse input w.r.t. any given eigenvalue $\lambda_{k,l}^m$, $\partial u_{inv}/\partial\lambda_{k,l}^m$, could be obtained by combining (2.39-2.41) with (2.42-2.44),

$$\frac{\partial u_{inv}(t)}{\partial\lambda_{k,l}^m} = M_\xi\frac{\partial\xi(t)}{\partial\lambda_{k,l}^m} + M_s\frac{\partial\eta_s(t)}{\partial\lambda_{k,l}^m} + M_u\frac{\partial\eta_u(t)}{\partial\lambda_{k,l}^m} + M_r\frac{\partial y^{(r)}(t)}{\partial\lambda_{k,l}^m}, \tag{2.45}$$

where

$$\frac{\partial\xi(t)}{\partial\lambda_{k,l}^m} = \begin{cases} \displaystyle\int_{t_{m,i}}^{t} e^{\hat{A}_\xi(t-\tau)}B_\xi\left[\frac{\partial H_\xi}{\partial\lambda_{k,l}^m}\xi(\tau) + \frac{\partial\gamma(\tau)}{\partial\lambda_{k,l}^m}\right]d\tau, & \text{for } t\in S_m = [t_{m,i},\ t_{m,f}], \\ 0, & \text{otherwise,} \end{cases} \tag{2.46}$$

$$\frac{\partial y^{(r)}(t)}{\partial\lambda_{k,l}^m} = \begin{cases} \dfrac{\partial H_\xi}{\partial\lambda_{k,l}^m}\xi(t) + H_\xi\dfrac{\partial\xi(t)}{\partial\lambda_{k,l}^m} + B_\xi\dfrac{\partial\gamma(t)}{\partial\lambda_{k,l}^m}, & \text{for } t\in S_m = [t_{m,i},\ t_{m,f}], \\ 0, & \text{otherwise,} \end{cases} \tag{2.47}$$

$$\begin{aligned}
\frac{\partial \eta_s(t)}{\partial \lambda^m_{k,l}} &= \int_{t_c}^{t} e^{A_{s1}(t-\tau)} \left[A_{s2}\frac{\partial \xi(\tau)}{\partial \lambda^m_{k,l}} + B_s \frac{\partial y^{(r)}(\tau)}{\partial \lambda^m_{k,l}} \right] d\tau, \\
\frac{\partial \eta_u(t)}{\partial \lambda^m_{k,l}} &= -\int_{t}^{t_c+T_{ep}} e^{-A_{u1}(\tau-t)} \left[A_{u2}\frac{\partial \xi(\tau)}{\partial \lambda^m_{k,l}} + B_u \frac{\partial y^{(r)}(\tau)}{\partial \lambda^m_{k,l}} \right] d\tau, \quad \text{with} \\
\frac{\partial H_\xi}{\partial \lambda^m_{k,l}} &= \mathrm{diag}\left(\left[0, 0, \ldots, \frac{\partial H_{\xi,k}}{\partial \lambda^m_{k,l}}, 0, \ldots, 0 \right] \right), \\
\frac{\partial H_{\xi,k}}{\partial \lambda^m_{k,l}} &= \left[-\frac{\partial h_{k,1}}{\partial \lambda^m_{k,l}}, -\frac{\partial h_{k,2}}{\partial \lambda^m_{k,l}}, \ldots, -\frac{\partial h_{k,r_k}}{\partial \lambda^m_{k,l}} \right],
\end{aligned} \tag{2.48}$$

Eqs. (2.45-2.48) show that to compute the derivative of the inverse input w.r.t. a given eigenvalue, the only unknown term is the derivative of the inverse input to the given eigenvalue, $\partial\gamma(t)/\partial\lambda^m_{k,l}$. By Eq. (2.25), $\partial\gamma(t)/\partial\lambda^m_{k,l}$ is given by

$$\begin{aligned}
\frac{\partial \gamma(t)}{\partial \lambda^m_{k,l}} &= R_\gamma^{-1} B_\xi^\top \left[\frac{\partial e^{\hat{A}_\xi^\top (t_{m,f}-t)}}{\partial \lambda^m_{k,l}} \mathcal{G}^{-1}(T_m) + e^{\hat{A}_\xi^\top (t_{m,f}-t)} \frac{\partial \mathcal{G}^{-1}(T_m)}{\partial \lambda^m_{k,l}} \right] \mathcal{L}(t_{m,i}, t_{m,f}) \\
&\quad + R_\gamma^{-1} B_\xi^\top e^{\hat{A}_\xi^\top (t_{m,f}-t)} \mathcal{G}^{-1}(T_m) \times \left[-\frac{\partial e^{\hat{A}_\xi (t_{m,f}-t_{m,i})}}{\partial \lambda^m_{k,l}} \xi_{dtr}(t_{m,i}) \right], \\
&\quad \text{with} \\
\frac{\partial \mathcal{G}^{-1}(T_m)}{\partial \lambda^m_{k,l}} &= -\mathcal{G}^{-1}(T_m) \frac{\partial \mathcal{G}(T_m)}{\partial \lambda^m_{k,l}} \mathcal{G}^{-1}(T_m), \\
\frac{\partial \mathcal{G}(T_m)}{\partial \lambda^m_{k,l}} &= \int_0^{T_m} \frac{\partial e^{\hat{A}_\xi (T_m-\tau)}}{\partial \lambda^m_{k,l}} B_\xi R_\gamma^{-1} B_\xi^\top e^{\hat{A}_\xi^\top (T_m-\tau)} d\tau \\
&\quad + \int_0^{T_m} e^{\hat{A}_\xi (T_m-\tau)} B_\xi R_\gamma^{-1} B_\xi^\top \frac{\partial e^{\hat{A}_\xi^\top (T_m-\tau)}}{\partial \lambda^m_{k,l}} d\tau.
\end{aligned} \tag{2.49}$$

Examining the above (2.49) reveals that finding $\partial u_{inv}/\lambda^m_{k,l}$ amounts to computing the derivative of $e^{\hat{A}_{\xi,k} t}$ w.r.t. $\lambda^m_{k,l}$. To that end, we consider the similarity transform $\hat{A}_{\xi,k} = P_k \Lambda_k P_k^{-1}$, and hence,

$$\frac{\partial e^{\hat{A}_{\xi,k} t}}{\partial \lambda^m_{k,l}} = \frac{\partial P_k}{\partial \lambda^m_{k,l}} e^{\Lambda_k t} P_k^{-1} + P_k \frac{\partial e^{\Lambda_k t}}{\partial \lambda^m_{k,l}} P_k^{-1} + P_k e^{\Lambda_k t} \frac{\partial P_k^{-1}}{\partial \lambda^m_{k,l}}. \tag{2.50}$$

Once the gradient of the inverse input w.r.t. the eigenvalues is obtained, we propose to obtain the optimal vector of eigenvalues that meet Objective $\mathcal{O}_3$ by using the improved conjugate gradient method in [33]: for a given cost function $\mathcal{J}(\mathcal{V}_\lambda)$ and its gradient $\mathbf{g}(\mathcal{V}_\lambda)$, i.e.,

$$\mathbf{g}(\mathcal{V}_\lambda) \triangleq \left[\frac{\partial \mathcal{J}(\mathcal{V}_\lambda)}{\partial \mathcal{V}_{\lambda,1}}, \frac{\partial \mathcal{J}(\mathcal{V}_\lambda)}{\partial \mathcal{V}_{\lambda,2}}, \ldots, \frac{\partial \mathcal{J}(\mathcal{V}_\lambda)}{\partial \mathcal{V}_{\lambda,s}} \right]^\top,$$

with

$$\frac{\partial \mathcal{J}(\mathcal{V}_\lambda)}{\partial \mathcal{V}_{\lambda,m}} = \left[\frac{\partial \mathcal{J}(\mathcal{V}_\lambda)}{\partial \mathcal{V}_{\lambda,m}(1)}, \frac{\partial \mathcal{J}(\mathcal{V}_\lambda)}{\partial \mathcal{V}_{\lambda,m}(2)}, \ldots, \frac{\partial \mathcal{J}(\mathcal{V}_\lambda)}{\partial \mathcal{V}_{\lambda,m}(N)}\right]^{\mathsf{T}}.$$

The optimal eigenvalue vector, $\mathcal{V}_\lambda^*$, is obtained via the following iterative process,

$$\mathcal{V}_\lambda^{k+1} = \mathcal{V}_\lambda^k + \alpha_k \mathbf{d}_k, \tag{2.51}$$

with

$$\mathbf{d}_k = \begin{cases} -\mathbf{g}_k & \text{for } k = 1, \\ -\mathbf{g}_k + \beta_k \mathbf{d}_{k-1} & \text{for } k \geq 2, \end{cases} \tag{2.52}$$

where $\mathbf{g}_k \triangleq \mathbf{g}(\mathcal{V}_\lambda^k)$, α_k is the step length, obtained by one dimensional search such that the following Condition 2 (specified immediately below) is satisfied, and β_k is a scaler chosen as

$$\beta_k = \begin{cases} -\beta_k^{FR} & \text{if } \beta_k^{PR} < -\beta_k^{FR}, \\ \beta_k^{PR} & \text{if } |\beta_k^{PR}| < \beta_k^{FR}, \\ \beta_k^{FR} & \text{if } \beta_k^{PR} > \beta_k^{FR}, \end{cases} \tag{2.53}$$

where

$$\beta_k^{FR} = \|\mathbf{g}_k\|^2 / \|\mathbf{g}_{k-1}\|^2, \text{ and} \tag{2.54}$$

$$\beta_k^{PR} = \langle \mathbf{g}_k,\ \mathbf{g}_k - \mathbf{g}_{k-1}\rangle / \|\mathbf{g}_{k-1}\|^2, \tag{2.55}$$

with $\langle \mathbf{g}_k,\ \mathbf{g}_j\rangle = \mathbf{g}_k^{\mathsf{T}} \cdot\ \mathbf{g}_j$ the inner product of $\mathbf{g}_k$ and $\mathbf{g}_j$.

It has been shown in [33] that the vector of the optimal eigenvalues that reaches the global minimal of $\mathcal{J}^*(\mathcal{V}_\lambda)$, can be obtained via the above iteration process provided that the following two conditions are satisfied:

Condition 1: For any given two sets of eigenvalues, $\mathcal{V}_{\lambda_\iota},\ \mathcal{V}_{\lambda_\kappa} \in D$ (given by (2.29, 2.32)), there exists a positive constant $L > 0$, such that the gradient of the cost function $\mathcal{J}(\mathcal{V}_\lambda)$ in (2.31) satisfies the following Lipschitz condition,

$$\|\mathbf{g}(\mathcal{V}_{\lambda_\iota}) - \mathbf{g}(\mathcal{V}_{\lambda_\kappa})\| \leq L\|\mathcal{V}_{\lambda_\iota} - \mathcal{V}_{\lambda_\kappa}\|, \tag{2.56}$$

for any $\mathcal{V}_{\lambda_\iota},\ \mathcal{V}_{\lambda_\kappa} \in D$.

Condition 2: At each iteration step k, the steplength α_k can be chosen such that either one of the following two conditions is satisfied: (1) the Wolfe condition,

$$\mathcal{J}(\mathcal{V}_\lambda^k + \alpha_k \mathbf{d}_k) \leq \mathcal{J}(\mathcal{V}_\lambda^k) + \sigma_1 \alpha_k \langle \mathbf{g}_k, \mathbf{d}_k \rangle, \tag{2.57}$$

$$\langle \mathbf{g}(\mathcal{V}_\lambda^k + \alpha_k \mathbf{d}_k), \mathbf{d}_k \rangle \geq \sigma_2 \langle \mathbf{g}_k, \mathbf{d}_k \rangle, \tag{2.58}$$

for any $0 < \sigma_1 < \sigma_2 < 0.5$;
or (2) the ideal line search condition,

$$\mathcal{J}(\mathcal{V}_\lambda^k + \alpha_k \mathbf{d}_k) \leq \mathcal{J}(\mathcal{V}_\lambda^k + \hat{\alpha}_k \mathbf{d}_k), \tag{2.59}$$

where $\hat{\alpha}_k$ is the smallest positive stationary point of the function $\phi_k(\alpha) \triangleq \mathcal{J}(\mathcal{V}_\lambda^k + \alpha \mathbf{d}_k)$, i.e., the smallest positive α satisfying $d\phi_k(\alpha)/d\alpha = 0$.
Note that the improved conjugate gradient method does not require that the cost function to be convex, [33], as usually needed in other optimization methods. Next, we show that the cost function $\mathcal{J}(\mathcal{V}_\lambda)$ satisfies the above two conditions. First, the existence of α_k that satisfies the Wolfe Condition in (2.57, 2.58) or the ideal line search condition in (2.59), by [33, 47], is guaranteed by the continuously differentiability and boundedness of the cost function $\mathcal{J}(\mathcal{V}_\lambda)$, and the existence of the local and global minima of the cost function $\mathcal{J}(\mathcal{V}_\lambda)$, respectively–Condition 2–can always hold by adjusting the step length α_k during the search process. As α_k is the only parameter to be determined in Condition 2, one dimensional search method, such as the bi-section method or the golden-section method [48], can be utilized to find such an α_k. The following Lemma shows that Condition 1 also holds.

Lemma 7. *For the inverse input $u_{inv}(t)$ given by (2.16) and the the desired transition trajectory given by (2.27), the gradient of the cost function $\mathcal{J}(\mathcal{V}_\lambda)$ w.r.t $\mathcal{V}_\lambda$ satisfies Condition 1 for any $\mathcal{V}_\lambda \in D$.*

Proof. The proof amounts to showing that the partial derivative of $\mathcal{J}(\mathcal{V}_\lambda)$, $\partial\mathcal{J}(\mathcal{V}_\lambda)/\partial\mathcal{V}_{\lambda,m}(v)$ (as in (2.36)), is continuous and bounded w.r.t. any given eigenvalue $\mathcal{V}_{\lambda,m}(v)$. First, by (2.36), the continuity of $\partial\mathcal{J}(\mathcal{V}_\lambda)/\partial\mathcal{V}_{\lambda,m}(v)$ follows by the continuity of $u_{inv}(t)$ and $\partial u_{inv}(t)/\partial\mathcal{V}_{\lambda,m}(v)$ in $\mathcal{V}_{\lambda,m}(v)$.
Next, we show that $\partial\mathcal{J}(\mathcal{V}_\lambda)/\partial\mathcal{V}_{\lambda,m}(v)$ is bounded. Note that the inverse input $u_{inv}(t)$

is bounded. As $-C \leq \mathcal{V}_{\lambda,m}(v)$ for some finite $C > 0$, and the transition time T_m is limited, thereby $e^{\hat{A}_\xi t}$ is bounded and $\gamma(t)$, $\xi(t)$, $y^r(t)$, are all bounded (by Eqs. (2.25), (2.20) and (2.27), respectively). Thus, it remains to show that $\partial u_{inv}(t)/\partial \mathcal{V}_{\lambda,m}(v)$ is bounded. Since

$$\left\| \frac{\partial \hat{A}_\xi^n/n!}{\partial \mathcal{V}_{\lambda,m}(v)} \right\|_\sigma = \frac{\left\| \sum_{i=1}^n \hat{A}_\xi^{n-i} \frac{\partial \hat{A}_\xi}{\partial \mathcal{V}_{\lambda,m}(v)} \hat{A}_\xi^{i-1} \right\|_\sigma}{n!} \leq \frac{\sigma_1^{n-1}\sigma_2}{(n-1)!},$$

where $\sigma_1 = \|\hat{A}_\xi\|_\sigma$, $\sigma_2 = \|\partial \hat{A}_\xi/\partial \mathcal{V}_{\lambda,m}(v)\|_\sigma$, and $\|A\|_\sigma$ denotes the maximal singular value of A,

$$\begin{aligned}
\left\| \partial e^{\hat{A}_\xi t}/\partial \mathcal{V}_{\lambda,m}(v) \right\|_\sigma &= \left\| \frac{\partial (I + \hat{A}_\xi t + \hat{A}_\xi^2 t^2/2 + \cdots + \hat{A}_\xi^n t^n/n! + \dots)}{\partial \mathcal{V}_{\lambda,m}(v)} \right\|_\sigma \\
&= \left\| \frac{\partial \hat{A}_\xi t}{\partial \mathcal{V}_{\lambda,m}(v)} + \frac{\partial \hat{A}_\xi^2 t^2}{\partial \mathcal{V}_{\lambda,m}(v)} + \cdots + \frac{\partial \hat{A}_\xi^n t^n/n!}{\partial \mathcal{V}_{\lambda,m}(v)} + \dots \right\|_\sigma \\
&\leq \sigma_2 t + \sigma_1 \sigma_2 t^2 + \frac{\sigma_1^2 \sigma_2 t^3}{2} + \cdots + \frac{\sigma_1^{n-1}\sigma_2 t^n}{(n-1)!} + \dots \\
&= e^{\sigma_1 t}\sigma_2 t,
\end{aligned}$$

is bounded at any finite t and thereby, by (2.49), $\partial \gamma(t)/\partial \mathcal{V}_{\lambda,m}(v)$ is bounded also. Thus, the boundedness of the derivative $\partial \mathcal{J}(\mathcal{V}_\lambda)/\partial \mathcal{V}_{\lambda,m}(v)$ follows by that of $\partial u_{j,inv}(t)/\partial \mathcal{V}_{\lambda,m}(v)$ (by (2.45, 2.48)). This completes the proof. □

We summarize below the algorithm [33, 49] to obtain the optimal desired output that meets Objective $\mathcal{O}_3$:

Algorithm 1 Input-output Engery Minimization

For the given cost function $\mathcal{J}(\mathcal{V}_\lambda)$ in Eq. (31). The conjugate gradient method is presented as following.

```
Step 1. Initialization: At $k = 1$, choose an initial point $\mathcal{V}_\lambda^1$, so that $\mathbf{g}_1 \neq 0$, then set $\mathbf{d}_1 = -\mathbf{g}_1$;
Step 2. Compute an $\alpha_k > 0$ that satisfies Condition 2 utilizing bi-section method;
Step 3. Let $\mathcal{V}_\lambda^{k+1} = \mathcal{V}_\lambda^k + \alpha_k \mathbf{d}_k$. If $|\mathbf{g}_{k+1}| > \epsilon_t$ (the chosen computation error threshold), compute $\beta_k$ by Eq. (58) and then update $\mathbf{d}_k$ by $\mathbf{d}_{k+1} = \beta_k \mathbf{d}_k - \mathbf{g}_k$. Let $k := k + 1$, go to Step 2. If $|\mathbf{g}_{k+1}| \leq \epsilon_t$, stop.
```

2.3.3 Objective $\mathcal{O}_4$: Optimal Transition Trajectory with Minimization of Transition Time

Once Objective $\mathcal{O}_3$ is satisfied, we further minimize the transition time for a given weighting matrix $H_{\mathbb{Y}}$, under the condition that the obtained inverse input amplitude stays within the given threshold value. Such a time minimization for each transition can be obtained by one dimensional search method, for example, the multi-dimensional bi-section method, the golden-section method, or the Fibonacci method [48].

We realize that in the proposed approach, Objective $\mathcal{O}_3$ is achieved with the transition time T_m fixed, thereby, might be compromised by the above time optimization process. Such an issue can be alleviated through an iteration process, by repeating the above two optimization steps, until the transition time cannot be reduced further (i.e., the change of the transition time is small enough). This iteration process leads to a sub-optimal minimal time solution that is optimal in the sense of Objectives $\mathcal{O}_1$-$\mathcal{O}_3$.

2.4 Simulation Example

We demonstrate the proposed approach through a simulation example of non-periodic tracking-transition in nanomanipulation applications. As the design of the optimal transition trajectory that achieves Objectives $\mathcal{O}_1$ and $\mathcal{O}_2$ and its tracking have been demonstrated in [9], central to the simulation is to illustrate that in addition to Objectives $\mathcal{O}_1$ and $\mathcal{O}_2$. Objectives $\mathcal{O}_3$ and $\mathcal{O}_4$ can also be achieved by using the proposed approach.

2.4.1 Simulation Model

Probe-based nanomanipulation (PBM) [9] provides an ideal scenario for applying the proposed approach where non-periodic tracking-transition switching is involved in the operation with one complete operation consisting of precision tracking of a given desired trajectory to perform trajectory dependent tasks (e.g., to push a nanotube to a specific location)–the tracking section, and swift repositioning the probe to the desired locations (e.g., to relocate the probe close to the nanotube where the next operation takes place)– the transition section. In the simulation, the following piezoelectric actuator model for nanomanipulation operation as in [9] was considered:

$$G(s) = K_x \prod_{q=1}^{4}(s - z_q) \Bigg/ \prod_{r=1}^{6}(s - p_r), \tag{2.60}$$

with the Laplace transform variable s in rad/ms, and

$$\begin{aligned} K_x &= 29.28, \\ z_q &= 9.274 \pm 41.659i, -2.484 \pm 30.434i, \\ p_r &= -0.188 \pm 31.326i, -0.857 \pm 24.570i, -20.263, -15.198. \end{aligned}$$

2.4.2 Implementation of the Proposed Approach

Without loss of generality, we consider that the entire operation consists of three tracking sessions and four transition sessions in alternation (see Fig. 1). First, the candidate transition trajectories that meet Objectives $\mathcal{O}_1$ and $\mathcal{O}_2$ were obtained by using the OTTDT technique [9], as summarized in Sec. 2.3.1. Then the optimal desired trajectory that meets Objectives $\mathcal{O}_1$–$\mathcal{O}_3$ was obtained by following the steps described in Sec. 3.2. Finally, the transition time was further minimized for the obtained optimal transition trajectory (Objective $\mathcal{O}_4$), with comparison to the solution obtained by using the OTTDT technique. When converting the constrained minimization problem (see (2.9)) to the unconstrained one (see (2.31)), the coefficient $c = 100$ in the cost function $\mathcal{J}(\mathcal{V}_\lambda)$ was chosen through a few tests so that it is sufficiently large to guarantee the equivalence of the unconstrained cost function in Lemma 5.

Next, the set of optimal eigenvalues for each transition session in the effective preview time was obtained. For the relative degree of two in system (2.60), a total of 8 eigenvalues were to be determined. The threshold of the input was set at $\mathbf{M}_u = 5$, and the length of the transition sessions, S_1, S_2, S_3, S_4 (see Fig. 2(a)), were initially set at $\{T_m\} = \{1.20,\ 1.20,\ 1.20,\ 0.60\}$ ms according to the average speed of the adjacent desired trajectories. To determine the domain D over which the optimal eigenvalues to be searched, we noted that as the optimal solution for Objectives $\mathcal{O}_1$ and $\mathcal{O}_2$,

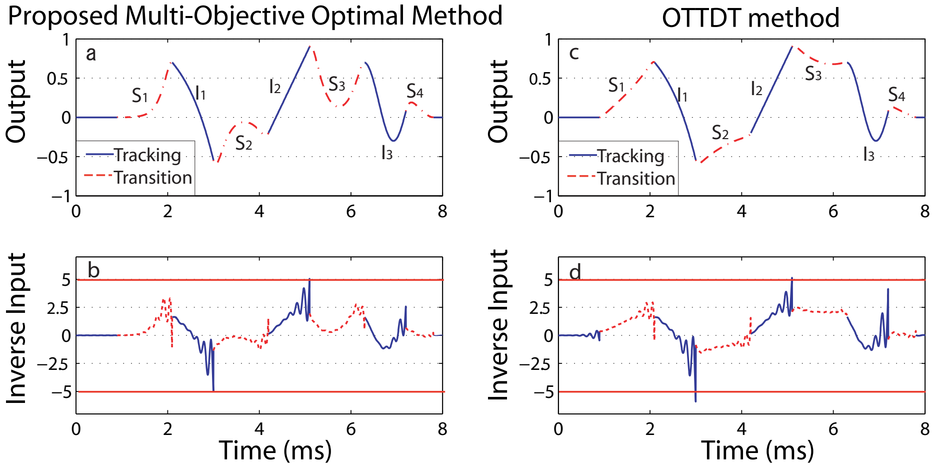

Figure 2.2: The comparison of the proposed multi-objective optimal method and the OTTDT method before the minimization of the time intervals.

$\gamma^*(t)$, was symmetrical w.r.t each eigenvalue of the stabilized state matrix of the output subdynamics $\hat{A}_\xi$, so were $u_{inv}(t)$ and $\mathcal{J}(\mathcal{V}_\lambda)$, the range of the eigenvalues could be determined by one dimensional search method by choosing a large enough upper bound for the input amplitude during the search. The search range of the eigenvalues for each transition $S_1,\ S_2,\ S_3,\ S_4$ were chosen as, respectively, $D_1 = [-19, 0] \times [-19, 0]$, $\quad D_2 = [-18, 0] \times [-18, 0]$, $D_3 = [-15, 0] \times [-15, 0]$, $\quad D_4 = [-42, 0] \times [-42, 0]$.

2.4.3 Simulation Results and Discussion

The optimal set of eigenvalues and the corresponding minimal transition time intervals were at:

$$S_1:\ \mathcal{V}_{\lambda,1} = [-8.17,\ \ -8.19], \quad S_2:\ \mathcal{V}_{\lambda,2} = [-7.65,\ \ -7.56],$$
$$S_3:\ \mathcal{V}_{\lambda,3} = [-7.04,\ \ -6.94], \quad S_4:\ \mathcal{V}_{\lambda,4} = [-6.30,\ \ -6.69],$$
$$\{T_m^*\} = \{0.22,\ 0.30,\ 0.25,\ 0.14\}.$$

The transition trajectories and the corresponding inputs obtained by using the proposed method are compared with those obtained by using the OTTDT technique (with $H_{\xi,k} = [31.62\ 32.61]$ for each transition session) in Fig. 2 for the initial choice of transition times (i.e., without minimizing the transition time). The minimized transition time intervals obtained above were employed in the OTTDT technique [9], and the obtained entire output trajectory and the corresponding input are compared to those

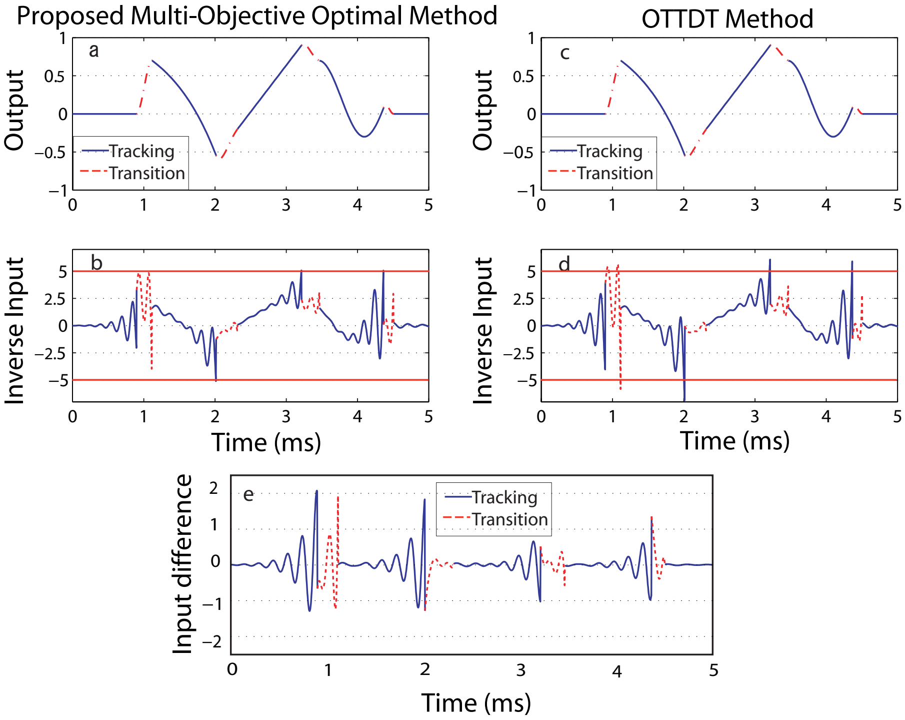

Figure 2.3: The comparison of the proposed multi-objective optimal method and the OTTDT method after the minimization of the time intervals.

obtained by using the proposed method in Fig. 3, and the input differences between these two methods are compared in Fig. 3(e).

The simulation results with the fixed (not minimized) transition periods clearly showed the improvements of the proposed approach over the OTTDT technique in minimizing the input-energy under amplitude constraint. As shown in Fig. 2, although smooth transition output trajectories were obtained by using both methods, the input obtained by using the OTTDT method exceeded the threshold value of 5, with the input energy over 53.3% larger than that by using the proposed approach (Compared Fig. 2(b) to Fig. 2(d)). We note that the output obtained by the proposed approach varied in transition periods S_2 and S_3 (Compared Fig. 2(a) to Fig. 2(c)), however, no large input oscillations occurred and the output was still smooth. Such an output energy and input energy minimization trade-off can be easily achieved by using the proposed approach. Therefore, the proposed approach provided an effective mean to the amplitude-constrained input-energy minimization in optimal output design and tracking in non-periodic tracking-transition switching.

The advantage of the proposed approach are further demonstrated via additional transition time minimization. With the minimized transition periods, the control input obtained by using the proposed approach still stayed within the threshold value at 5 (see Fig. 3(b)), whereas the input obtained by using the OTTDT technique violated the amplitude constraint at even more places, with larger oscillations (Compared Fig. 3(b) to Fig. 3(d)), and thereby, larger energy (15.9% higher) than that by the proposed approach. We further note that the similarity of the output in contrast to the significant input difference (see Fig. 3 (d)) manifested the efficacy of the proposed approach in achieving multi-objective input-output optimization for non-periodic tracking transition switching. The proposed approach can be further enhanced for robustness in the presence of system dynamics variation and disturbances and online numerical computation efficiency. These robustness and computation related issues will be addressed in future work.

2.5 Conclusion

In this chapter, a multi-objective optimization technique to trajectory design and tracking with non-periodic tracking-transition switching is proposed for non-minimum phase systems. The proposed approach extended the previous work on trajectory design and tracking of non-periodic tracking-transition switching to further minimize the input energy under amplitude constraint and then the transition time. It is shown that the amplitude-constrained input-energy minimization problem can be converted to an unconstrained one. Then the optimal parameters in designing the transition output trajectory were sought through an improved conjugate gradient method, and the minimal transition time was further obtained via one dimensional search.

Chapter 3

On Superposition of Hammerstein Systems: Application to Simultaneous Hysteresis-Dynamics Compensation—Piezoelectric Actuator Experiment

Abstract

Superposition principle (SP)—the response (output) of a linear system to a weighted combination of inputs equals to the same weighted combination of the outputs each corresponding to the individual inputs, respectively—is one of the most fundamental properties of linear systems, and has been exploited for controls, for example, in the development of model predictive control. Extension of the SP beyond linear systems, however, is largely limited. In this chapter, the almost superposition of Hammerstein systems (ASHS) and its application to precision control of hysteresis-Hammerstein systems is studied. We first show, under some minor conditions, the existence of a non-strict form ASHS, and under one further condition, the strict-form ASHS. We then present one application of the ASHS—simultaneous hysteresis and dynamics compensation in output-tracking of hysteresis-Hammerstein systems, where offline tracking of output elements based on iterative learning control is integrated with online input synthesis based on an inverse Preisach modeling. The proposed ASHS-based technique is further enhanced through two online optimization schemes. The proposed technique is demonstrated through experiments on a piezoelectric actuator.

3.1 Introduction

Superposition principle (SP), where the response (output) of a system to a linear combination of inputs equals to the same linear combination of the responses (outputs) each corresponding to the individual inputs, respectively, is one of the most fundamental properties of linear systems (e.g., [50]). The superposition principle not only significantly simplifies the analysis of linear systems, but also finds important applications in control areas [51, 3, 26]. Extending the SP beyond linear systems, however, is rather limited, and nonlinear superposition principle (NSP) has only been shown for specific types of nonlinear systems [52]. Contrary to its importance in physics to study, for example, the supersymmetric quantum mechanics [53], NSP has rather limited implementation in controls. However, SP provides a natural and conceptually straightforward avenue to exploit *a priori offline* learning for *online* tracking. These challenges and needs motivate this work to study the almost superposition of hysteresis systems (ASHS) and its application to the control of smart actuators.

SP plays an important role in controls in both theoretical and applicational perspectives. For example, the superposition principle is central to modeling multi-input-multi-output systems when implementing the model predictive control (MPC) in industries such as petroleum factories and chemical plants [54, 51], where system modeling constitutes the major cost of the MPC implementation [54, 51]. Beyond modeling, recently, the superposition principle has also been explored for precision tracking/regulation in advanced manufacturing applications beyond repetitive tasks [3, 55]. The idea of exploiting the advantages of iterative learning control (e.g., high performance, good robustness against system variations, and ease of implementation) beyond repetitive operations has been extended to general output-tracking of online-specified, partially-known (i.e., previewed) desired trajectory for linear systems [26, 34]. Nonlinear superposition principle, however, is only known for the so called *Lie dynamic systems* [52], where the general response of the system can be given by a known function (e.g., a rational function) of a few specific responses. Contrary to its impact on the analysis of these systems (e.g., stability) [56, 57], the NSP has limited use in tracking/regulation of nonlinear systems. Particularly, the input-output behavior of hysteresis systems does not admit the form of Lie dynamic systems, but that of Hammerstein systems instead—a static nonlinear operator followed by a linear dynamics [19]. Thus, study of the superposition principle of Hammerstein systems is needed to, for example, compensate for both hysteresis and dynamics in the control of these systems.

Currently, simultaneous hysteresis and dynamics compensation still remains challenging in the control of hysteresis systems. For example, the Preisach model has been explored in a feedforward-feedback control manner [58]. The Preisach-based feedforward controller, however, is complicated to implement. Instead of finding the Preisach model then inverting it, it has been proposed to directly use the inverse Preisach model in the controller design [11, 12], or update the inverse of Preisach model adaptively [13]. These techniques, however, are limited to compensating for hysteresis in the quasi-static case—accounting for both dynamics and hysteresis effects was not adequately addressed. Alternatively, it has been proposed to modify the inverse Preisach model to account for the rate-dependent hysteresis effect (i.e., the combined hysteresis and dynamics effects) [14]. However, the compensation of the dynamics effect is rather constrained as only a few excitation frequencies are considered [14]. To improve the dynamics compensation, techniques based on an extension of neural network [15, 16] and the inverse rate-dependent Prandtl-Ishlinskii (PI) model [17] have also been proposed. These existing approaches, however, require a complicated modeling process—which is prone to modeling errors and heavy online computations. Also, the performance-robustness trade-off involved in these approaches may not be efficient in accounting for the usually quasi-static uncertainty/variation in hysteresis systems such as smart actuators. These robustness and complexity (in both modeling and control) related issues can be largely alleviated through the iterative learning control (ILC) framework [18, 19], as the high tracking performance obtained via the ILC-based techniques can be maintained against model uncertainties and/or system variations through a couple of iterations. Therefore, the superposition principle of hysteresis systems can serve as a bridge to extend ILC techniques for more general non-repetitive tracking/regulation.

The main contribution of this work is the development of the ASHS. First, we show that for Hammerstein operators satisfying a Lipschitz condition, a non-strict form ASHS exists, where the difference between a linear combination of outputs and the output to the linear combination of the corresponding inputs can be rendered arbitrarily small, provided that there are sufficiently large number of inputs (or outputs) satisfying some minor conditions, and the coefficients of the output combination and the input combination are allowed to be different. Under one further nonlinearity condition of the Hammerstein operator, the strict-form ASHS also holds, where the coefficients of the input combination and the output combination are the same. The ASHS can not only be a powerful tool for the analysis of Hammerstein systems, but also be exploited for control of these systems. As an example, we present, in this work, its use in simultaneous hysteresis-dynamics compensation in preview-based precision output tracking, where online output decomposition (via uniform B-splines) and input synthesis (via an inverse Preisach modeling) is integrated with *a priori* off-line ILC-based learning of the input-output elements mapping. In this approach, the ASHS provides the foundation to exploit *offline, a priori learning* for online tracking/regulation, thereby, decoupling "training" from "execution" in output tracking. Such a decoupling might be an effective avenue to alleviate complexities and challenges in modeling and control of Hammerstein systems [26, 34]. The efficacy of the proposed approach is demonstrated through experiments on a piezoelectric actuator.

3.2 Superposition Principle of Hammerstein Systems

3.2.1 Problem Formulation

We consider a Hammerstein system consisting of a rate-independent Hammerstein operator $\mathbb{H}[\cdot]$ followed by a linear dynamics, where the static Hammerstein operator, $\mathbb{H}[\cdot] : \Re \to \Re$, given by

$$v(t) = \mathbb{H}[u(t)], \tag{3.1}$$

satisfies the following assumption:

Assumption 8. *The Hammerstein operator* $\mathbb{H}[\cdot]$ *is bounded input, bounded output(BIBO) stable and bi-Lipschitz, i.e, for any given continuous input* $u(t)$*, there exists a constant* $L_h \in \Re^+$ *such that for any given two time instants* t_1*,* t_2*,*

$$\frac{1}{L_h}|u(t_1) - u(t_2)| \leq |v(t_1) - v(t_2)| \leq L_h|u(t_1) - u(t_2)|. \tag{3.2}$$

Without much restriction, we further assume that the output trajectories are all Lipschitz continuous:

Assumption 9. *The output trajectories $v_d(\cdot)$s to be tracked by system (3.1) are compactly supported, and belong to the set of Lipschitz continuous functions, $\mathcal{V}$, i.e, for any $v_d(\cdot) \in \mathcal{V}$, there exists a Lipschitz constant $L_v \in \Re^+$ such that for any given two time instants $t_1,\ t_2$,*

$$|v_d(t_1) - v_d(t_2)| \leq L_v|t_1 - t_2|. \tag{3.3}$$

The above two assumptions imply that the input of system (3.1) is also Lipschitz:

Lemma 10. *For any given Hammerstein system (3.1), let Assumption 8 hold, then for any given output $v(t)$ satisfying Assumption 9, the corresponding input $u(t)$ is also Lipschitz, and the Lipschitz constant L_u equals to L_hL_v.*

Towards the superposition of Hammerstein systems, we consider the input elements $u_i(t)$s corresponding to those outputs in set $\mathcal{V}$, i.e., $H[u_i(t)] \in \mathcal{V}$, are from a set $\mathcal{U}_M$,

$$\begin{gathered}\mathcal{U}_M = \{u_i(t)|\ H[u_i(t)] \in \mathcal{V},\ i = 0,\ 1, \ldots,\ M,\\ M \in \mathbb{N}\ (\mathbb{N} : \text{Set of natural numbers})\},\end{gathered} \tag{3.4}$$

and satisfies the following assumption:

Assumption 11. *The input elements $u_i(t)$s $\in \mathcal{U}_M$ are smooth and compactly supported. Moreover, for any given $M \geq 2,\ M \in \mathbb{N}$ and any $2 \leq N \leq M,\ N \in \mathbb{N}$, the following set of distinct time instants*

$$\begin{gathered}\mathcal{S}_{\mathcal{T},N} \triangleq \{t_j|\ (j-1)\mathcal{T}/N \leq t_j < j\mathcal{T}/N,\\ j = 1\ , 2, \ldots, N\},\end{gathered} \tag{3.5}$$

$\mathcal{T} \in \Re$, satisfies the condition that for any two different time instants $t_m,\ t_n \in \mathcal{S}_{\mathcal{T},N}$, $m \neq n$, the set of input elements that are non-zero at t_m is different from that at t_n, i.e., for any two time instants $t_m,\ t_n \in \mathcal{S}_{\mathcal{T},N}$, $\Theta_m \cap \Theta_n^c \neq \emptyset$ and $\Theta_m^c \cap \Theta_n \neq \emptyset$, where set Θ_i is defined as $\Theta_i \triangleq \{u_j(t)|\ u_j(t_i) \neq 0,\ u_j(t) \in \mathcal{U}_M\}$, and Θ^c is the complement set of set Θ.

The set of output elements each corresponding to those inputs in $\mathcal{U}_M$, respectively, is denoted as $\mathcal{V}_M$.
In control applications, Assumption 11 can be satisfied through the design of the inputs set $\mathcal{U}_M$, for example, by constructing $\mathcal{U}_M$ as a set of compact-supported spline function (e.g. B-splines) and its time-shifting copies.

3.2.2 Non-strict Almost Superposition of Hammerstein Systems

Definition 12. Non-strict Almost Superposition of Hammerstein Systems (Non-strict ASHS) Let Assumptions 8, 9, and 11 be satisfied, then the Non-strict ASHS is to, for any given $\epsilon > 0$ and any given finite summation of M distinct number of output elements from set $\mathcal{V}_M$, $v_d(t)$, defined in a compact domain $[0,\ T]$, $T \in \Re$,

$$v_d(t) \triangleq \sum_{k=1}^{M} g_k v_k(t), \quad t \in [0,\ T],\ v_k(t) \in \mathcal{V}_M, \tag{3.6}$$

obtain a finite number $M^* \in \mathbb{N}$ and a sequence of coefficients $\mathbf{p}_M = [p_1,\ p_2, \dots,\ p_M]^\top$, such that for the following finite summation of the corresponding input elements $u_k(\cdot)s \in \mathcal{U}_M$, $u(t)$,

$$u(t) = \sum_{k=1}^{M} p_k u_k(t), \quad t \in [0,\ T],\ p_k \in \Re, \tag{3.7}$$

almost superposition (AS) holds for any $M \geq M^*$ in the sense that

$$\|e_u(t)\|_{\mathcal{L}_{\infty,T}} = \|u_d(t) - u(t)\|_{\mathcal{L}_{\infty,T}} \leq \epsilon \tag{3.8}$$

holds, where in Eqs. (3.6-3.8), g_ks, p_ks $\in \Re$ are constants, $u_d(\cdot)$ is the exact-tracking input for the summed output $v_d(\cdot)$, i.e., $\|v_d(t) - \mathbb{H}[u_d(t)]\|_{\mathcal{L}_{\infty,T}} = 0$, and $\|f(\cdot)\|_{\mathcal{L}_{\infty,T}}$ denotes the $\mathcal{L}_\infty$ norm of $f(\cdot)$ restricted to a finite time interval $[0,\ T]$, $T < \infty$.

Lemma 13. *Let the conditions given in Definition 12 be satisfied, then*

- *For any given $M \geq N$ and time $T \in \Re$, there exists a set of time instants $\mathcal{S}_{T,N}$,* *such that rank*$(U_e) = N$*, where*

$$U_e = \begin{pmatrix} u_1(t_1) & u_2(t_1) & \dots & u_M(t_1) \\ u_1(t_2) & u_2(t_2) & \dots & u_M(t_2) \\ \vdots & \vdots & \vdots & \vdots \\ u_1(t_N) & u_2(t_N) & \dots & u_M(t_N) \end{pmatrix}_{N\times M} \quad ; \tag{3.9}$$

- *For any given* $\epsilon > 0$*, the constant* M^* *for which the Non-strict ASHS holds is given by*

  $$M^* \triangleq \lceil 2TL_hL_v/\epsilon \rceil\,, \tag{3.10}$$

  where $\lceil x \rceil$ *denotes the ceiling function (i.e., the smallest integer larger than* x*), and* L_h *and* L_v *are as given in Eq. (3.2, 3.3), respectively;*

- *And the corresponding vector of coefficients* $\boldsymbol{p}_M$ *is given by an equivalent class defined as*

  $$\{U_e^g \boldsymbol{u}_d + \boldsymbol{p}|\ \boldsymbol{p} \in Null(U_e)\}, \tag{3.11}$$

  where $\boldsymbol{u}_d = [u_d(t_1),\ u_d(t_2), \dots,\ u_d(t_N)]^\top$*,* X^g *denotes the Moore-Penrose pseudoinverse of matrix* X*, and Null*(U_e) *denotes the null space of the matrix* U_e*.*

Proof. First, we show that rank$(U_e) = N$. Assumption 11 implies there exists a finite set of $N \leq M$ number of time instants in $\mathcal{S}_{T,N}$, such that the matrix U_e has full row rank, i.e., rank$(U_e) = N$.

To show the second and third statements, matching the linear combination of the input elements and the desired input $u_d(t)$ at the time instants $\mathcal{S}_{T,N}$ yields

$$u(t_j) = \sum_{k=1}^{M} p_k u_k(t_j) = u_d(t_j), \tag{3.12}$$

which can be rewritten in matrix form as

$$U_e \mathbf{p}_M = \mathbf{u}_d, \tag{3.13}$$

where U_e is given by Eq. (3.9). The above Eq. (3.13) implies that the non-trivial solution of $\mathbf{p}_M$ falls into an equivalent class determined by the null space of U_e and a special solution depending on $\mathbf{u}_d$.

As the maximal time interval between two successive time instants in set $\mathcal{S}_{T,N}$ is given by $\delta t_{\max} \triangleq \max_{j=1,2,\ldots,N-1}(t_{j+1} - t_j) < 2T/N$, and the input and the summation of the input elements are all Lipschitz with Lipschitz constant $L_h L_v$, for any given $\epsilon > 0$, the superposition error can be rendered to be below ϵ by selecting $N \geq N^* \triangleq \lceil 2TL_hL_v/\epsilon \rceil$, and thereby, $M \geq M^*$, such that

$$\delta t < \epsilon/(L_h L_v).$$

As $N \leq M$, $M^* = N^*$, and the corresponding vector of coefficients $\mathbf{p}_M$ is as given by Eq. (3.11). This completes the proof. □

3.2.3 Strict Almost Superposition of Hammerstein Systems

As clearly the strict almost superposition of Hammerstein systems holds when the Hammerstein operator $\mathbb{H}[\cdot]$ is linear, it is further assumed that

Assumption 14. *The Hammerstein operator* $\mathbb{H}[\cdot]$ *is nonlinear almost everywhere, i.e.,* $m(\mathbb{H}[x] = rx) = 0$ *for any given* $x \in \Re$*, and any* $r \in \Re$ *(*$m(Z)$*: the Lebesgue measure of any measurable set* $Z \subset \Re$*). Moreover,* $v(t) = \mathbb{H}[u(t)] = 0$ *if and only if* $u(t) = 0$ *for any* $t \in \Re$*.*

With this Assumption, the Lemma below can be easily verified.

Lemma 15. *Let Assumptions 8, 11, and 14 hold, then the properties of the inputs as stated in Assumption 11 also hold for the corresponding output elements.*

Definition 16. Strict Almost Superposition of Hammerstein Systems (Strict-ASHS) Let Assumptions 8, 9, 11, and 14 be satisfied, then the Strict-ASHS is to, for any given $\epsilon > 0$, and any given set of distinct output elements from $\mathcal{V}_M$ with $M \geq M^*$, obtain a finite $M^* \in \mathbb{N}$ and a sequence of coefficients $\mathbf{g}_M = [g_1, g_2, \ldots, g_M]^\top$, such that the linear superposition holds in the above AS sense for any $M \geq M^*$ between the following finite summation of output elements, $v_d(t)$, defined in a compact domain $[0,\ T]$, $T \in \Re$,

$$v_d(t) \triangleq \sum_{k=1}^{M} g_k v_k(t), \quad t \in [0,\ T],\ v_k(t) \in \mathcal{V}_M, \tag{3.14}$$

and the corresponding summation of the respective input elements $u_k(\cdot)s \in \mathcal{U}_M$, $u(t)$,

$$u(t) = \sum_{k=1}^{M} g_k u_k(t), \quad t \in [0,\ T]. \tag{3.15}$$

Lemma 17. *Let the conditions in Definition 16 hold, then*

- *For any given $M \in \mathbb{N}$, there exists a finite set of N number of time instants $\mathcal{S}_{T,N}$, such that rank$(W_e) = 2N$ for $2N \leq M$, where*

$$W_e = [V_e^\top\ U_e^\top]^\top, \quad \textit{with} \tag{3.16}$$

$$V_e = \begin{pmatrix} v_1(t_1) & v_2(t_1) & \dots & v_M(t_1) \\ v_1(t_2) & v_2(t_2) & \dots & v_M(t_2) \\ \vdots & \vdots & \vdots & \vdots \\ v_1(t_N) & v_2(t_N) & \dots & v_M(t_N) \end{pmatrix}_{N\times M}, \tag{3.17}$$

and

$$U_e = \begin{pmatrix} u_1(t_1) & u_2(t_1) & \dots & u_M(t_1) \\ u_1(t_2) & u_2(t_2) & \dots & u_M(t_2) \\ \vdots & \vdots & \vdots & \vdots \\ u_1(t_N) & u_2(t_N) & \dots & u_M(t_N) \end{pmatrix}_{N\times M}; \tag{3.18}$$

- *For any given $\epsilon > 0$, the constant M^* for which the Strict-ASHS holds is given by*

$$M^* \triangleq \lceil 4TL_hL_v/\epsilon \rceil, \tag{3.19}$$

where L_h and L_v are given in Eq. (3.2, 3.3), respectively;

- *The corresponding vector of coefficients $\boldsymbol{g}_M$ is given by an equivalent class defined as*

$$\{W_e^g[\boldsymbol{v}_d^\top\ \boldsymbol{u}_d^\top]^\top + \boldsymbol{g}|\ \boldsymbol{g} \in Null(W_e)\}, \tag{3.20}$$

where $\boldsymbol{v}_d = [v_d(t_1),\ v_d(t_2), \dots,\ v_d(t_N)]^\top$, and $\boldsymbol{u}_d = [u_d(t_1),\ u_d(t_2), \dots,\ u_d(t_N)]^\top$.

Proof. First, we show that there exists a finite set of N number of time instants $\mathcal{S}_{T,N}$, such that rank$(W_e^\top) = 2N$ for any given $2N \leq M$. Assumption 11 implies that the matrix U_e has full row rank, i.e., rank$(U_e) = N$, and, consequently, by Lemma 15, rank$(V_e) = N$. Moreover, by Assumption 14, any row in matrix V_e, $V_{e,j}$, for $j =$

$1,\ 2, \ldots,\ N$, is linearly independent with the corresponding row in matrix U_e, $U_{e,j}$, and linearly independent with all other rows of matrix U_e, as each row in matrix V_e, $V_{e,j}$, for $j = 1,\ 2, \ldots,\ N$, has zero elements at locations where any other row of matrix U_e, $U_{e,i}$ for $i \neq j$ is not zero. Therefore, $\text{rank}(W_e) = 2N$.

To show the second and third statements, matching the linear combination of the input elements and the desired input $v(t)$ at the time instants $\mathcal{S}_{T,N}$ gives

$$v_d(t_j) = \sum_{k=1}^{M} g_k v_k(t_j), \tag{3.21}$$

and the matching of corresponding outputs yields,

$$u(t_j) = \sum_{k=1}^{M} g_k u_k(t_j) = u_d(t_j). \tag{3.22}$$

Representing the above Eqs. (3.21, 3.22) in a matrix form yields

$$W_e^\top \mathbf{g}_M = [\mathbf{v}_d^\top\ \mathbf{u}_d^\top]^\top, \tag{3.23}$$

where W_e is given in Eq. (3.16). The above Eq. (3.23) implies that the non-trivial solution of $\mathbf{g}_M$ falls into an equivalent class determined by the null space of W_e and a special solution depending on $[\mathbf{v}_d^\top\ \mathbf{u}_d^\top]^\top$.

The rest of the proof is similar to the counterpart in Lemma 13, and thereby omitted to save the space. This completes the proof. □

3.3 A Realization of ASHS to Hysteresis Systems

Next, we present a realization and implementation of the ASHS for hysteresis systems—one type of Hammerstein systems with wide applications, where the input-output mapping of the hysteresis operator $\mathbf{H}[\cdot]$ in

$$v(t) = \mathbf{H}[u(t)], \tag{3.24}$$

satisfies Assumptions 8, 14, and furthermore, can be described by a Preisach model [59]. Particularly, the proposed approach below exploits uniform B-splines for output decomposition and an inverse Preisach modeling for input synthesis. With output tracking in mind, we assume that

Assumption 18. *The output trajectory $v_d(t)$ is smooth enough, and at any given time instant t_c, the future desired trajectory is known (i.e. previewed) for a finite amount of preview time $T_p < \infty$.*

The previewed desired output trajectory is then decomposed/approximated as

$$v_d(t) \approx \sum_{k=1}^{M} g_k v_k(t), \tag{3.25}$$

with the approximation error small enough (with respect to the desired tracking precision). Provided that the conditions in the Non-strict ASHS are satisfied, the corresponding control input can then be synthesized as

$$u_d(t) \approx \sum_{k=1}^{M} p_k u_k(t) = \sum_{k=1}^{M} h_k g_k u_k(t), \tag{3.26}$$

when the number of input elements $u_k(t)$s are large enough, and furthermore, provided Assumption 14 is satisfied, $p_k = g_k$, i.e., $h_k = 1$. We seek to obtain h_ks analytically based on an inverse Preisach model. We start with the construction of the library of input-output elements, $\mathcal{L}_H$, based on uniform B-splines.

3.3.1 Uniform B-spline-based Library for Trajectory Decomposition

The main advantage of using uniform B-splines for trajectory decomposition is that only a few types of output elements are needed for the decomposition [26]—in offline learning, tracking of only a few different output elements needs to be obtained. As hysteresis effect is range-dependent, it is beneficial to employ output elements of different amplitudes in the decomposition–to exploit the input-output mapping for hysteresis compensation. Thus, we propose a library of nestle structure where the main library $\mathcal{L}_H$ consists of sub-libraries $\mathcal{L}_{H,i}$ each containing output elements of the same amplitude:

$$\mathcal{L}_H = \{\mathcal{L}_{H,1},\ \mathcal{L}_{H,2},\ \ldots,\ \mathcal{L}_{H,N_H}\}, \quad \text{with} \tag{3.27}$$

$$\mathcal{L}_{H,i} = \{(v_{e,k}(\cdot,\tau_i),\ u_{e,k}(\cdot,\tau_i))|\ k = 1,\ 2,\ldots,\ N_e\}, \tag{3.28}$$

where $v_{e,k}(\cdot,\tau_i)$s are output elements of amplitude $\tau_i \in [\tau_{\min},\ \tau_{\max}]$, and N_H, N_e are the number of total sub-libraries (i.e., number of output amplitude levels) and the number of total elements in each sub-library, respectively. Without loss of generality, the output range of τ_is, $[\tau_{\min},\ \tau_{\max}]$, are uniformly discretized into N_H levels, with the step length $\Delta\tau$ given as

$$\Delta\tau = (\tau_{\max} - \tau_{\min})/(N_H - 1). \tag{3.29}$$

The output elements $v_{e,k}(\cdot, \tau_i)$s satisfying Assumption 9 are homogeneous with respect to τ_i, i.e.,

$$v_{e,k}(t, \tau_i) = \tau_i \hat{v}_{e,k}(t), \quad \text{for } k = 1,\ 2, \ldots,\ N_e, \tag{3.30}$$

where $\hat{v}_{e,k}(t)$s, with no loss of generality, are the output elements at a pre-chosen amplitude (called the *base output elements* below), and $u_{e,k}(t,\ \tau_i)s \in \Re$ are the exact-tracking input elements corresponding to $v_{e,k}(t, \tau_i)$s.

B-spline-based Library Construction Without loss of generality, we present the construction of the base output element $\hat{v}_{e,k}(t)$ via B-splines only (the output elements of any given sub-library $\mathcal{L}_{H,i}$ are specified via Eq. (3.30) accordingly),

$$\hat{v}_{e,k}(t) = B_{k,s}(t), \tag{3.31}$$

where $B_{k,s}(t)$ is one of the s^{th}-degree B-spline basic functions obtained recursively via

$$\begin{aligned} B_{k,s}(t) =& \frac{t - t_k}{t_{k+s} - t_k} B_{k,s-1}(t) \\ &+ \frac{t_{k+s+1} - t}{t_{k+s+1} - t_{k+1}} B_{k+1,s-1}(t), \end{aligned} \tag{3.32}$$

where

$$t_k = k\Delta T \tag{3.33}$$

is the decomposition knots, with $\Delta T = T_p/m$ the knot period. Initially,

$$B_{k,0} = \begin{cases} 1, & t_k \le t < t_{k+1}; \\ 0, & otherwise, \end{cases}$$

for $k = -s+1,\ -s+2, \ldots,\ m+s-1$, and m is the total number of the decomposition intervals in $[t_c, t_c + T_p]$. The decomposition coefficients g_ks can be obtained through a linear quadratic programming scheme [22] [60]. Moreover, the uniform B-splines and the corresponding inputs satisfy Assumptions 9, 11.

Note that usually in the decomposition, truncation of the B-splines are usually needed to match the beginning and the end portions of the trajectory to be decomposed, i.e., the output elements in sub-library $\mathcal{L}_{H,i}$ in Eq. (3.28) consists of one uniform B-spline and other truncated ones that are extended and smoothly transited to zero (so that exact tracking of the truncated ones exists and can be obtained a priori via ILC) [26]. However, as shown in [61], such a truncation can be avoided by shifting the decomposition instants and redefining the decomposition coefficients g_ks. Thus, each sub-library $\mathcal{L}_{H,i}$ consists of only one element ($N_e = 1$ in Eq. (3.28)). This is attractive in practices when output elements of different amplitudes are used.

As there is only one element in each sub-library, we simplify, in the rest of the chapter, the notation of input-output element, $v_{e,k}(t,\ \tau_i)$ and $u_{e,k}(t,\ \tau_i)$, as $v_e(t,\ \tau_i)$ and $u_e(t,\ \tau_i)$, respectively.

3.3.2 Superposition of Hysteresis Systems via an Inverse Preisach Model

Preisach Model and Its Inverse[59] [12]

Next, we propose an inverse hysteresis model based on the Preisach modeling approach. The output of the Preisach model is represented by the integral of the hysteresis operator over the interesting range, i.e.,

$$v(t) = \iint_S \mu(\alpha,\beta)\hat{\gamma}_{\alpha\beta}[u(t)]d\alpha d\beta, \tag{3.34}$$

where $\mu(\alpha,\beta)$ is the weighting function of the Preisach model, which could be determined experimentally, $\gamma_{\alpha\beta}[u(t)]$ is the Preisach hysteresis operator shown in Fig. (3.1a) and the region S is defined by

$$\begin{aligned} S \triangleq \{(\alpha,\beta) : \alpha \geq \beta,\ \alpha \in [-\alpha_0,\ \alpha_0], \\ \beta \in [\beta_0,\ -\beta_0], \text{for } \alpha_0 \in \Re^+,\ \beta_0 \in \Re^-\}, \end{aligned} \tag{3.35}$$

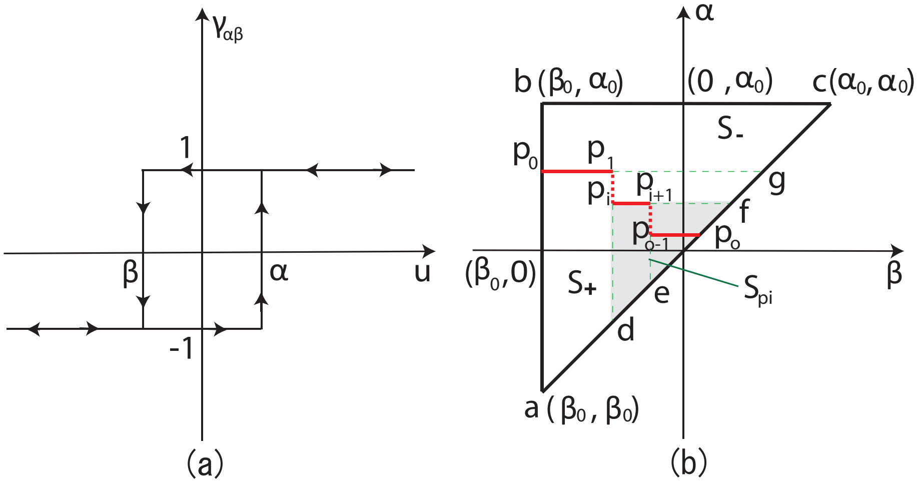

Figure 3.1: a. The hysteresis operator; b The hysteresis curve.

Observing from the Preisach plane shown in Fig. (3.1b), the above Eq. (3.34) can be transformed as

$$v(t) = 2 \iint_{S_+} \mu(\alpha, \beta) d\alpha d\beta - C, \tag{3.36}$$

where S_+ denotes the region upon which the values of hysteresis operator $\hat{\gamma}_{\alpha\beta}$ is 1, $C = \iint_S \mu(\alpha, \beta) d\alpha d\beta$ is a constant. Note that $S_+ = \sum_{i=0}^{o-1} (-1)^i S_{p_i}$, where S_{p_i} represents the region encircled by the vertical and horizontal line pass the vertex p_i, $\overline{p_i d}$, $\overline{p_i f}$, and the straight line $\overline{ac}$ defined by $\alpha = \beta$. Thus, the output of the hysteresis operator can be written as

$$v(t) = \sum_{i=0}^{o-1} (-1)^i f(\alpha_{p_i}, \beta_{p_i}) - C, \tag{3.37}$$

where p_i, $i = 0, \ 1, \ldots, \ o-1$ are the vertexes corresponding to the remaining past local maxima and minima, p_o is the vertex corresponding to the current input with $\alpha_{p_o} = u(t)$ when o is even and $\beta_{p_o} = u(t)$ when o is odd, and the function $f(\cdot, \cdot)$ is defined as

$$f(\alpha_{p_i}, \beta_{p_i}) \triangleq \iint_{S_{p_i}} 2\mu(\alpha, \beta) d\alpha d\beta, \tag{3.38}$$

where $[\alpha_{p_i}, \beta_{p_i}]$ are the coordinates of any point $p_i \in S$ in the Preisach plane.

Next, the inverse Preisach model is established under the following assumption,

Assumption 19. *The weighting function $\mu(\alpha, \beta)$ of the Preisach model is positive, i.e., $\mu(\alpha, \beta) > 0$.*

This assumption only requires the branch of any given hysteresis loop to be composed by a monotone increasing or decreasing curve, which is satisfied in most applications. Under Assumption 1, the existence of the inverse of $f(\cdot,\cdot)$ in both directions is guaranteed. The inverse functions with respect to α and β are defined, respectively, as

$$G_1(\alpha, y) \triangleq f^{-1}(\alpha,\cdot)[y], \quad G_2(y,\beta) \triangleq f^{-1}(\cdot,\beta)[y]. \tag{3.39}$$

Then the inverse Preisach model is represented below,

$$u_r(t) = \begin{cases} G_1(\alpha_{p_o}, v(t) - \sum\limits_{i=0}^{o-1}(-1)^i f(\alpha_{p_i},\beta_{p_i}) + C), \text{ when } o \text{ is even;} \\ G_2(v(t) - \sum\limits_{i=0}^{o-1}(-1)^i f(\alpha_{p_i},\beta_{p_i}) + C, \beta_{p_o}), \text{ when } o \text{ is odd.} \end{cases} \tag{3.40}$$

The output of the above inverse Preisach model (3.40) serves as the reference input, $u_d(t)$, in the proposed B-spline based decomposition process below—to be matched by the synthesized input $u(t)$ (see Eq. (3.7)).

Inverse-Preisach-Model-based Decomposition via Uniform B-splines

We use the 3rd-order B-splines as an example to present the proposed realization– the same approach can be easily extended to other orders B-splines. To simplify the implementation, we consider that, at any given decomposition instant, only one sub-library is used,

$$u(t) = \sum_{k=0}^{M} p_k u_e(t - t_{d,k},\ \tau) = \sum_{k=0}^{M} h_k g_k u_e(t - t_{d,k},\ \tau). \tag{3.41}$$

Note that for the 3rd-order uniform B-splines,

$$u_e[k-j] = 0, \quad k - j \neq -1,\ 0,\ 1, \tag{3.42}$$

where $u_e[k] \triangleq u_e(k\Delta T, \tau)$. Thus, at any given decomposition knot $t_{d,k} = t_c + k\Delta T$, $k = 0,\ 1, \dots,\ M$, only three output elements are involved (see Fig. 3.2). Then, in the interval $T_k \triangleq [t_{d,k}, t_{d,k} + l\Delta T]$, matching the summed (synthesized) input in Eq. (3.41) with the inverse Preisach model based input $u_d(t)$ in Eq. (3.40) at the decomposition knots $k,\ k+1, \dots,\ k+l-1$, for $k \geq 1$, $l \geq 2$, $k + l \leq M$, and $k \in \mathbb{N},\ l \in \mathbb{N}$, yields

$$\mathbf{A}_l \mathbf{H}_l = \mathbf{U}_{d,l}, \tag{3.43}$$

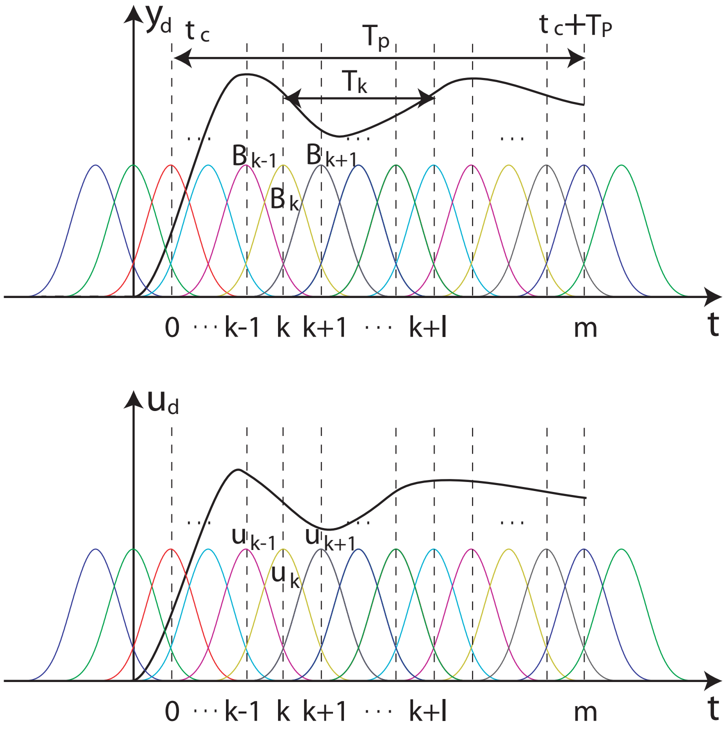

Figure 3.2: The B-spline decomposition and input synthesis process.

where

$$\mathbf{A}_l = \begin{pmatrix} g_k & c_q g_{k+1} & & & \\ c_p g_k & g_{k+1} & c_q g_{k+2} & & \\ & \ddots & \ddots & \ddots & \\ & & c_p g_{k+l-3} & g_{k+l-2} & c_q g_{k+l-1} \\ & & & c_p g_{k+l-2} & g_{k+l-1} \end{pmatrix}, \tag{3.44}$$

$$\mathbf{H}_l = \begin{pmatrix} h_k & h_{k+1} & \ldots & h_{k+l-2} & h_{k+l-1} \end{pmatrix}^\top, \tag{3.45}$$

and

$$\mathbf{U}_{d,l} = \begin{pmatrix} \bar{u}_d[k] - h_{k-1} g_{k-1} c_p \\ \bar{u}_d[k+1] \\ \vdots \\ \bar{u}_d[k+l-2] \\ \bar{u}_d[k+l-1] - h_{k+l} g_{k+l} c_q \end{pmatrix}, \tag{3.46}$$

with $\bar{u}_d[k] = u_d[k]/u_e[0]$, $u_d[k] \triangleq u_d(t_{d,k})$, $c_p \triangleq u_e[-1]/u_e[0]$ and $c_q \triangleq u_e[1]/u_e[0]$. Note in Eqs. (3.44, 3.46), $0 < c_p < 1$, $0 < c_q < 1$ (as $u_e[0]$ is the maxima of the input element).

By Lemma 13, the coefficient vector $\mathbf{H}_l$ exists as the copies of time-shifting B-splines satisfy Assumption 11, and the inverse Preisach model, the desired output $v_d(t)$, and thereby, the inverse input $u_d(t)$ and the synthesized input $u(t)$, by Assumptions 8 and 9, are all Lipschitz. However, as the first and the last elements in $\mathbf{U}_{d,l}$ depend on the decomposition coefficients h_{k-1} and h_{k+l} that are both outside the decomposition period, the exact solution of $\mathbf{H}_l$ requires the entire desired trajectory to be known a priori—not feasible in preview-based tracking. Thus, we propose to obtain the compensation factors h_ks successively via a moving horizon, and show that the truncation-caused error can be rendered arbitrarily small by choosing a small enough knot period ΔT.

As $\det \mathbf{A}_l = \prod_{i=1}^{l} G_i g_{k+i+1}$ with $G_i = 1 - c_p c_q / G_{i-1}$ when $i = 2,\ 3, \ldots,\ l$ and $G_1 = 1$, g_ks and G_is are all non-zero, $\det \mathbf{A}_l \neq 0$, $\mathbf{A}_l$ is invertible, the solution for the compensation factor vector is readily given by

$$\mathbf{H}_l = \mathbf{A}_l^{-1} \mathbf{U}_{d,l}. \tag{3.47}$$

In the proposed approach, only the first element of the matrix $\mathbf{H}_l$, h_k, is computed, the rest of the compensation factors, h_k, for $k = k+1,\ k+2, \ldots,\ k+l-1$ are computed in a moving horizon manner, i.e., h_{k+1} is obtained by moving the truncation horizon one step forward (the horizon can always be moved in the preview window as $2l\Delta T \ll T_p$ in practice). To find h_k, obtain the first row of $\mathbf{A}_l^{-1}$ as

$$(\mathbf{A}_l^{-1})_{1,s} = (-1)^{1+s} c_p^{s-1} \prod_{i=1}^{s-1} g_{k+i} \frac{\det \phi_{i+1}}{\det A_l}, \tag{3.48}$$

where $\phi_{i+1} = A_l(i+1 : l, i+1 : l)$, and

$$\det \phi_{i+1} = \prod_{j=i+1}^{l-1} G_{j-i} g_{k+j}. \tag{3.49}$$

and the first element in the matrix $\mathbf{H}_l$ as

$$\begin{aligned} h_k = {} & \frac{1}{g_k G_l} (\bar{u}_d[k] - h_{k-1} g_{k-1} c_p) + \sum_{i=2}^{l-1} \frac{(-1)^{(1+i)} c_p^{i-1}}{g_k \prod_{k=l-i+1}^{l} G_k} \bar{u}_d[k+i-1] \\ & + \frac{(-1)^{(1+l)} c_p^{l-1}}{g_k \prod_{k=1}^{l} G_k} (\bar{u}_d[k+l-1] - h_{k+l} g_{k+l} c_q). \end{aligned} \tag{3.50}$$

The above Eq. (3.50) implies that the influence of the input element decays exponentially as the coefficient $c_p \in (0,\ 1)$, and thereby, can be truncated to l step with arbitrary precision by having a small enough knot period ΔT,

$$h_k = \frac{1}{g_k G_l}(\bar{u}_d[k] - h_{k-1}g_{k-1}c_p) + \sum_{i=2}^{l} \frac{(-1)^{(1+i)}c_p^{i-1}}{g_k \prod_{k=l-i+1}^{l} G_k}\bar{u}_d[k+i-1]. \tag{3.51}$$

Lemma 20. *Let Assumption 18 hold, and at the k^{th} decomposition knot, let the compensation factor h_k be obtained with l step-truncation given by Eq. (3.51), then there exists a $l_0 \in \mathbb{N}$ such that for any given $\epsilon > 0$, the truncation-caused error $|e_k| \le \epsilon$ for any given $l > l_0$.*

The proof is omitted to save space.

Lemma 21. *Let Assumption 18 hold, the decomposition knot period ΔT be given by Eq. (3.33), and the element amplitude be discretized as in Eq. (3.29), then the input superposition error $e_u(t)$ can be bounded as*

$$|e_u(t)| \le L_h L_v \Delta T, \quad t \in [t_c,\ t_c + T_p]. \tag{3.52}$$

Based on the congruency property of the Preisach model [59], the number of output elements in the decomposition-synthesis process (i.e., M^* in Eq. (3.8)) for the Non-strict ASHS to hold can be further quantified. The Lipschitz constants in Assumption 8 for hysteresis system is range-dependent, i.e.,

$$L_h = L_h(\chi), \tag{3.53}$$

where χ is the displacement range of a hysteresis loop, i.e, χ is the difference between two consecutive local extremas of the output $v(t)$.

Lemma 22. *Let Assumptions 8, 9, 11 hold, for any given $\epsilon > 0$, and output trajectory with finite number of local extremas at the time instants $t_c = t_1^* < t_2^* < \ldots < t_N^* = t_c + T_p$, then there exists $M \ge M^*$ output elements $v_1(t),\ v_2(t), \ldots,\ v_M(t)$ and its corresponding input elements $u_k(t)$s, such that the Non-strict ASHS holds for system (3.24) for any $M \ge M^*$, where*

$$M^* \triangleq 2\sum_{i=1}^{N-1} \left\lceil L_h(\chi_i)L_v(t_{i+1}^* - t_i^*)/\epsilon \right\rceil.$$

Proof. The Lemma can be shown by applying Lemma 13 to each monotonic session of the input $u_d(t)$, and finding the corresponding M_i^* for any given $\epsilon > 0$ as given in Eq. (3.10), and finally, adding the needed number of elements in each session together. □

3.3.3 Optimization for Implementation

Two optimizations are proposed to enhance practical implementations of the proposed ASHS for output tracking.

Optimization of the Compensation Factors

We first optimize the selection of the input-output elements to maximize the hysteresis compensation at the element mapping level, i.e., we aim to select the input-output element of optimal amplitude at which the compensation factor $h_k \to 1$ in Eq. (3.26). Specifically, the optimal input/output amplitude factor, τ^*, is obtained by minimizing the superposition-caused input error with respect to τ at the decomposition knots for $h_k = 1$ with $k = -1,\ 0,\ \ldots,\ m+1$, i.e.,

$$\begin{aligned}\min_{\tau} \mathcal{J}(\tau) &= \min_{\tau} \sum_{k=0}^{m} \left(\sum_{j=-1}^{m+1} \hat{g}_j/\tau u_e((k-j)\Delta T,\ \tau) - u_d(k\Delta T) \right)^2, \\ &= \min_{\tau} [\mathbf{G}\mathbf{U}_e(\tau) - \mathbf{U}_d]^\top [\mathbf{G}\mathbf{U}_e(\tau) - \mathbf{U}_d],\end{aligned} \tag{3.54}$$

where $\hat{g}_j$s are the decomposition coefficients when the base output elements $\hat{v}_e(t)$s in Eq. (3.30) are utilized in the output decomposition,

$$\mathbf{G} = \begin{pmatrix} \hat{g}_{-1} & \hat{g}_0 & \hat{g}_1 \\ \hat{g}_0 & \hat{g}_1 & \hat{g}_2 \\ \vdots & \vdots & \vdots \\ \hat{g}_{m-1} & \hat{g}_m & \hat{g}_{m+1} \end{pmatrix}, \quad \text{and} \tag{3.55}$$

$$\mathbf{U}_e(\tau) = [u_e(\Delta T,\ \tau)\ u_e(0,\ \tau)\ u_e(-\Delta T,\ \tau)]^\top/\tau, \tag{3.56}$$

and

$$\mathbf{U}_d = [u_d(0)\ u_d(\Delta T) \ldots\ u_d(m\Delta T)]^\top. \tag{3.57}$$

Note that in the above Eq. (3.54), we employed the fact that $u_e((k-j)\Delta T,\ \tau) = 0$ if $|k-j| \geq 2$. The solution to Eq. (3.54) can be obtained by setting

$$\frac{\partial \mathcal{J}(\tau)}{\partial \tau} = [\mathbf{G}\mathbf{U}_e^*(\tau) - \mathbf{U}_d]^\top \mathbf{G} \frac{\partial \mathbf{U}_e(\tau)}{\partial \tau} = 0, \tag{3.58}$$

Solution to Eq. (3.58) depends on the expression of $\mathbf{U}_e(\tau)$ that can be established experimentally. In practice, we can assume that the input elements with different amplitude are linearly dependent to each other, i.e.,

$$\mathbf{U}_e(\tau) = \frac{\varphi(\tau)}{\tau\varphi(1)}\hat{\mathbf{U}}_e, \tag{3.59}$$

with $\hat{\mathbf{U}}_e = [\hat{u}_e(\Delta T)\ \hat{u}_e(0)\ \hat{u}_e(-\Delta T)]^\top$ and

$$\varphi(\tau) = \sup_t |u_e(t,\ \tau)|. \tag{3.60}$$

Then the solution is obtained as

$$\frac{\varphi(\tau_1^*)}{\tau_1^*} = \varphi'(\tau_1^*), \tag{3.61}$$

or

$$\frac{\varphi(\tau_2^*)}{\tau_2^*} = -\frac{\varphi(1)\mathbf{U}_d^\top \mathbf{G}\hat{\mathbf{U}}_e}{\hat{\mathbf{U}}_e^\top \mathbf{G}^\top \mathbf{G}\hat{\mathbf{U}}_e}. \tag{3.62}$$

and the optimal value of τ^* is given as

$$\tau^* = \arg\min_{\tau\in\Theta} \mathcal{J}(\tau), \quad \text{with } \Theta = \{\tau_{\min},\ \tau_1^*,\ \tau_2^*,\ \tau_{\max}\}.$$

Online Adjustment of the Decomposition Coefficients

We present the online optimization of the compensation factors h_ks to account for uncertainties and disturbances by using the 3rd-order uniform B-splines as an example. For 3rd-order uniform B-splines, at any given time instant, there are four output elements involved in the decomposition at any time instant, i.e.,

$$v_d(t) = \sum_{j=k-1}^{k+2} \hat{g}_j v_e(t,\ \tau)/\tau, \quad t \in I_k, \tag{3.63}$$

with $I_k = [k\Delta T,\ (k+1)\Delta T]$, and the corresponding synthesized input is then updated online by using correction factors λ_j for $j = k-1,\ k,\ k+1,\ k+2$ as

$$u(t) = \sum_{j=k-1}^{k+2} (h_j\hat{g}_j + \alpha\lambda_j)u_e(t,\ \tau)/\tau, \quad t \in I_k, \tag{3.64}$$

where $\alpha \in \Re$ is the DC-gain of the system (can be easily determined through experiment). The optimal sequence of adjusting factors,

$$\mathbf{\Lambda}^* = [\lambda_{k-1}\ \lambda_k\ \lambda_{k+1}\ \lambda_{k+2}]^T,$$

can be obtained by minimizing the $\mathcal{L}_2$ norm of the predicted tracking error,

$$\min_{\mathbf{\Lambda}} \|\mathbf{Y}(t)^T \mathbf{\Lambda} - \Omega(t, e(t))\|_{\mathcal{L}_2}, \quad t \in I_k, \tag{3.65}$$

where $\|f(\cdot)\|_{\mathcal{L}_2}$ is the $\mathcal{L}_2$ norm restricted on I_k, $\mathbf{Y}(t) = [v_e(t - t_{d,k-1},\ \tau)\ v_e(t - t_{d,k},\ \tau)\ v_e(t - t_{d,k+1},\ \tau)\ v_e(t - t_{d,k+2},\ \tau)]^T$, $e(t)$ is the tracking error, and $\Omega(t, e(t))$ is an innovation function to predict the tracking error $e(t)$ for $t \in I_k$,

$$\Omega(t, e(t)) = e(t - \Delta T) + \kappa \Omega(t - \Delta T, e(t - \Delta T)),$$

where $\kappa < 1$ is the forgetting factor and $\Omega(t, e(t)) = 0$, for $t \in [0,\ \Delta T]$.

The solution to the above least-mean square minimization problem (3.65) can be readily obtained as

$$\Lambda^* = \left(\int_{I_k} \mathbf{Y}(t)\mathbf{Y}(t)^T dt \right)^{-1} \int_{I_k} \Omega(t, e(t))\mathbf{Y}(t) dt. \tag{3.66}$$

As each element in $\mathbf{Y}(t)$, corresponding to the value of output elements, is always selected to be positive, the first term in Eq. (3.66) is guaranteed to be positive definite, thereby, invertible.

3.4 Application: Decomposition-Learning based Simultaneous Hysteresis-Dynamics Compensation

We demonstrate one application of the proposed ASHS: Simultaneous hysteresis and dynamics compensation in preview-based output tracking for smart actuators such as the piezoelectric actuator. The corresponding input-output mapping of the system, $u(t) \to v(t)$ is given as

$$y(t) = \mathbf{G}[v(t)], \quad \text{with } v(t) = \mathbf{H}[u(t)], \tag{3.67}$$

and we assume that

Assumption 23. *The linear operator* $\mathbf{G}[\cdot]$ *is stabilizable, detectable, and hyperbolic with a well-defined relative degree* r.

The above assumption is utilized to facilitate practical implementation of the ASHS. Specifically, the stability condition is needed for practical implementations of ILC. The hyperbolic condition, i.e., the linear dynamics part of the system has no pure imaginary zeros, and the detectability condition are needed, based on the stable inversion theory [38, 44], to guarantee that the pre- and post- actuation time for exact tracking of a given trajectory exponentially decays to zero, and can be truncated to a finite time [26]. All these conditions in Assumption 23 can be verified via experiments.

The output tracking problem now is to find a synthesized input $u(t)$ in the form of Eq. (3.6) to track the desired output of the whole system (3.67), $y_d(t)$. We propose a two-step scheme: First, the superposition principle is employed to account for the dynamics part of system (3.67) by finding the intermediate output $v_d(t)$ as [26]

$$v_d(t) = \sum_{k=0}^{n} g_k \bar{v}_{e,i_k}(t - t_{d,k}, s), \tag{3.68}$$

with the decomposition coefficients g_ks given by the decomposition of the desired trajectory $y_d(t)$, i.e.,

$$y_d(t) = \sum_{k=0}^{n} g_k \bar{y}_{e,i_k}(t - t_{d,k}, s), \tag{3.69}$$

where $(\bar{y}_{e,i_k}(t,s), \bar{v}_{e,i_k}(t,s))$ are the input-output pairs selected from the library constructed for dynamics compensation,

$$\mathcal{L}_D = \{(\bar{y}_{e,i}(\cdot,\ s),\ \bar{v}_{e,i}(\cdot,\ s)) :\ i = 1,\ 2,\ \ldots,\ N_D,\ s \in \Re^+\}, \tag{3.70}$$

with $\bar{y}_{e,i}(t,\ s) = \hat{y}_e(t/s)$, where $\hat{y}_e(t)$s are output elements at a pre-chosen speed, i.e., s is the speed factor of the element, and N_D is the number of elements at the same speed. Thus, tracking of the intermediate input $v_d(\cdot)$ leads to the tracking of the desired output $y_d(\cdot)$.

Then secondly, the Non-strict ASHS proposed in Sec. 3.3 is employed to account for the hysteresis part of system (3.67) by finding the synthesized input $u_d(t)$, such that the required tracking precision of the desired intermediate output $v_d(t)$ is achieved.

The first step–finding the intermediate output $v_d(\cdot)$ via the superposition principle in

Eqs. (3.68, 3.69)–has been addressed recently [26]. Particularly, note that finite pre- and post- actuation (i.e., truncation) is needed when applying the synthesized input, i.e., when using the synthesized intermediate output $v_d(\cdot)$ in Eq. (3.68) as the desired output in the second step. The needed finite pre- and post- actuation time for achieving the desired tracking precision has been quantified– readers are referred to Ref. [26] for the details.

Note that the output elements $v_{e,i}(\cdot,\tau)$s in (3.28) can be chosen differently from those $\bar{v}_{e,i}(\cdot,s)$s in (3.70) to facilitate the decomposition, i.e., the original basis $\bar{v}_{e,i}(\cdot,s)$s in Eq.(3.70) have been convoluted with the dynamics of system (3.67) (in order to compensate for the dynamics of system (3.67)), thereby may not have the desired properties for hysteresis compensation in the second step.

Implementation of the above two-step approach requires decoupling dynamics and hysteresis compensation in both offline library construction and online decomposition-synthesis process. We propose a temporal and spatial scale scheme based on, respectively, the range-dependence of the hysteresis and the rate-dependence of the dynamics effects. Specifically, during offline library construction, the dynamics-compensation library L_D can be obtained by keeping the amplitude of output elements $\bar{y}_{e,i}(\cdot,s)$s small enough, and the hysteresis-compensation libraries $L_{H,i}$s can be obtained by keeping the speed of the intermediate output elements $v_{e,i}(\cdot,\tau)$s slow enough. Moreover, different speeds can be applied to the output elements $\bar{y}_{e,i}(\cdot,s)$s and different amplitude to the intermediate output elements $v_{e,i}(\cdot,\tau)$s, respectively, such that in online tracking, the elements with different speed factors s_js in library $\mathcal{L}_D$ can be utilized to compensate for the dynamics in different frequency ranges, and elements of different amplitude coefficients τ_js in sub-library $L_{H,i}$s can be utilized to compensate for the hysteresis in different amplitude ranges, respectively.

During online implementation, the previewed part of the desired output $y_d(t)$ is spatially scaled down first as:

$$y_d(t) \longrightarrow \sigma y_d(t), \quad \text{for } t \in [t_c, t_c + T_p], \tag{3.71}$$

such that for a pre-chosen amplitude threshold value M_H(i.e., below which the hysteresis effect is negligible for the desired tracking precision),

$$\sigma \sup_{t\in[t_c,t_c+T_p]} |y_d(t)| \leq M_H, \tag{3.72}$$

where $\sigma \in (0,1)$ is a pre-chosen constant, and both σ and M_H can be easily determined in practices. Then the scaled-down previewed desired output $\sigma y_d(t)$ is taken as the desired output to obtain the desired pre-intermediate output $\bar{v}_d(t)$ and the corresponding decomposition coefficients $\bar{g}_k$s, i.e., replace y_d, v_d and g_k in Eqs. (3.68, 3.69) with $\bar{y}_d$, $\bar{v}_d$ and $\bar{g}_k$, respectively, and the real desired intermediate output and corresponding decomposition coefficients g_ks are obtained as

$$v_d(t) = \bar{v}_d(t)/\sigma, \quad \text{and } g_k = \bar{g}_k/\sigma. \tag{3.73}$$

Then secondly, the above desired intermediate output $v_d(t)$ is temporally scaled (slowed) down by using a time-scale factor η,

$$v_d(t) \longrightarrow v_d(t/\eta), \quad \text{for } t \in [t_c, t_c + T_p], \tag{3.74}$$

such that the dynamics effect of the system can be ignored. The corresponding input $u_d(t/\eta)$ as in

$$v_d(t/\eta) = \mathbf{H}[u_d(t/\eta)] \tag{3.75}$$

is obtained by using the proposed hysteresis-compensation decomposition approach in Sec. 3.3.2, and Sec. 3.3.2. Finally, the obtained input $u_d(t/\eta)$ is scaled back in time domain to obtain the control input $u(t)$ for implementation.

Library Construction To construct the libraries $\mathcal{L}_{H,i}$s and $\mathcal{L}_D$ a priori, the inversion-based iterative learning control (IIC) technique [19] is used to obtain the corresponding input elements. As an example, we present below the inversion-based iterative learning control (IIC) [34],

$$\hat{U}_k(j\omega) = \hat{U}_{k-1}(j\omega) + \rho(j\omega)G^{-1}(j\omega)(\hat{Y}_d(j\omega) - \hat{Y}_{k-1}(j\omega)), \tag{3.76}$$

where for $k \in \mathbb{N}^+$, $\hat{U}_k(j\omega)$ and $\hat{Y}_k(j\omega)$ are the Fourier transform of the kth iteration input and output, respectively, with $\hat{U}_0(j\omega) = 0$ and $\hat{Y}_0(j\omega) = 0$ initially, $\rho(j\omega)$ is the weighting coefficient, $\hat{Y}_d(j\omega)$ is the Fourier transform of the desired trajectory, and $G(j\omega)$ is the transfer function model of the system.

It has been shown and demonstrated in experiments that the IIC techniques can account for both hysteresis and dynamics in piezoelectric actuators [34], achieving high-precision tracking. Moreover, the effect of model uncertainties and/or system variations can be largely removed by simply updating the decomposition library via IIC.

3.5 Experimental Implementation

3.5.1 Experiment Setup

We demonstrate the proposed ASHS approach for output tracking by implementing it to the trajectory tracking of a piezoactuator for lateral scanning (x,y direction) on an atomic force microscope (Dimension-ICON, Bruker Nano. Inc.). The first resonant frequency and the bandwidth of the piezoactuator were at $\sim$ 650 Hz and $\sim$ 450 Hz, respectively, with full displacement range around 72 μm. The control algorithm was designed and implemented in the MATLAB-xPC-target (MathWorks, Inc.) via a data acquisition system (PCI-6259, National Instrument Inc.).

3.5.2 Implementation of the Proposed ASHS Approach

Library Construction

The 3rd-order uniform B-spline of 20 different amplitudes were used to construct the hysteresis-compensation library $\mathcal{L}_H$ of 20 sub-libraries, $\mathcal{L}_{H,i}$ with $i = 1,\ 2, \cdots, 20$. With a time duration of over 30 seconds–compared to the time constant of the piezoactuator at 0.5 second, the B-splines were slow enough to avoid exciting the dynamics of the piezoactuators, and the IIC technique above was utilized to obtain the corresponding input elements, as shown in Fig. 3.3(a). The function $\varphi(\tau)$ (see Eq. (3.60)) for optimizing the output element amplitude in Sec. 3.3.3 was obtained experimentally as shown in Fig. 3.3(b).

For dynamics compensation in different frequency ranges, a library $\mathcal{L}_D$ consisting of 3

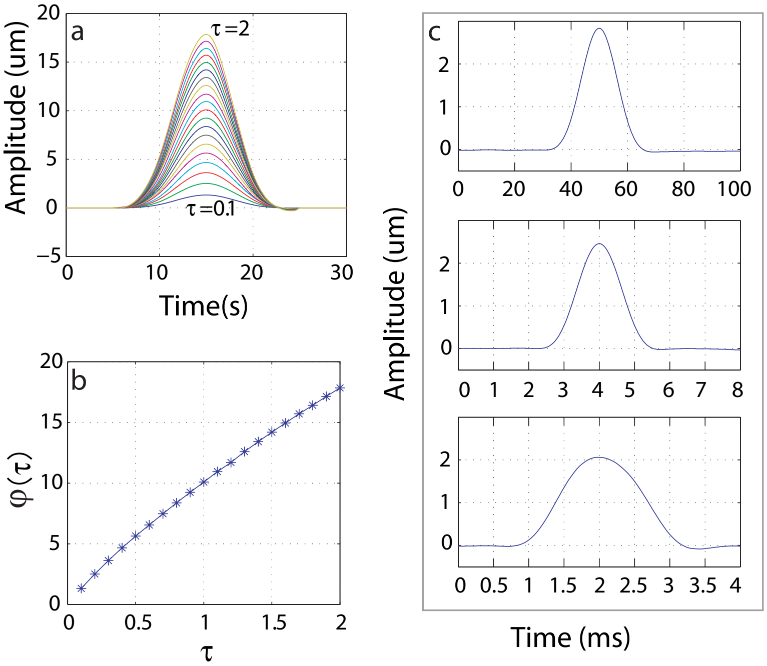

Figure 3.3: (a) The input elements for the uniform B-spline based outputs at 20 different amplitudes; (b) The $\varphi(\tau)$ function, and (c) The input elements for the uniform B-spline based outputs at 3 different speeds.

different speeds ($N_D = 3$) was constructed in the similar way, as shown in Fig. 3.3(c), where the amplitude was kept at $2\mu m$ (2.8% of the total displacement range), so that the hysteresis effect can be ignored.

Inverse Preisach Modeling of the Hysteresis

The weighting-integration function $f(\cdot,\cdot)$ of the inverse Preisach model, as shown in Fig. 3.4 was identified by using a sinusoidal wave with decaying amplitude as an excitation signal, after the local memory effect had been removed by driving the piezoactuator to the full range. The model was validated by using a sinusoidal wave of varying amplitude as the input, and comparing the experimentally measured output to the model-predicted one, as shown in Fig. 3.5(a). The major hysteresis loop presented in Fig. 3.5(b) demonstrates that the hysteresis effect was, indeed, significant in this piezoactuator system. Monotonic in both directions, the identified weighting-integration function $f(\cdot,\cdot)$ was then employed online at each decomposition instant t_c to

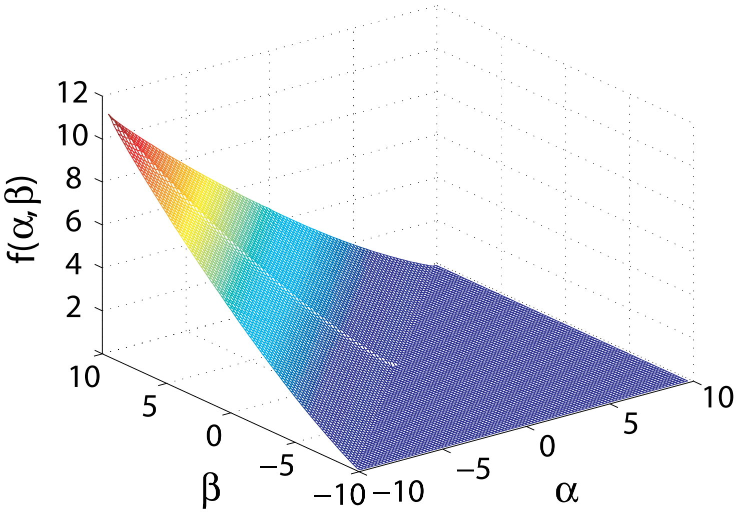

Figure 3.4: The weighting-integration function $f(\cdot,\cdot)$ of the Preisach model identified through experiment.

obtain the values of the reference input $u_d(t)$ at the decomposition knots in the interval $[t_c, t_c + T_p]$.

Online Output Decomposition, Input Synthesis and Optimization

To demonstrate the proposed approach for simultaneous hysteresis-dynamics compensation, a trajectory with peak-to-peak amplitude of $67.6\mu m$ (94% of the total displacement range) at three different speeds were chosen as the desired trajectory. Only a finite preview of the desired trajectory with preview time of $T_p = 500,\ 40$ and 20 ms, respectively, was used in the tracking at the low, medium, and high speeds, respectively.

In online implementation, first, during each decomposition period, the corresponding portion of the previewed desired trajectory $y_d(t)$ was spatially scaled down by 20 times and then decomposed by using the dynamics-compensation library $\mathcal{L}_D$ via linear superposition to obtain the intermediate output decomposition coefficients $\bar{g}_k$s (see Eqs. (3.68, 3.69)) and the intermediate input $v_d(t)$ (after scaling the signal back); Secondly, the intermediate output $v_d(t)$ was slowed down by 210, 2625, and 5250 times

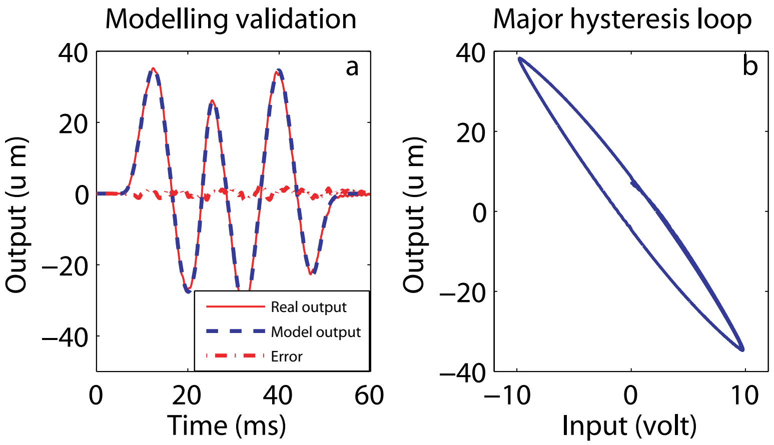

Figure 3.5: (a)Validation of the inverse Preisach model ; (b) The major hysteresis loop of x-axis piezoactuator.

for the low, medium, and high speed, respectively, and then decomposed by using the hysteresis-compensation library $\mathcal{L}_H$. Finally, the compensation factors h_ks was obtained by Eq. (3.51), and then online updated via Eq. (3.66), and then combined with the intermediate decomposition coefficients g_ks to obtain the control input.

3.5.3 Tracking Results and Discussion

For comparison, tracking by using a well-tuned PI controller and the DC-gain method (where the control input was obtained by simply scaling the desired output by the DC-gain of the system) were also conducted in the experiment, as shown in Figs. 3.6-3.8 for the low, medium and high speed, respectively. The comparison of the open-loop to the closed-loop frequency response in Fig. 3.9 showed that with the PI-controller, the bandwidth of the system was well maintained while the resonant peak was removed. The corresponding relative RMS tracking errors were compared in Table. 5.1.

The experimental results clearly demonstrated the performance of the proposed method in compensating for both the dynamics and the hysteresis effectively in achieving accurate tracking, particularly, during high-speed, large-amplitude tracking. At low speed (where the hysteresis effect was pronounced, see Fig. 3.6, in which the tracking error of the DC-gain clearly showed the hysteresis effect), the hysteresis effect was almost

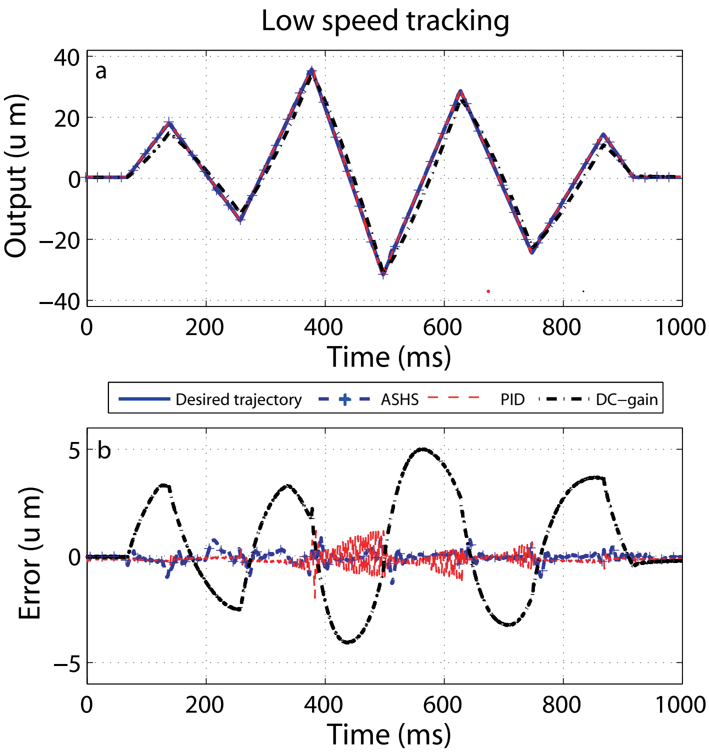

Figure 3.6: Comparison of (a) the tracking results by using the proposed ASHS approach to those obtained by using a well-tuned PI controller and the DC-gain method, and (b) the corresponding tracking error for low-speed tracking.

completely removed by both the proposed method and the PI feedback. However, in medium and high speed tracking, where both the hysteresis and the vibrational dynamics effects were pronounced, the tracking error of the DC-gain method clearly showed the combined hysteresis-dynamics effects (see Fig. 3.7 and Fig. 3.8). The tracking of the PI-feedback control degraded as the spectrum of the desired output approached to the bandwidth of the closed-loop system, whereas with the proposed technique, precision tracking of only 3∼4% RMS tracking error was still maintained (see Table 5.1).

Table 3.1: The relative RMS tracking error of the three methods

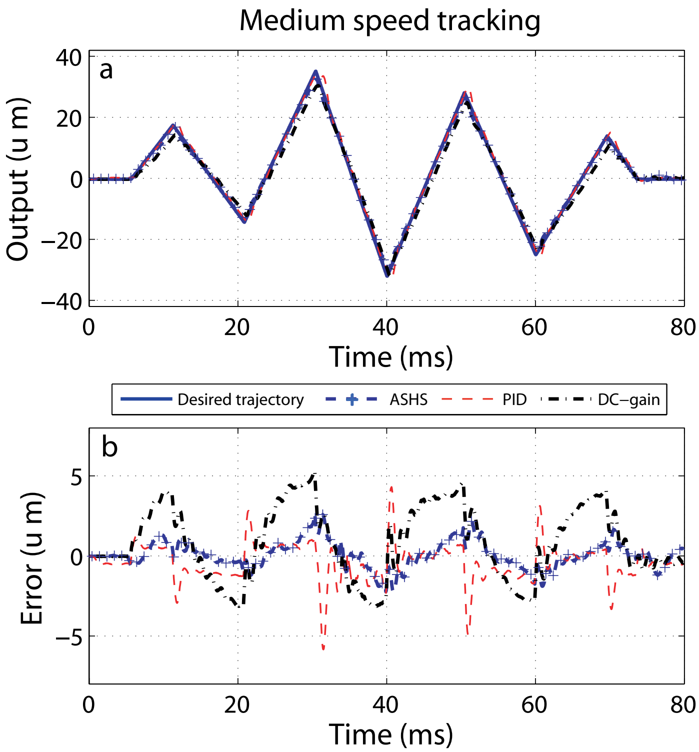

Figure 3.7: Comparison of (a) the tracking results by using the proposed ASHS approach to those obtained by using a well-tuned PI controller and the DC-gain method, and (b) the corresponding tracking error for medium-speed tracking.

	Low (%)	Medium (%)	High (%)
ASHS	1.84	3.35	3.96
PI	2.12	10.34	22.79
DC-gain	18.32	17.28	26.59

The experimental results also demonstrated the efficacy of the proposed approach in exploiting the benefits of offline learning to avoid the complexity in modeling, controller design and implementation in output tracking of hysteresis-Hammerstein systems. For example, in the high-speed tracking with preview time $T_p = 20$ ms (half of the total time duration), only 20 input-output elements were used in the decomposition-synthesis process, and the online computation was distributed to ∼100 sampling periods. In each sampling period, the input synthesis process only amounted to a couple of simple multiplications and additions. Moreover, quasi-static uncertainties in day-to-day operation,

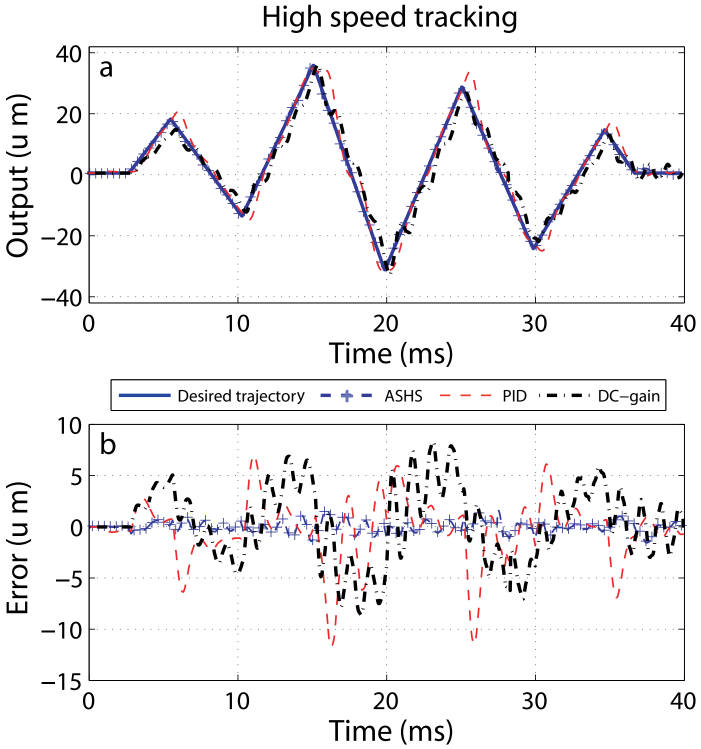

Figure 3.8: Comparison of (a) the tracking results by using the proposed ASHS approach to those obtained by using a well-tuned PI controller and the DC-gain method, and (b) the corresponding tracking error for high-speed tracking.

such as day to day variation of the piezoactuator, can be accounted for with minor to no loss of tracking performance by updating the input-output libraries $\mathcal{L}_H$, $\mathcal{L}_D$ and a simple weighting-integration function $f(\cdot,\cdot)$ of the Preisach model regularly through a straightforward offline process.

3.6 Conclusion

In this chapter, the superposition of Hammerstein systems has been studied through the development of the almost superposition principle of Hammerstein systems. It has been shown that the superposition error can be rendered arbitrarily small, i.e.,

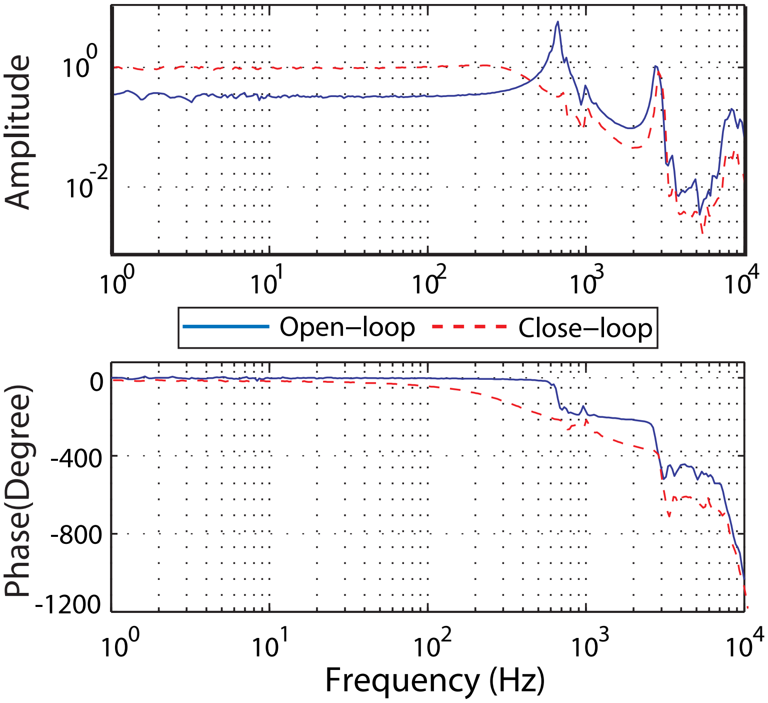

Figure 3.9: (a) The Bode plot of the x-axis open-loop response; (b) The Bode plot of the x-axis PI feedback close-loop response.

the almost superposition holds for Hammerstein systems, provided that the number of output elements is large enough. A realization of the ASHS for output tracking has been proposed by combining uniform B-splines for trajectory decomposition with an inverse Preisach model for input synthesis. Two optimization schemes are further proposed to enhance the implementation of the ASHS in practice. Moreover, the proposed ASHS has been implemented for simultaneous compensation of hysteresis and dynamics in precision output tracking. Experiments were performed to control the piezoactuator on an AFM system in different frequency range, and the results demonstrated the efficacy of the proposed method.

Chapter 4

On Single-basis Online Trajectory Decomposition for Control Applications

Abstract

In this chapter, the problem of decomposing a trajectory online by using single basis function (and its time-shifted copies) is considered for control applications. Trajectory decomposition has been explored in control areas including fuzzy and neural-network control, system identification, and learning control, where a given trajectory is decomposed (approximated) by using a set of basis functions. The existing decomposition methods, however, are limited to special applications (e.g., contour tracking in machining), and/or requires truncation of the basis functions. Although these limitations might be alleviated by, for example, using additional basis functions, the complexity in implementations is inevitably increased. This work aims to achieve online asymptotic decomposition of a partially-known trajectory by using only one basis function and its time-shifted copies without truncation. First, we consider the problem as a least-square minimization problem, then truncation of the basis functions in the decomposition is resolved through a zero-period extension of the trajectory (i.e., extending the beginning of the trajectory at its initial value for a finite period). It is shown that as the length of the extended period increases, the decomposition coefficients at the beginning portion of the extended period approach to zero. Finally, a sectional interactive decomposition algorithm is proposed for online trajectory decomposition. Numerical example by using B-spline as the basis function is presented to demonstrate the proposed decomposition approach.

4.1 Introduction

In this chapter, the problem of decomposing a trajectory using only single basis function and its time-shifted copies is considered for control applications. Trajectory decomposition has been explored in areas including data regression [62, 63], signal analysis [64], and dynamics and control [65, 26, 66]. Particularly, in control areas, trajectory decomposition has been utilized in fuzzy-neural network control to approximate/interpolate the input-output mapping [66], in system identification to approximate the linear parameter-varying systems [65], and in iterative learning control (ILC) to extend ILC to non-repetitive operations [26]. In these areas, it is very much desired to achieve accurate decomposition by using only few types of basis functions (as few as, possibly, only one type), and without intensive computation. The existing methods, however, not only involve truncations of the basis functions and extraneous steps to address truncation-caused issues, but also demand heavy computations, thereby, not suitable for online implementations. These needs in trajectory decomposition and the challenges involved motivate this work.

Limitations exist in current techniques for online trajectory decomposition. For example, although precision trajectory decomposition can be obtained by using B-splines as the basis functions [21, 22], these methods do not account for control objectives such as high-order smoothness of the trajectory and ease of obtaining the control input for precision output tracking. These control-oriented objectives can be taken into account by integrating the decomposition technique with iterative learning control (ILC). For example, a task-based basis function method is employed to extend ILC to micro-robotic deposition [3]. However, this technique is limited to specific types of trajectories with limited variations (e.g., sinusoidal and/or triangle waves, and concatenations of these trajectories). To broaden the applications of ILCs, model-based rational basis functions have been proposed for tracking more general trajectories through extrapolation [23, 25]. Alternatively, multiple basis functions each corresponding to a different basic state of a linear time variant system can be identified through ILC, and inputs for other states of the system can then be obtained via interpolation [24]. However, the number of the types of basis functions used are increased significantly, rendering the learning processing time consuming and the computation complicated. The complexity caused by the large number of basis function types (used in the input-output decomposition) can be alleviated by using uniform B-splines based basis functions [26]. All these methods, however, require multiple types basis functions and, when the trajectory is only partially known, truncation of the basis functions at the boundary of the decomposition period [26], thereby, increasing the complexities in the applications.

In existing trajectory decomposition methods, truncation of the basis functions in the decomposition has resulted in extraneous difficulties and complexities. For example, to exploit the advantage of iterative learning control (ILC) in attaining both high-performance and robustness (against quasi-static and slow system dynamics variations) in high-speed tracking, it has been proposed to combine online decomposition of the partially known (i.e., finite previewed) desired trajectory via B-spline-based basis functions with ILC based offline tracking of the basis functions via ILC for general output tracking of linear dynamic systems [26]. Such an idea of offline-learning, and online-decomposition-synthesis has also been extended to Hammerstein systems [67]. Truncation of basis functions is needed to decompose partially-known (finite-previewed) of desired trajectory that, in general, does not start and/or end at zero. However, it is rather difficult (if not completely impossible) to track such truncated basis functions without inducing pronounced transient tracking error [68]. Although the discontinuous-beginning-end-caused tracking error can be addressed by smoothly extending and transiting the truncated basis functions to zero, extra basis functions are induced, resulting in not only extra work (in redesigning the new basis function), but also additional types of basis functions and possibly, extraneous tracking error, complicating the decomposition procedure. Therefore, there are needs to study online trajectory decomposition by using minimal types of basis functions without truncation.

The main contribution of this work is the development of an asymptotic trajectory decomposition technique by only using single type of basis functions without truncation. First, the problem of trajectory decomposition using single basis function (and its time-shifted copies) is considered via least-square minimization. It is shown that arbitrary decomposition precision can be attained by using enough number of single-type basis functions. Then the issue of truncating basis functions in the decomposition is resolved via a zero-period extension of the trajectory (i.e., by extending the beginning of the trajectory at its initial value for a finite period). The decomposition coefficients in the beginning portion approach to zero as the length of the extended period increases. Finally, we proposed a sectional interactive decomposition algorithm for online trajectory decomposition (where the trajectory to be decomposed is partially known). The proposed decomposition method is illustrated through a numerical example by using B-spline as the basis function. Although the basic idea of the proposed approach has been explored in our recent publication [61], the development in [61] is rather preliminary. In this chapter, much more rigorous theoretical treatment, including the proof of the asymptotic decomposition, the completeness of the sectional interactive decomposition scheme, and a numerical technique to further reduce the online computation complexity are presented, along with more simulation results and discussions that demonstrate the proposed approach more clearly.

4.2 Problem Formulation: Single-basis Trajectory Decomposition

We assume, without loss of much generality, that

Assumption 24. *The trajectory $y_d(t): \Re \to \Re$ is smooth and compactly supported in a given period $[0, T]$.*

Assumption 25. *The basis function $\mathcal{B}(t): \Re \to \Re$ is positive, smooth, compactly supported in $[-\tau_1,\ \tau_2]$ with $\tau_1,\ \tau_2 > 0$, and $\tau_1,\ \tau_2 \ll T$, and Lipschitz, i.e.,*

$$|\mathcal{B}(t_1) - \mathcal{B}(t_2)| \le \mathcal{L}_b |t_1 - t_2|, \tag{4.1}$$

for any $t_1,\ t_2 \in [-\tau_1,\ \tau_2]$, with $\mathcal{L}_b \in \Re$ a positive constant. Moreover, the global maxima of $\mathcal{B}(t)$ occurs at $t = 0$, i.e., $\sup_t \mathcal{B}(t) = \mathcal{B}(0)$.

The requirement of $\sup_t \mathcal{B}(t) = \mathcal{B}(0)$ in Assumption 2 is employed to aid the presentation later, and can be easily satisfied via time-shifting.

To facilitate the implementation, we consider the decomposition of the trajectory in discrete domain, i.e.,

$$\mathbf{y}_d \triangleq \left[y_d[0], y_d[1], \ldots, y_d[N_b]\right] \tag{4.2}$$

with

$$y_d[i] = y_d(iT_s), \quad i = 0,\ 1, \ldots, N_b, \quad \text{and} \quad N_b = \lfloor T/T_s \rfloor,$$

where $T_s \in \Re$ is the sampling time, and $\lfloor \cdot \rfloor$ is the floor function (i.e., $\lfloor x \rfloor$ is the biggest integer smaller than $x \in \Re$). Thus, the trajectory to be decomposed and the corresponding basis functions are the discretized counterparts of those satisfying Assumptions 24, 25, respectively.

Definition 26. (**Decomposition Library**) The library of decomposition basis, $\mathcal{L}_e$, is an equivalent class of the basis function $\mathcal{B}[\cdot]$ generated via time-shifting,

$$\mathcal{L}_e = \{B_k[\cdot] : \ B_k[i] = \mathcal{B}(iT_s - k\mathit{\Psi}),\ k \in \mathbb{N},\ i \in \mathbb{Z}\}, \tag{4.3}$$

where $\mathbb{N}$ and $\mathbb{Z}$ denotes the set of natural and integer numbers, respectively, and $\mathit{\Psi} \in \Re^+$ with

$$\mathit{\Psi} \ll T \tag{4.4}$$

is the decomposition knot period, i.e., the time shift between two successive basis functions, which determines the density of the basis functions in the decomposition.

It can be easily verified that the decomposition library $\mathcal{L}_e$ defined above is indeed an equivalent class [69].

Single-basis Asymptotic Trajectory Decomposition (SBATD) The SBATD problem is to, under the above Assumptions 24, 25, achieve the following three objectives:

O_1: For any given trajectory $\mathbf{y}_d$ in Eq. (4.2), find a set of decomposition coefficients g_ks such that for

$$y_s[i] \triangleq \sum_{k=-p}^{N_d+q} g_k B_k[i], \quad \text{with} \ \ i = 0,\ 1, \cdots,\ N_b, \tag{4.5}$$

asymptotic decomposition is achieved in ℓ_2-norm, i.e.,

$$\lim_{\Psi\to 0} \mathcal{J}(\mathbf{y}_d, \Psi) = \lim_{\Psi\to 0} \|\mathbf{y}_d - \mathbf{y}_s\|_2 = 0, \tag{4.6}$$

where

$$\mathbf{y}_s \triangleq \left[y_s[0], y_s[1], \ldots, y_s[N_b]\right], \tag{4.7}$$

$$p = \lfloor \tau_2/\Psi \rfloor, \quad q = \lfloor \tau_1/\Psi \rfloor, \tag{4.8}$$

and $N_d \in \mathbb{N}$ is the total number of knot periods within the decomposition period T, and $\|\mathbf{y}\|_2$ denotes the ℓ_2 norm of a discrete sequence $\mathbf{y} = [y[0],\ y[1], \cdots,\ y[N]]$.

O_2: For a given decomposition knot period Ψ, and any $\epsilon > 0$, find the length of zero-extension period $T_{et,h} = T_{et,t} = (l_s + l_m)\Psi$ ($l_s,\ l_m \in \mathbb{N}$) to extend the trajectory at its beginning and its end (called the *head extension* and the *tail extension* below), respectively, such that for the zero-extended trajectory $\hat{\mathbf{y}}_d$ defined as

$$\hat{y}_d[i] = \begin{cases} 0, & \text{for } i = -\mathcal{K}_o, -\mathcal{K}_o + 1, \ldots,\ \ -1, \\ y_d[i], & i = 0,\ 1, \ldots,\ N_b, \\ 0, & \text{for } N_b + 1,\ N_b + 2, \ldots,\ N_b + \mathcal{K}_o, \end{cases} \tag{4.9}$$

the decomposition coefficients at the beginning of the head-extension period and the end of the tail-extension period are smaller than ϵ for any $l_s \geq l_s^*$ and any $l_m \geq l_m^*$,

$$\begin{aligned} |g_k| < \epsilon, \quad \text{for } k = -p - l_m, -p - l_m + 1, \ldots, q - l_m, \\ \text{and } \ N_d - p + l_s, N_d - p + l_s + 1, \ldots, N_d + q + l_s, \end{aligned} \tag{4.10}$$

where in Eq. (4.9), $\mathcal{K}_o = \lfloor T_{ex,h}/T_s \rfloor$.

O_3: For any given finite partition of the trajectory $\mathbf{y}_d$,

$$\mathbf{y}_d = \bigcup_{s=1}^{v} \mathbf{y}_{d,s}, \tag{4.11}$$

with

$$\mathbf{y}_{d,s} \triangleq \left[y_d[K_{s-1} + 1],\ y_d[K_{s-1} + 2], \ldots,\ y_d[K_s]\right], \tag{4.12}$$

for $s = 1,\ 2, \ldots,\ v$, redesign each section of the trajectory $\mathbf{y}_{d,s}$ as $\hat{\mathbf{y}}_{d,s}$ with an extension and smooth transition to zero of period length $\hat{\mathcal{K}}_o$, such that the decomposition coefficients of $\hat{\mathbf{y}}_{d,s}$, $\hat{g}_k$s, approach to those obtained by decomposing the entire trajectory as a whole, i.e.,

$$\lim_{K_{s-1}-\hat{\mathcal{K}}_o \to 0} |\hat{g}_k - g_k| = 0, \tag{4.13}$$

for each $s = 1,\ 2, \ldots,\ v$ and each $k = m_{s-1},\ m_{s-1}+1, \ldots,\ m_s - 1$, where $K_{s-1},\ K_s$ are the first and the last original index of the s^{th} section, respectively, and $m_s \in \mathbb{N}$ is the index of the last decomposition coefficients in the s^{th} section (globally indexed from the first section).

Note that Objective O_1 addressed the basic question of decomposing a trajectory using only one type of basis functions. However, the solution obtained requires, in general, truncation of the basis functions, as the domain of the basis functions (involved in the decomposition) is larger than that of the trajectory $\mathbf{y}_d$ itself. Thus, Objective O_2 is proposed to avoid the truncation of the basis functions via a zero-extension of the trajectory. Finally, Objective O_3 is to address two needs in online decomposition where the trajectory is not completely known and heavy online computation needs to be avoided.

4.3 Objective O_1: Trajectory Decomposition based on Least Square Minimization

In this section, we first present the decomposition solution, then show that the solution is asymptotic.

By minimizing the ℓ_2-norm of the decomposition error $\mathcal{J}(\mathbf{y}_d, \mathit{\Psi})$ in (4.6), the optimal decomposition coefficients

$$\mathbf{G} = [g_{-p},\ g_{-p+1},\ \ldots,\ g_{N_d+q}]^\top \tag{4.14}$$

can be readily obtained via the least-square-minimization [22, 60],

$$\mathbf{G}^* = (M^\top M)^{-1} M^\top \mathbf{y}_d, \tag{4.15}$$

where $\mathbf{y}_d$ is given by (4.2), and

$$M = \begin{bmatrix} B_{-p}(0) & B_{-p+1}(0) & \dots & B_{N_d+q}(0) \\ B_{-p}(1) & B_{-p+1}(1) & \dots & B_{N_d+q}(1) \\ \vdots & \vdots & \vdots & \vdots \\ B_{-p}(N_b) & B_{-p+1}(N_b) & \dots & B_{N_d+q}(N_b) \end{bmatrix}_{N'_b \times N'_d}, \tag{4.16}$$

with $N'_b = N_b + 1$ and $N'_d = N_d + p + q + 1$. Typically, $N'_b \geq N'_d$, as in practices the number of elements employed in the decomposition shall be smaller than the total number of sampled points. By Assumption 25,

$$B_k[i] = 0, \quad \text{for} \quad i \leq \lfloor (k - q - 1)\mathit{\Psi}/T_s \rfloor, \quad \text{or} \;\; i \geq \lceil (k + p + 1)\mathit{\Psi}/T_s \rceil,$$

and each $k = -p, \;\; -p+1, \dots, \;\; N_d + q$. This implies that at each time instant, only $p + q$ number of basis functions are involved in the decomposition, and M is a banded matrix [70], i.e., for all $i = \;1, \;2, \dots, \;N'_b$,

$$\begin{aligned} M_{i,j} = 0, \quad & \text{for } 1 \leq j \leq \max\{1, \; i - p + 1\} \\ & \text{or } \min\{i + q - 1, \; N'_d\} \leq j \leq N'_d. \end{aligned} \tag{4.17}$$

where $A_{i,j}$ denotes the element of any matrix $A \in \Re^{n \times m}$ at the i^{th} row and j^{th} column for any $1 \leq i \leq n$ and $1 \leq j \leq m$. The approximation error is given by [22]

$$\|e(\mathbf{y}_d, N'_b)\|_2 = (\mathbf{y}_d^\top \left(I - M(M^\top M)^{-1} M^\top \right) \mathbf{y}_d)^{1/2}. \tag{4.18}$$

Next, we show that the decomposition error approaches to zero as the knot period $\mathit{\Psi}$ decreases to zero, or equivalently, as the number of the basis functions used in the decomposition increases.

Lemma 27. *Let Assumptions 24 and 25 be satisfied, and let the decomposition coefficients and the corresponding decomposition error be given by Eqs. (4.15, 4.18), respectively. Then, for any given $\epsilon > 0$, there exists a decomposition knot period $\mathit{\Psi}$ satisfying $0 < \mathit{\Psi} \leq T / \left(N'_b - (\frac{\epsilon}{N'_b \|\boldsymbol{y}_d\|_1})^2 \right)$, such that the approximation error $\|e(\boldsymbol{y}_d, N'_b)\|_2 \leq \epsilon$.*

Proof. proof We proceed by noting that $M \in \Re^{N'_b \times N'_d}$ has full column rank, thereby, in the decomposition error $e(\mathbf{y}_d, N'_b)$ in Eq. (4.18), the matrix multiplication $M(M^\top M)^{-1} M^\top$

can be rewritten via the singular value decomposition of M,

$$M = U\Sigma V^\top \tag{4.19}$$

as

$$M(M^\top M)^{-1}M^\top = U\Sigma V^\top(V\Sigma^\top U^\top U\Sigma V^\top)^{-1}V\Sigma U^\top = UI_{sub}U^\top \tag{4.20}$$

where

$$I_{sub} = \begin{bmatrix} I_{N'_d \times N'_d} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{N'_b \times N'_b}, \tag{4.21}$$

$U \in \Re^{N'_b \times N'_b}$ and $V \in \Re^{N'_d \times N'_d}$ are both unitary matrix, and $\Sigma \in \Re^{N'_b \times N'_d}$ is a diagonal matrix. Thus by Eq. (4.20), the decomposition error $e(\mathbf{y}_d, N'_b)$ can be quantified as

$$\begin{aligned} \|e(\mathbf{y}_d, N'_b)\|_2 &= \left(\mathbf{y}_d^\top U(I - I_{sub})U^\top \mathbf{y}_d\right)^{1/2} \\ &= \left(\sum_{i=N'_d+1}^{N'_b} (\mathbf{y}_d^\top u_i)^2\right)^{1/2} \\ &\leq \sqrt{N'_b - N'_d} N'_b \|\mathbf{y}_d\|_1 \end{aligned}, \tag{4.22}$$

where u_i is the i^{th} column vector of U. The proof is completed by choosing $N'_d \geq N'_b - (\frac{\epsilon}{N'_b\|\mathbf{y}_d\|_1})^2$, i.e., $\Psi \leq T/\left(N'_b - (\frac{\epsilon}{N'_b\|\mathbf{y}_d\|_1})^2\right)$. □

4.4 Objective O_2: Trajectory Decomposition without Basis Function Truncation

Next, we show that the truncation of the basis functions can be avoided by adding a zero-extension period at the beginning and the end of the trajectory to be decomposed. As the matrix

$$\mathcal{M} \triangleq MM^\top \tag{4.23}$$

is a banded matrix (for M itself is a banded matrix), the decomposition coefficient g_k for $k = -p, -p+1, \ldots, N_d + q$ can be obtained from Eq. (4.15) as

$$g_k = \sum_{j=1}^{N'_b} (\mathcal{M}^{-1})_{k,j}\varrho_{\mathbf{y}_d,j}, \tag{4.24}$$

where

$$\varrho_{\mathbf{y}_d,j} = \sum_{i=1}^{N'_d} M_{j,i} y_d[i]. \tag{4.25}$$

Thus, central to show that $g_k \to 0$ as $k \to \pm\infty$ is to quantify $(\mathcal{M}^{-1})_{i,j}$ for any given i and j. Such a quantification is given by the Theorem below [70].

Theorem 28. *[70] Let $S = \{1, 2, \cdots, N\}$, $\ell_2(S) = \{\{x_n\}_{n=1}^N : \sum_{n=1}^N x_n^2 < \infty\}$ be a Hilbert space ($\ell_2(S)$: Set of all square summable sequences $\{x_n\}_{n=1}^N$) [71], and $B(\ell_2(S))$ be a Banach space on the Hilbert space $\ell_2(S)$, then for any A and A^{-1} in $B(l_2(S))$,*

$$|(A^{-1})_{i,j}| \leq C(A)\lambda(A)^{|i-j|}. \tag{4.26}$$

where

1. *If A is positive definite and m-banded,*

   $$\lambda(A) = f^2(\sqrt{cond(A)}) \tag{4.27}$$

   with

   $$f(x) \triangleq \left(\frac{x-1}{x+1}\right)^{1/m}, \tag{4.28}$$

   $$cond(A) \triangleq \|A\|\|A^{-1}\|, \quad \|A\| = max\{s : s \in \sigma(A)\},$$

   and $\sigma(A)$ the singular value of matrix A,

   $$C(A) = \|A^{-1}\| g(\sqrt{cond(A)}), \tag{4.29}$$

   where $g(x) \triangleq \max\{1, (1+x)^2/(2x^2)\}$.

2. *If A is non-positive definite but still m-banded, quasi-centered, bounded and invertible,*

   $$\lambda(A) = f(cond(A)) \tag{4.30}$$

   and

   $$C(A) = (m+1)\lambda^{-m}(A)\|A^{-1}\|cond(A)g(cond(A)). \tag{4.31}$$

The matrix $\mathcal{M} \in \Re^{N'_b \times N'_b}$ in (4.23) satisfies the conditions in the above Theorem as M is invertible by design (through the adjustment of the sampling instants).

Lemma 29. *Let Assumptions 24, 25 be satisfied, and for any given k^{th} decomposition knot, let $y_d[i]$ be zero for $i = \chi(k-l_s),\ \chi(k-l_s)+1, \ldots,\ \chi(k+l_m)$, where*

$$\chi(k) = \lceil k\Psi/T_s \rceil, \tag{4.32}$$

and the decomposition coefficients be given in Eq. (4.15), then for any given $\epsilon > 0$, there exist l_s and $l_m \in \mathbb{N}$ with

$$l_s \geq \left\lceil \ln \frac{\epsilon(1-\lambda(M))}{C(M)\|\boldsymbol{y}_d\|_1} / \ln \lambda(M) + p \right\rceil, \tag{4.33}$$

and

$$l_m \geq \left\lceil \ln \frac{\epsilon(1-\lambda(M))}{C(M)\|\boldsymbol{y}_d\|_1} / \ln \lambda(M) + q \right\rceil, \tag{4.34}$$

such that the decomposition coefficients of the i^{th} decomposition element g_i, satisfying $|g_i| \leq \epsilon$, where $\lambda(M)$ and $C(M)$ are given in Theorem 28 (with A replaced by M), respectively, and $\lceil\cdot\rceil$ and $\|\mathbf{y}[\cdot]\|_1$ denote the ceiling function (i.e., for any $x \in \Re$, $\lceil x \rceil$ is the smallest integer bigger than x) and the ℓ_1-norm of a discrete sequence $\mathbf{y}[\cdot]$, respectively.

Proof. proof As matrix $\mathcal{M}$ is a banded matrix, by Eq. (4.26),

$$(\mathcal{M}^{-1})_{i,j} \leq C(M)\lambda(M)^{|i-j|}. \tag{4.35}$$

Thus, by Eq. (4.24),

$$g_i = \sum_{j=1}^{N'_b} (\mathcal{M}^{-1})_{i,j} \varrho_{\mathbf{y}_d,j} \leq \sum_{j=1}^{N'_b} C(M)\lambda(M)^{|i-j|} \varrho_{\mathbf{y}_d,j}. \tag{4.36}$$

It can be verified that by choosing l_s and l_m as in (4.33) and (4.34), respectively, we have

$$\varrho_{\mathbf{y}_d,j} = 0 \quad \text{for } i-p+1 \leq j \leq i+q-1, \tag{4.37}$$

and

$$g_i \leq \sum_{j=1}^{i-l_s} C(M)\lambda(M)^{|i-j|} \varrho_{\mathbf{y}_d,j} + \sum_{j=i+l_m}^{N'_b} C(M)\lambda(M)^{|i-j|} \varrho_{\mathbf{y}_d,j} \leq \epsilon. \tag{4.38}$$

This completes the proof. □

The following corollary follows directly.

Corollary 30. *Let Assumptions 24, 25 be satisfied, then for any given $\epsilon > 0$, and any given zero period length*

$$\mathcal{K}_o = \lfloor (l_s + l_m)\varPsi/T_s \rfloor \tag{4.39}$$

with l_s and l_m given by (4.33), (4.34), respectively, the decomposition coefficients g_is in Eq. (4.15) satisfy $|g_i| \leq \epsilon$ for $i = -p - l_m, \ -p - l_m + 1, \ldots, \ q - l_m$ and $i = N_d - p + l_s, \ N_d - p + l_s + 1, \ldots, \ N_d + q + l_s$.

The above corollary implies that by having a long enough zero extension at the beginning and the end of the trajectory, the elements involved in the decomposition vanish as those elements are farther away from the "original" beginning and end points of the trajectory, respectively, and hence, truncation of the basis functions can be avoided.

4.5 Objective O_3: Sectional Interactive Trajectory Decomposition

Under the following Assumption:

Assumption 31. *The domain of each section is much large than the knot period, i.e., $(K_s - K_{s-1})T_s \gg \varPsi$ for any given $s = 1, \ 2, \ldots, \ v$.*

We aim to further achieve Objective O_3 by utilizing a sectional interactive decomposition (SID) algorithm without truncation of basis functions, where each section of the trajectory, in general, does not start and/or end at zero. Note that the above Assumption 31 can be always satisfied by selecting a sufficiently small knot period $\varPsi$.
More concretely, when decomposing any s^{th} partition of the trajectory $\mathbf{y}_{d,s}$ (given by Eq. (4.12)), the basis functions $B_k[\cdot]$ for $k = m_s - p, \ m_s - p + 1, \ldots, \ m_s + q$ (m_s: the index of the last decomposition knot in the s^{th} section), are used (the green splines in Fig. 4.1(a)), where a subset of these basis functions, $B_k[i]$ with $i > K_s$, for $k = m_s - p, \ m_s - p + 1, \ldots, \ m_s + q$, falls outside the domain of the s^{th} section trajectory (the shadowed part in Fig. 4.1(a)), resulting in the truncation of these basis functions. Thus, we propose to redesign the beginning portion of the current section of the trajectory by deducting from it the weighted summation of those B-splines involved

in the decomposition of the preceding section (that falls into the current section), such that the redesigned current section starts from zero and smoothly transits to the next section. As such, truncation of the elements in the decomposition is avoided. Specifically, for the s^{th} section, the weighted summation of the B-splines that fall outside the s^{th} section domain and into the $s+1^{th}$ section domain (the red dashed curve in Fig. 4.1(a)), $\mathbf{y}_{e,s}$, is given by

$$y_{e,s}[i] = \begin{cases} \displaystyle\sum_{k=m_s-p}^{m_s+q} g_k B_k[i], \quad i = K_s+1, K_s+2, \ldots, K_s+\chi(p); \\ 0, \quad \text{otherwise}, \end{cases} \tag{4.40}$$

then, the $s+1^{th}$ section trajectory is redesigned as (see Fig. 4.1(b) for the redesigned trajectory)

$$\hat{\mathbf{y}}_{d,s+1} = \mathbf{y}_{d,s+1} - \mathbf{y}_{e,s}. \tag{4.41}$$

Then the decomposition coefficients of the redesigned $s+1^{th}$ section trajectory $\hat{g}_i$, $i = m_s - l_m + q,\ m_s - l_m + q + 1, \ldots,\ m_{s+1} + q$ are used to update the decomposition coefficients for the original $s+1^{th}$ section.

We summarize this decomposition scheme in Algorithm 1 below. As shown in

: Sectional Interactive Decomposition (SID)

Step 1. For any given $s+1^{th}$ section of the trajectory, let

$$K'_s = K_s - \chi(l_s + l_m),$$

where $l_s,\ l_m$ are given by Eq. (4.33, 4.34), respectively. Then we redefine the modified trajectory $\hat{\mathbf{y}}_{d,s+1}$ as

$$\hat{y}_{d,s+1}[i] = \begin{cases} 0, \quad i = K'_s,\ K'_s+1, \ldots,\ K_s; \\ y_{d,s+1}[i] - y_{e,s}[i], \quad i = K_s+1,\ K_s+2, \ldots,\ K_s+\chi(p+q); \\ y_{d,s+1}[i], \quad i = K_s+\chi(p+q)+1, \ldots, K_{s+1}. \end{cases} \tag{4.42}$$

Step 2. Update the decomposition coefficients as

$$g_i = \begin{cases} g_i + \hat{g}_i, \quad i = m_s - l_m + q,\ m_s - l_m + q + 1, \ldots,\ m_s + q; \\ \hat{g}_i, \quad i = m_s + q + 1,\ m_s + q + 2, \ldots,\ m_{s+1} + q. \end{cases} \tag{4.43}$$

Fig. 4.1(b), the redesigned $s+1^{th}$ section of the trajectory $\hat{\mathbf{y}}_{d,s+1}$ starts with zero.

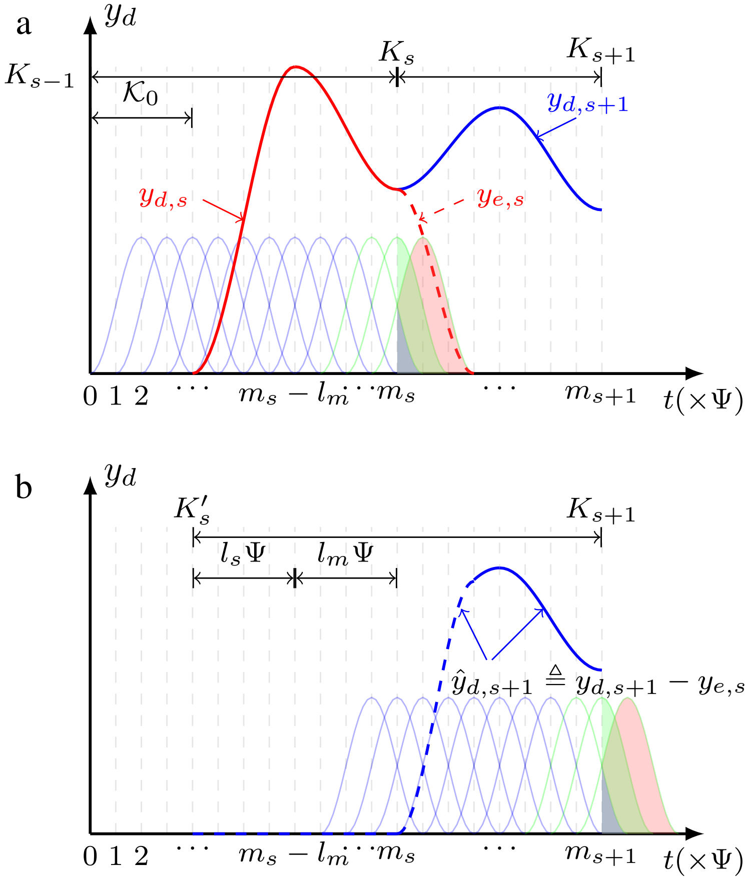

Figure 4.1: Sectional decomposition process.

Moreover, by extending the beginning of each redesigned section sufficiently long, the basis functions for the decomposition of the extended beginning portion can be ignored (by Corollary 30). Thus, the truncation issue in online trajectory decomposition is avoided. The following lemma shows that with a long enough zero-extension period, the decomposition coefficients of the above SID and those obtained when decomposing the entire trajectory as a whole are equivalent to each other.

Lemma 32. *Let Assumptions 24, 25 and 31 be satisfied, and let g_is and g_i°s be the coefficients for decomposing the trajectory $\boldsymbol{y}_d$ using the above SID technique and as a whole via Eq. (4.15), respectively. Then, $lim_{K'_s \to 0}|g_i - g_i^\circ| = 0$.*

Proof. proof We first show that the coefficients obtained for the redesigned s^{th} section is equivalent to the decomposition of the trajectory $\tilde{y}_{d,s}$ defined as

$$\tilde{y}_{d,s}[i] = \begin{cases} \hat{y}_{d,s}[i], & i = K'_{s-1},\ K'_{s-1}+1, \ldots,\ K_s; \\ y_{e,s}[i], & i = K_s + 1,\ ,K_s + 2, \ldots,\ N_b. \end{cases} \tag{4.44}$$

By the definition of $\mathbf{y}_{e,s}$ in Eq. (4.40), we can find that the decomposition coefficients of $\tilde{\mathbf{y}}_{d,s}$ obtained through the least square minimization process equal to those of $\hat{\mathbf{y}}_{d,s}$, and other decomposition coefficients beyond the domain of $\hat{\mathbf{y}}_{d,s}$ are all zeros.
Then, if we let $K'_s = 0$ for each $s = 1,\ 2, \ldots,\ v$, the summation of these trajectory $\tilde{\mathbf{y}}_{d,s}$ for $s = 1,\ 2, \ldots,\ v$ is equivalent to the whole trajectory $\mathbf{y}_d$, i.e., $\mathbf{y}_d = \sum_{s=1}^{v} \tilde{\mathbf{y}}_{d,s}$, thereby, it can be simply implied that the vector summation of the decomposition coefficient vectors corresponding to each trajectory $\tilde{\mathbf{y}}_{d,s}$, for $s = 1,\ 2, \ldots,\ v$, is equivalent to the decomposition coefficient vector obtained when decomposing the trajectory as a whole. This completes the proof. □

Note that in online implementation, the computation can be substantially reduced by evenly partitioning the trajectory, so that the matrix $\mathcal{M} = MM^\top$ in Eq. (4.15), which can be computed offline a priori, is the same for the decomposition of all sections. Also, we can further reduce the computation in computing each g_i by exploiting the decaying property of matrix $\mathcal{M}$ in Eq. (4.15), i.e., the coefficient g_i can be computed as

$$g_i = \sum_{j=l_u}^{l_b} (\mathcal{M}^{-1})_{i,j} \varrho_{\mathbf{y}_d,j}, \tag{4.45}$$

where $l_u = \max\{1, i - l_s\}$, $l_b = \min\{N_d, i + l_m\}$, and $\varrho_{\mathbf{y}_d}$ is given in (4.25). As the term $(\mathcal{M}^{-1})_{i,j}$ can be computed offline a priori, the computation involved in Eq. (4.45) is substantially reduced and thereby, more suitable than Eq. (4.15) in online implementation.

4.6 Numerical Example: B-spline-based Trajectory Decomposition

We next use B-spline as an example of basis functions to demonstrate the proposed online trajectory decomposition method, i.e., the three Objectives O_1 to O_3.

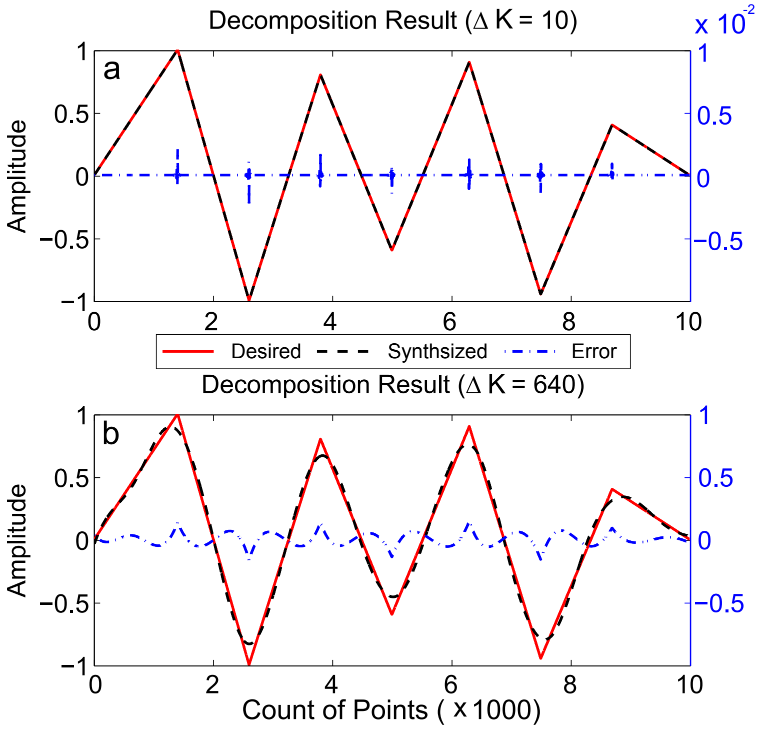

Figure 4.2: Comparison of the decomposition results when the knot period $\varPsi = 10$ (a) and $\varPsi = 640$ (b).

4.6.1 B-spline

Satisfying all the conditions required in Assumption 25, the s^{th}-degree B-spline basis functions can be obtained recursively via

$$B_{i,s}(t) = \frac{t - t_{d,i}}{t_{d,i+s} - t_{d,i}} B_{i,s-1}(t) + \frac{t_{d,i+s+1} - t}{t_{d,i+s+1} - t_{d,i+1}} B_{i+1,s-1}(t), \tag{4.46}$$

where

$$B_{i,0} = \begin{cases} 1, & i\Delta t \leq t < (i+1)\Delta t; \\ 0, & otherwise. \end{cases} \tag{4.47}$$

The basis element $\mathcal{B}(t)$ in Eq. (4.3) can be constructed by shifting any $B_{i,s}(\cdot)$ to centering at zero. Note that Δt in Eq. (4.47) corresponds to the knot period $\varPsi$ in continuous time domain. The 3^{rd} degree ($s = 3$ in (4.46)) B-splines are utilized in the decomposition, where $p = q = 1$ (See Eq. (4.8)), i.e., at any given time instant, at most 4 B-splines (same type) are involved in the decomposition.

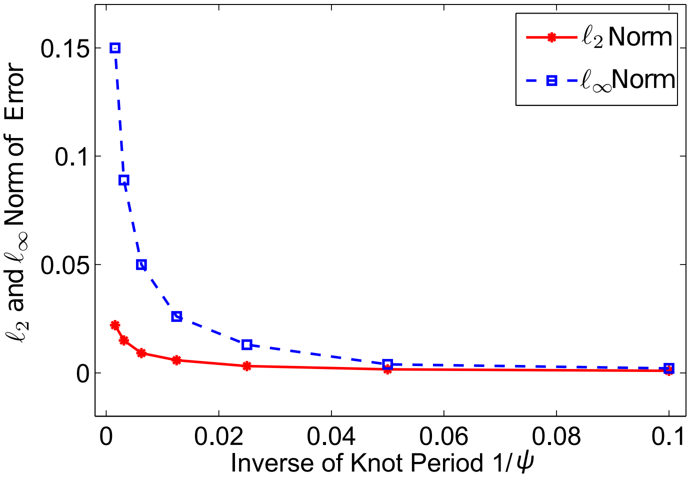

Figure 4.3: The ℓ_2-norm and ℓ_∞-norm of the decomposition error vs. the knot period Ψ.

4.6.2 Simulation Results

To simplify the presentation, the sampling time was normalized in the simulation, i.e., one unit of the time axis represents one sampling period.

First, we show the simulation results to illustrate Objective O_1. A triangle wave signal with varying amplitude (see Fig. 4.2) was employed as the trajectory to be decomposed. The approximated trajectories by using knot period $\Psi = 10$ (i.e., 10 sampling periods) and $\Psi = 640$, respectively, are compared with the original trajectory in Fig. 4.2, where a total of 1004 (for $\Psi = 10$) and 20 (for $\Psi = 640$) uniform B-splines were used for $\Psi = 10$ and $\Psi = 640$, respectively. The ℓ_2-norm and ℓ_∞-norm of the decomposition errors as the knot period decreased from 640 to 10 are shown in Fig. 4.3, respectively. It can be seen from Fig. 4.2 that the decomposition error occurred around the vertex of the triangle wave. However, by using enough number of decomposition knots, precision decomposition can be achieved, even around the vertexes. As shown in Fig. 4.3, both the ℓ_2-norm and ℓ_∞-norm of the decomposition error monotonically and rapidly decreased to 9.5×10^{-4} and 2.1×10^{-3}, respectively, as the knot period decreased to $\Psi = 10$. Thus, arbitrary decomposition precision and asymptotic decomposition can be achieved.

Secondly, we show the results to illustrate Objective O_2 by using a triangle wave (see

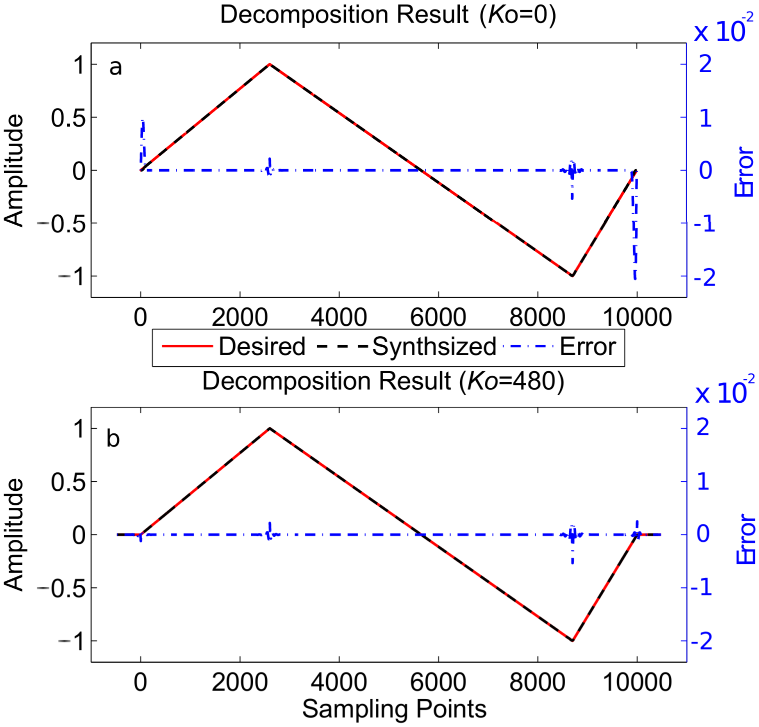

Figure 4.4: Comparison of the decomposition results when the zero extension period is $\mathcal{K}_o = 0$ (a) and $\mathcal{K}_o = 480$ (b).

Fig. 4.4), where the knot period was chosen at $\varPsi = 40$ so that the decomposition error became negligible and truncation of the basis function was allowed.

To examine the effect of the length of the zero-extension period on the values of the decomposition coefficients, the approximated/decomposed trajectories for two different lengths of the zero-extension period at $\mathcal{K}_o = 0$ and $\mathcal{K}_o = 12\varPsi = 480$, respectively, are compared with the original trajectory in Fig. 4.4, where the approximated trajectory was obtained by neglecting those B-splines that fell outside the corresponding zero-extension period (a total of 3 B-splines). Thus, by Lemma 29 and Corollary 30, as the length of the zero-extension period increased, the coefficients of those three B-splines at the beginning of the head extension and the end of the tail extension approached to zero, respectively, so did the approximation error caused by ignoring/removing those three B-splines. Thus, we defined the head-coefficient cutoff indicator and the tail-coefficient

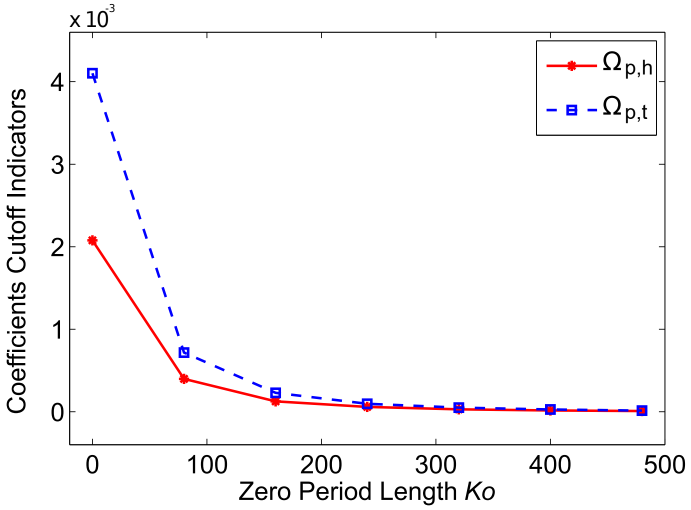

Figure 4.5: The the head coefficient-cutoff indicator $\Omega_{p,h}$ and the tail coefficient-cutoff indicator $\Omega_{p,t}$ vs. the zero-extension period $\mathcal{K}_o$.

cutoff indicator, $\Omega_{p,h}$ and $\Omega_{p,t}$, respectively, as

$$\Omega_{p,h} = \sum_{k=l_s-p}^{l_s+q} |p_k|, \text{ and } \Omega_{p,t} = \sum_{k=N_d+l_m-p}^{N_d+l_m+q} |p_k|. \tag{4.48}$$

The variations of the head and the tail coefficient-cutoff indicators as the length of the zero-extension period increased from $\mathcal{K}_o = 0$ to $\mathcal{K}_o = 12\Psi = 480$ are shown in Fig. 4.5, respectively.

The simulation results demonstrated that truncation of the basis elements can be avoided in the decomposition by having a long enough zero-extension period. The approximation error was pronounced when there was no zero-extension and the three B-splines beyond the domain of the trajectory were ignored, particularly, at the beginning and the end of the trajectory (see Fig. 4.4(a)). However, by having a zero-extension of 12 knot periods, such an approximation caused by ignoring the three outside-domain B-splines were dramatically reduced (see Fig. 4.4(b)). Particularly, as shown in Fig. 4.5, the amplitude of the decomposition coefficients in both the beginning and the end of the head and the tail extension period decreased exponentially and rapidly towards zero as the length of zero-extension period increased.

Last, we show the simulation results to demonstrate Objective O_3. A long triangle wave with large amplitude variations was employed in the decomposition (see Fig. 4.7).

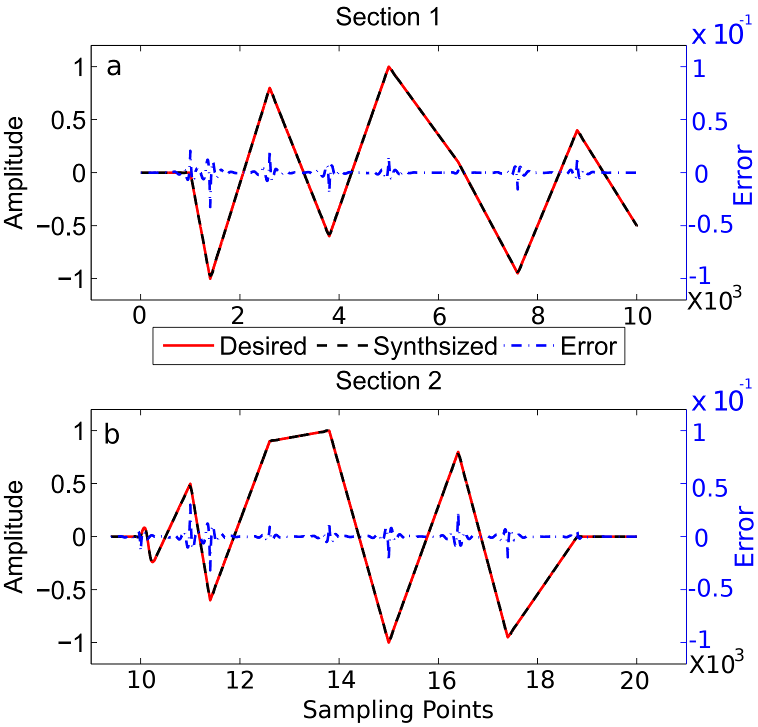

Figure 4.6: Sectional decomposition for a long trajectory: section 1 (a); section 2 (b).

The trajectory was partitioned into two sections at the middle point of the entire trajectory (marked by the vertical dashed line in Fig. 4.7). The SID algorithm proposed in Sec. 4.5 was utilized to decompose the trajectory sectionally. The trajectory to be decomposed was partitioned into two sections (shown in Fig. 4.6), and each section was decomposed sequentially. The knot period $\varPsi = 100$ was chosen, and the zero-extension period for each section was chosen to be long enough at $6\varPsi$. The decomposed trajectory obtained via the SID algorithm is compared to the original trajectory in Fig. 4.7, and the comparison for each section (the synthesized vs. the section or redesigned section to be decomposed) with the SID algorithm is presented in Fig. 4.6. Moreover, the decomposition coefficients obtained by using the SID algorithm are compared to those obtained by decomposing the entire trajectory as a whole (called total decomposition) in Fig. 4.8, where the difference between these two sets of coefficients is also shown.

In Fig. 4.6, no tail-extension was applied to the first section as those three B-splines

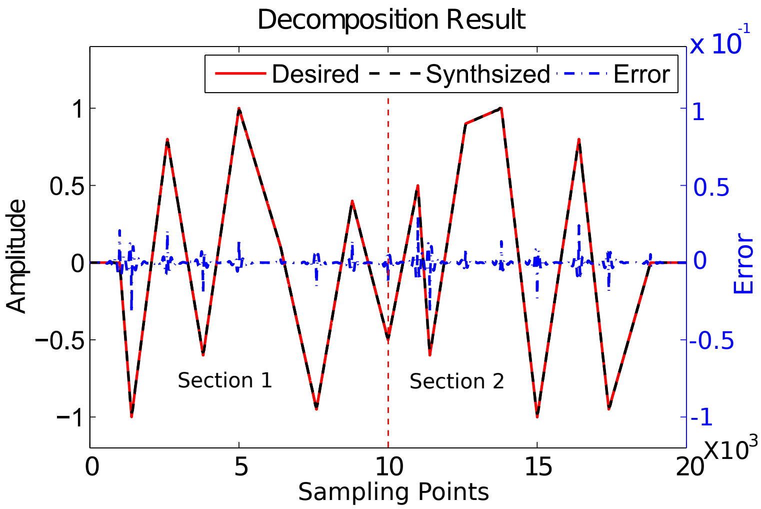

Figure 4.7: Sectional decomposition for a long trajectory with both section combined.

exceeding the end of the first section were accounted for in the decomposition of the second section, and both sections achieved high precision decomposition. Fig. 4.7 further demonstrated that when combined together, the proposed SID method achieved high precision, with the ℓ_2-norm and ℓ_∞-norm of the decomposition error at 2.6×10^{-3} and 3.0×10^{-2}, respectively, and the difference was negligible when compared with the decomposition error with the method proposed in Sec. 4.3. Moreover, Fig. 4.8 shows that the coefficients obtained by both methods almost coincided with each other with negligible difference. Therefore, the simulation results show that the proposed SID can achieve same precision decomposition with less computation complexity.

4.7 Conclusion

In this chapter, a trajectory decomposition method using single basis were proposed for the applications in control. First, a trajectory decomposition method based on a least square minimization process was proposed and the decomposition error was proved to be arbitrarily small as the decomposition knot period goes to zero. Then, we showed that by adding a zero period at the beginning and the end of the trajectory, the first few coefficients can be ignored, thus, the truncation of the basis elements can be avoided.

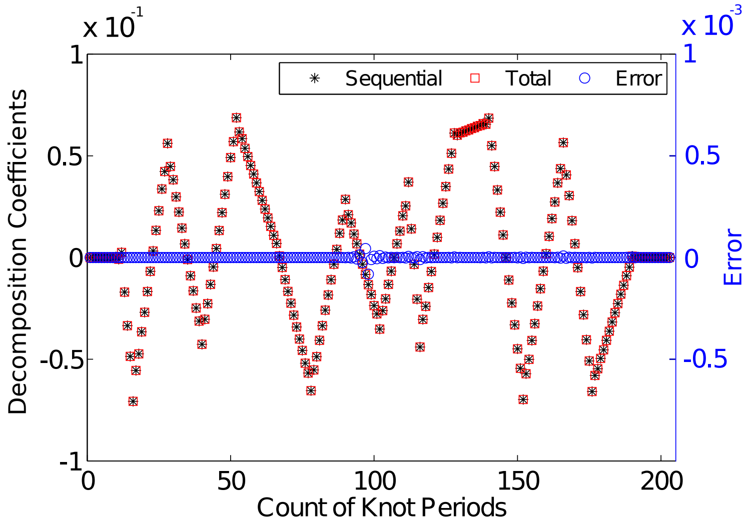

Figure 4.8: Comparison of the decomposition coefficients obtained by SID method and total decomposition.

Furthermore, we proposed a sectional interactive decomposition algorithm (SID) to decompose a previewed trajectory with only one basis function. The numerical results demonstrated the efficacy of the proposed method.

Chapter 5

Optimal Time-Distributed Fast Fourier Transform: Application to Online Iterative Learning Control—Experimental High-Speed Nanopositioning Example

Abstract

In this chapter, an optimal time-distributed fast Fourier transform algorithm and a time-distributed inverse fast Fourier transform (OTD-FFT/TD-IFFT) algorithm are proposed. This work is motivated by the need to implement FFT/IFFT online on general microprocessors (e.g., Intel's x86 microprocessors) in control applications and signal processing, for example, the online implementation of frequency domain iterative learning control (FD-ILC) techniques. In these applications, the conventional FFT algorithm executes all the calculations within one single sampling period, thereby, becoming the bottleneck in online implementations of signal processing and control algorithms. In the proposed OTD-FFT technique, the FFT computation of an online sampled data sequence is optimally distributed among all the sampling periods without increasing the total computational complexity, arriving at the minimal per-sampling-period computational complexity. As a result, the entire Fourier transformed sequence is obtained without latency. The proposed approach is extended to online IFFT computation, and then applied to online FD-ILC implementation. The computational complexity analysis shows that by using the proposed approach, the per-sampling-period computational complexity is substantially reduced. The efficacy of the proposed OTD-FFT/TD-FFT for online implementation of FD-ILC technique is demonstrated through experiments of high-speed trajectory tracking on a piezoelectric actuator.

5.1 Introduction

In this chapter, an optimal time-distributed fast Fourier transform algorithm and a time-distributed inverse fast Fourier transform algorithm (OTD-FFT/TD-IFFT) are developed for online implementation of FFT/IFFT in applications including frequency domain iterative learning control (FD-ILC) [72, 1, 34], digital audio signal processing [73], and online signal detection [74, 75]. In these applications, online FFT/IFFT is central in the corresponding signal processing and control algorithm execution, and demands the most intensive computation. The intensive online FFT/IFFT computation, however, can be challenging for the hardware used, and becomes the bottleneck in these implementations. Although special hardware and/or software have been proposed for online FFT/IFFT [76, 28, 75], these existing approaches are not suitable for applications such as the online implementation of FD-ILC on general-purpose microprocessors of Von-Neumann architecture (e.g., Intel's x86 microprocessors), where the FFT/IFFT computation needs to be executed in one single sampling period. Through the proposed OTD-FFT/TD-IFFT, we aim to address such challenges and substantially reduce the per-sampling-period computation in online control and signal processing applications. Limitations exist in current techniques developed for online implementation of FFT/IFFT. For example, a pipeline processor based on very large scale integration (VLSI) technology has been proposed for computing FFT of fixed-length (e.g., 256 points) [27], and extended to process data sequences of longer but still *fixed* lengths [28], by using a modified representation of the FFT algorithm. These methods, however, require specially designed VLSI, and hence, are not flexible for processing data sequences of various lengths. These limitations in FFT computation for different data lengths might be addressed through hardware development based on field-programmable gate array (FPGA) and digital signal processor (DSP) technologies, such as a radix-2 single-path delay feedback pipelined FFT/IFFT processor based on FPGA [29], and a compact online FFT algorithm based on DSP [30]. Although these FPGA/DSP-based techniques improved the flexibility in online FFT computation, specially-designed hardware is needed. Alternatively, algorithms have been proposed to improve online computation of FFT on general purpose microprocessors (e.g., Intel's x86 microprocessors), including a distributed FFT algorithm to process a large block of incoming data [31] that spreads the computation across smaller blocks of the incoming data, and a multi-rate controller based on a new discrete lifted system model to save computation by reducing the length of input data [32]. In these efforts, however, the per-sampling-period complexity is still significant, and latency is induced when the transformed (output) sequence is obtained. Therefore, there are needs to develop techniques to further improve the real time implementation of FFT/IFFT, particularly, on general-purpose microprocessors.

High-speed online implementation of FFT/IFFT is needed in applications such as online application of control techniques like FD-ILC [1] and audio signal processing [73]. For example, FD-ILC techniques have been developed and integrated with feedback control in the feedforward-feedback two-degree-of-freedom (2DOF) control configuration, and applied in applications such as tracking of the sample topography in high-speed atomic force microscope (AFM) imaging [1, 77, 78]. It is demonstrated that by using this FD-ILC-based 2DOF control approach, the imaging speed can be significantly increased [1, 78]. Further increase of the imaging speed, however, has been limited by the hardware incapability to online execute the FFT/IFFT computation fast enough [1, 78]. Although such a hardware limitation might be alleviated by converting and representing the FD-ILC algorithms [79, 35, 78, 80] in time domain via, for example, the corresponding finite impulse response [81], a lifted model [25] and a transfer function model, approximation is induced in the system identification process, and high-order approximation is needed for tracking at high speeds. Moreover, in application such as fast audio signal processing, FFT is linked to operations such as visualization, signal mixture, and watermarking on output devices such as monitors and speakers [82, 83], requiring the microprocessor to be flexible and possess various peripheral interfaces. Such a combined requirement—heavy computation coupled with versatile peripheral interactions—renders general purpose microprocessors a more suitable choice. Thus, highly-efficient FFT/IFFT implementation is essential in these microprocessor-based real-time control and mechatronics applications.

The main contribution of this work is the development of an OTD-FFT/TD-IFFT technique with direct application to online FD-ILC. Without increasing the total computational complexity, the proposed OTD-FFT/TD-IFFT algorithm exploits the butterfly structure of FFT/IFFT to optimally distribute the FFT computation of an online sampled data sequence to each of the sampling period, such that the per-sampling-period computational complexity is minimized, and the entire Fourier transform sequence is obtained without latency. Specifically, through the proposed OTD-FFT, the maximal per-sampling-period computational complexity for a data sequence of length 2^N is reduced from $2^N \log 2^N$ multiplications and $2^N \log 2^N$ additions to $2^{N+1} - 2$ multiplications and $2^{N+1} - 2$ additions, respectively. Similar idea is extended to the IFFT computation in the proposed TD-IFFT algorithm, where, with the frequency domain data of length 2^N already known, the butterfly computation of IFFT is sequentially distributed to the first $2N$ sampling periods. The proposed OTD-FFT/TD-IFFT algorithm is applied to online implementation of FD-ILC, and the computational complexity of the proposed OTD-FFT/TD-IFFT and the FD-ILC is analyzed. The efficacy of the proposed approach in online control applications is demonstrated through experimental implementation of a recently-developed FD-ILC technique [35] to high-speed precision trajectory tracking on a nanopositioning system.

5.2 Optimal Time-Distributed FFT/IFFT with Minimal Per-Sampling-Period Computational Complexity

In this section, we present, first, the FD-ILC algorithm as a motivational example, and then the proposed optimal time-distributed FFT and time-distributed IFFT (OTD-FFT/TD-IFFT) algorithms.

5.2.1 Motivational Example: Frequency-Domain Iterative Learning Control

Consider a general form of the FD-ILC algorithm given in the discrete-frequency domain [79]

$$U_i[j\omega_k] = Q[j\omega_k](U_{i-1}[j\omega_k] + L[j\omega_k]e_{i-1}[j\omega_k]), \tag{5.1}$$

where $\omega_k = 2k\pi/T$, $k = 0,\ 1, \ldots,\ N_s$, $N_s = \lfloor T/(2T_s) \rfloor$, T and T_s are the period of the iteration and the sampling period, respectively, $e_{i-1}[j\omega_k]$ is the tracking error of the $(i-1)^{\text{th}}$ iteration output, $Y_{i-1}[j\omega_k]$, with respect to the desired trajectory, $Y_d[j\omega_k]$, i.e.,

$$e_{i-1}[j\omega_k] = Y_d[j\omega_k] - Y_{i-1}[j\omega_k], \quad i \in \mathbb{N}^+, \tag{5.2}$$

$Y_i[j\omega_k]$ and $U_i[j\omega_k]$ are, respectively, the discrete Fourier transform of the input and output in the i^{th} iteration, and $Q[j\omega_k]$ and $L[j\omega_k]$ are the"Q filter" and "L filter" in the ILC to be designed, respectively [79].

When implementing the above FD-ILC directly in frequency domain, $Q[j\omega_k]$, $L[j\omega_k]$, and $Y_d[j\omega_k]$ can be determined a priori (i.e., the frequency responses corresponding to $Q[j\omega_k]$ and $L[j\omega_k]$, respectively), and the discrete Fourier transform, $Y_d[j\omega_k]$, can be obtained in advance. Thus, the implementation amounts to, after the $(i-1)^{\text{th}}$ iteration, obtaining the discrete Fourier transform (DFT) of the output trajectory via FFT, $Y_{i-1}[j\omega_k]$, then the frequency domain input $U_i[j\omega_k]$ via Eq. (5.1), and finally, the time domain input $u_i[n]$ for the i^{th} iteration via the inverse DFT.

In offline implementation, the FFT/IFFT computation required in the above frequency domain ILC algorithm (5.1) usually will not pose computational challenge, i.e., after obtaining the $(i-1)^{\text{th}}$ iteration output $y_{i-1}[n]$, the control operation is paused, and the i^{th} iteration input $u_i[n]$ is computed offline and then, after the operation is resumed, applied in the i^{th} iteration. Implementing the ILC (5.1) *online* on a general-purpose microprocessor, however, can be challenging—At the beginning of each iteration, the iterative control input must be obtained and available to be applied. Thus, in such an online implementation via classical FFT/IFFT, both the FFT and the IFFT must be computed (along with other additional computations) all within one sampling period– the sampling period right before the first sampled point of the next iteration trial. Such heavy computations become the bottleneck in online implementation of FD-ILC.

5.2.2 Optimal Time-Distributed Real-time FFT/IFFT

Through the proposed optimal time-distributed FFT/IFFT (OTD-FFT/TD-IFFT), we aim to minimize the per-sampling-period computational complexity without increasing the total computational complexity. First, the classic radix-2 FFT/IFFT is briefly reviewed.

Review: Radix-2 FFT and IFFT [84, 85]

FFT/IFFT exploits the periodicity and symmetricity existing in discrete-Fourier transform (DFT) to dramatically reduce the computational complexity and increase the execution efficiency. Particularly, for a real sequence of length of power of 2, $\{x[n] \in \Re|\ n = 0,\ 1, \ldots,\ M-1,\ M = 2^N,\ N \in \mathbb{N}\}$, $\mathbb{N}$: the set of natural numbers, its DFT, $X[k]$, can be obtained via [84]:

$$\begin{aligned} X[k] &= \sum_{n=0}^{2^N-1} x[n] w_M^{kn}, \\ &= \sum_{r=0}^{2^{(N-1)}-1} x[2r] w_M^{2rk} + w_M^k \sum_{r=0}^{2^{(N-1)}-1} x[2r+1] w_M^{2rk}, \\ &\triangleq E_N[k] + w_M^k O_N[k]. \end{aligned} \tag{5.3}$$

where

$$w_M = e^{-j(2\pi/M)}, \quad \text{and } w_M^k = e^{-j(2k\pi/M)}, \tag{5.4}$$

and $E_N[\cdot]$ and $O_N[\cdot]$ are the even and odd subsequence of $x[\cdot]$, respectively. $E_N[\cdot]$ and $O_N[\cdot]$ are recursively decomposed in the same manner until the subsequences $E_0[\cdot]$ and $O_0[\cdot]$ each contain only one element from $x[\cdot]$.

In practices, the above calculation of FFT induces $N+1$ number of intermediate sequences successively,

$$X_0[\cdot],\ X_1[\cdot], \ldots,\ X_N[\cdot] \quad \text{with } X_N[\cdot] = X[\cdot]$$

and each intermediate sequence $X_p[\cdot] \in \mathbb{C}^{1\times M}$ ($\mathbb{C}$: the set of complex numbers), for $p = 0,\ 1, \ldots, N$, is obtained via an induction process,

$$\begin{aligned} X_p[I_i : I_f] &= [P_{p-1}(I_i), P_{p-1}(I_i)] + \mathbf{w}_{2p} \circ [S_{p-1}(I_f), S_{p-1}(I_f)] \\ &\triangleq \Phi(X_{p-1}[I_i : I_f], \mathbf{w}_{2p}), \end{aligned} \tag{5.5}$$

where $X_p[I_i : I_f] \triangleq [X_p[I_i], X_p[I_i+1], \ldots, X_p[I_f-1], X_p[I_f]]$, "$\circ$" denotes the Hadamard product of two vectors of the same size (i.e., element-wise product), e.g., for any given two vectors $[a_1,\ b_1,\ c_1]$ and $[a_2,\ b_2,\ c_2]$, $[a_1,\ b_1,\ c_1] \circ [a_2,\ b_2,\ c_2] = [a_1a_2,\ b_1b_2,\ c_1c_2]$, and for $i = 1,\ 2, \cdots,\ 2^{N-p}$,

$$\begin{aligned} I_i =& 2^p(i-1), \quad I_f = 2^p i - 1, \\ P_{p-1}(I_i) &= X_{p-1}[I_i : I_i + 2^{p-1} - 1], \\ S_{p-1}(I_f) &= X_{p-1}[I_f - 2^{p-1} + 1 : I_f], \end{aligned} \tag{5.6}$$

with

$$\mathbf{w}_{2^p} = [w_{2^p}^0,\ w_{2^p}^1, \ldots,\ w_{2^p}^{2^p-1}]. \tag{5.7}$$

Initially,

$$X_0[\mathbf{I}(n)] = x[n], \quad \text{with } \mathbf{I}(n) = \text{dec}(\text{rev}(\text{bin}(n))), \tag{5.8}$$

where bin(n) denotes the binary representation of an integer n, rev(n) denotes the reverse of n, e.g., rev$(1011) = 1101$, and dec(b) denotes the decimal representation of a binary number b, e.g., dec$(1101) = 13$. Thus, the function $\mathbf{I}(n)$ is identical to its inverse, i.e., for any given n, $\mathbf{I}(n) = \mathbf{I}^{-1}(n)$.

Similarly, IFFT is computed in the same manner. Conventionally, the computation of FFT/IFFT requires the whole sequence $x[\cdot]$ or $X[\cdot]$ to be known a priori, and the computation—on general purpose microprocessors of Von-Neumann architecture—needs to be completed in one single sampling period, thus, not suitable for online implementation.

5.2.3 Optimal Time-Distributed FFT and IFFT

Time-Distributed FFT

The proposed algorithm is for computing FFT concurrently when the sampled data sequence is acquired, and obtaining the FFT result without latency.

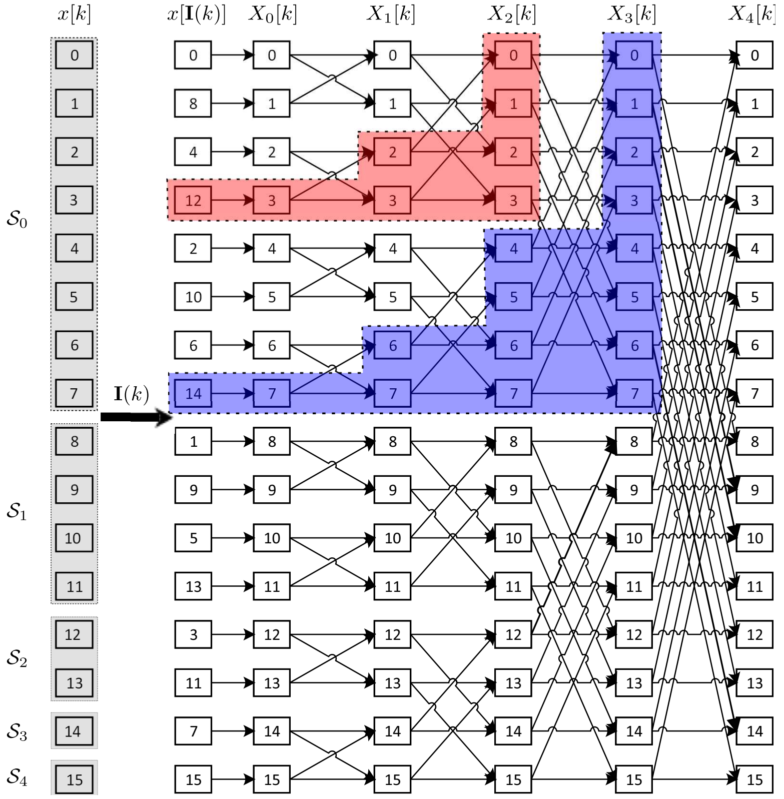

Figure 5.1: The schematic chart of Radix-2 FFT, where the pink and purple dashed-line boxes indicate the computation sequences enabled by the newly-sampled data $x[12]$ (pink) and $x[14]$ (purple) at the 13th and the 15th sampling period, respectively.

Optimal Time-Distributed FFT (OTD-FFT): Let $\{x[n] \in \Re |\, n = 0,\ 1, \cdots,\ M-1\}$ with $M = 2^N,$ and $N \in \mathbb{N}^+$ be a sampled input sequence acquired in real time, i.e., at any sampling time instant $p \in [0, M-1]$, $x[n]$ is known $\forall n \leq p$, then the OTD-FFT problem is to obtain the DFT sequence of $x[\cdot]$, $X[\cdot]$, such that

- The total computational complexity is the same as that of FFT;
- The maximal per-sampling-period computational complexity is minimized; and
- The entire DFT sequence, $X[\cdot]$, is obtained without latency.

To solve the above OTD-FFT problem, we observe that during the intermediate steps of the Fourier transform computation, not the entire but part of the input data sequence is used. For example, as depicted in Fig. (5.1), when $N = 4$, during the 13th sampling period, the newly-sampled data $x[I(3)] =\ x[12]$ allows the computation of the intermediate sequence $X_1[k]$ for $k = 2,\ 3$, and thereby, $X_2[k]$ for $k =\ 0,\ 1,\ 2,\ 3$ (see the pink

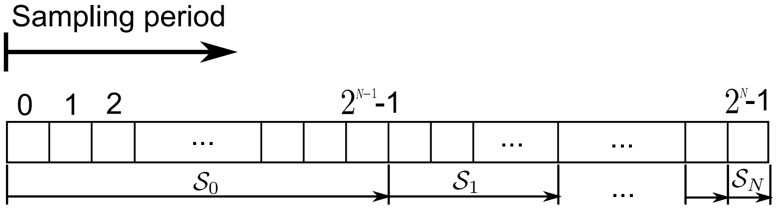

Figure 5.2: Computation stages of the proposed OTD-FFT scheme.

dashed box in Fig. (5.1)), and during the 15th sampling period, the newly-sampled data $x[I(4)] = x[14]$ allows the computation of $X_3[k]$ for $k = 0,\ 1, \ldots,\ 7$ (see the purple dashed box in Fig. (5.1)). Similar pattern can be carried out until the last sampling period of the sequence. This observation implies that the FFT computation can be distributed across the sampling periods when the sequence of $x[\cdot]$ is acquired.

Specifically, the computation of the Fourier transform of the time sequence $\{x[n] \in \Re |\ n = 0,\ 1, \ldots,\ M-1\}$ with $M = 2^N$ can be split into $N+1$ stages, $\mathcal{S}_0,\ \mathcal{S}_1, \ldots,\ \mathcal{S}_N$, where the range of the sampling instants of stage $\mathcal{S}_p$ is given by (see Fig. (5.2)),

$$S_p = \begin{cases} [m_p : m_{p+1} - 1] & \text{for } p = 0,\ 1, \ldots, N-1, \\ M & \text{for } p = N, \end{cases} \tag{5.9}$$

where the index m_p is given by

$$m_p = 2^{N-p}(2^p - 1) \quad \text{for } p = 0,\ 1, \ldots,\ N. \tag{5.10}$$

In each stage $\mathcal{S}_p,\ p =\ 0,\ 1, \cdots,\ N$, only a portion of the intermediate sequences, $X_i[\cdot]$, for $i = 0,\ 1, \cdots, p-1$, are computed by using the already known part of the input sequence, denoted as vector V_p and given by:

$$V_p = \begin{cases} \{x[m_p : m_{p+1} - 1]\} & \text{for } p = 0,\ 1, \ldots, N-1, \\ \{x[M]\}, & \text{for } p = N. \end{cases} \tag{5.11}$$

In each stage, the computation in each sampling period to obtain the corresponding part of the intermediate sequence, $X_p[\cdot]$, is similar. Considering the input sequence with length no less than $2^N = 2^4 = 16$ (In almost all practices, the length of the sampled sequence $N \gg 4$), then in stage $\mathcal{S}_0$, the operation in each sampling period of stage $\mathcal{S}_0$ only amounts to putting the newly sampled input data sequentially into the corresponding location of the intermediate sequence $X_0[\cdot]$ (see Eq. (5.8)), i.e., during the n^{th} sampling period for any n in $m_0 \leq n \leq m_1 - 1$,

$$X_0[\mathbf{I}(n)] = x[n], \tag{5.12}$$

and at the last sampling period of stage $\mathcal{S}_0$, the following part of $X_0[\cdot]$ have already been obtained,

$$X_0[\mathbf{I}(m_0 : m_1 - 1)] = V_0,$$

The calculation procedure in stage $\mathcal{S}_1$ to $\mathcal{S}_N$ is described below.

In stage $\mathcal{S}_1$, the newly-sampled elements in $V_1 = \{x[m_1 : m_2 - 1]\}$ will be put in sequence $X_0[\cdot]$ sequentially in the same manner as in stage $\mathcal{S}_0$. Then during the n^{th} sampling period of stage $\mathcal{S}_1$ for $n = m_1,\ m_1 + 1, \ldots,\ m_2 - 1$, the following part of the sequence $X_1[\cdot]$, $X_1[\mathbf{I}(n) - 1\ \mathbf{I}(n)]$, is obtained via one step calculation as follows,

$$\begin{cases} X_1[\mathbf{I}(n) - 1] = X_0[\mathbf{I}(n) - 1] + w_2^0 X_0[\mathbf{I}(n)], \\ \quad X_1[\mathbf{I}(n)] = X_0[\mathbf{I}(n) - 1] + w_2^1 X_0[\mathbf{I}(n)], \end{cases} \tag{5.13}$$

where ω_2^j for $j = 0,\ 1$ is as defined in Eq. (5.4). Thus, in stage $\mathcal{S}_2$, the known part of $X_1[\cdot]$ from the above computation can be used to compute the following part of sequence $X_2[\cdot]$, $X_2[\mathbf{I}(n) - 3 : \mathbf{I}(n)]$, for $n = \ m_2,\ m_2 + 1, \ldots,\ m_3 - 1$, and recursively, by stage $\mathcal{S}_p$ ($p = \ 2,\ 3, \ldots,\ N - 1$), the following part of the intermediate sequences $X_k[\cdot]$, $k = \ 1,\ 2, \cdots,\ p - 1$, are obtained,

$$\begin{aligned} & X_k[I_i : I_f], \quad \text{for} \\ & I_i = \mathbf{I}(n) - 2^k + 1, \quad I_f = \mathbf{I}(n), \quad \text{and} \\ & n = \ m_k,\ m_k + 1, \ldots,\ m_{k+1} - 1. \end{aligned} \tag{5.14}$$

Thus, in general, during the n^{th} sampling period in stage $\mathcal{S}_p$ for $n = m_p,\ m_p + 1, \ldots,\ m_{p+1} - 1$ and $1 \leq p \leq N - 1$, the newly sampled input data $x[n]$ are passed into $X_0[\mathbf{I}(n)]$ as in Eq. (5.12), and the following part of the sequence $X_p[\cdot]$, $X_p[\mathbf{I}(n) - 2^p + 1 : \mathbf{I}(n)]$, is obtained via p-step calculations below,

$$\begin{aligned} & X_k[\mathbf{I}(n) - 2^k + 1 : \mathbf{I}(n)] = \Phi(X_{k-1}[\mathbf{I}(n) - 2^k + 1 : \mathbf{I}(n)],\ \mathbf{w}_{2k}), \\ & \quad \text{for } k = \ 1,\ 2, \ldots,\ p, \end{aligned} \tag{5.15}$$

where the function $\Phi(\cdot,\cdot)$ is defined in Eq. (5.5), and $X_0[\mathbf{I}(n)]$ for $n = m_p,\ m_p + 1, \ldots,\ m_{p+1} - 1$ are obtained the same manner as in Eq. (5.13).

In stage $\mathcal{S}_N$, i.e., $n = 2^N$, $p = N$ steps calculation is completed—by letting $p = N$ in Eq. (5.15), and the entire sequence $X_N[\cdot]$ is obtained as the FFT of the whole sequence $x[\cdot]$, i.e., $X[\cdot] = X_N[\cdot]$.

Next, we show that the above time-distributed computation scheme is optimal in minimizing the maximal per-sampling-period computational complexity without latency (The computational complexity is discussed later in Sec. 5.2.4).

Lemma 33. *For any given sampled input sequence* $\{x[n] \in \Re|\ n = 0,\ 1, \ldots,\ M - 1,\ M = 2^N,\ N \in \mathbb{N}\}$, *the proposed OTD-FFT algorithm minimizes the maximal per-sampling-period computational complexity without latency.*

Proof. Examining the per-sampling-period computational complexity in the above OTD-FFT algorithm shows that the maximal per-sampling-period computational complexity occurs at the last sampling period–the $(M-1)^{\text{th}}$ sampling period, and equals to $2^{N+1}-2$ multiplications and $2^{N+1} - 2$ additions. Such a computation cannot be further reduced without latency as at any k^{th} sampling period for $0 \leq k \leq M - 1$, the computation using the already available sampled input sequence $x[n]$ for $n = 0,\ 1, \ldots,\ k$ has been completed, i.e., any further computation would require future input data $x[n]$ for $n > k$ that are not yet available. Moreover, all the calculations conducted in the $(M - 1)^{\text{th}}$ sampling period require the newly-sampled data $x[M - 1]$ (see Eq. (5.15)). Thus, the maximal per-sampling-period computational complexity cannot be further reduced. This completes the proof. □

Time-Distributed IFFT

It can be seen—from the above description—that the TD-IFFT can be computed in the same manner by splitting the computation into $N + 1$ stage across the entire sampling sequence. Alternatively, as usually the entire Fourier transformed sequence $X[\cdot]$ is already known a priori, computation of the inverse Fourier transform can be distributed across any $K \leq M (M = 2^N)$ number of sampling periods. Such a time-distributed computation of IFFT is needed in applications such as real-time FD-ILC, where the output of the IFFT—the corresponding time domain control input sequence $x[\cdot]$—is needed not at the last sampling instant of each iteration period, but much earlier. Thus, it is proposed next to compute the TD-IFFT in the first $2N$ sampling periods (for sequence length of 2^N length). The $2N$ periods of computation window is chosen as such a choice not only allows the calculation to be distributed in a pattern corresponding to the butterfly structure of IFFT, but also guarantees that in online applications such as FD-ILC, the per-sampling-period computational complexity in the first $2N$ sampling instants does not exceed the maximal per-sampling period complexity in the OTD-FFT, occurring at the last M^{th} sampling period (see Sec. 5.2.4), even when the OTD-FFT, TD-IFFT, and part of IDFT are concurrently computed (as needed in real-time FD-ILC, see Sec. 5.2.4 later).

Similar as in the OTD-FFT, the computation of TD-IFFT induces $N+1$ number of intermediate sequences successively,

$$x_0[\cdot],\ x_1[\cdot], \ldots,\ x_N[\cdot] \quad \text{with } x_N[\cdot] = x[\cdot]$$

and the intermediate sequence $x_q[\cdot] \in \mathbb{C}^{1\times M}$, for $q = 1,\ 2, \ldots,\ N$, is obtained recursively by

$$x_q[I_i : I_f] = \Phi(x_{q-1}[I_i : I_f], \bar{\mathbf{w}}_{2^q}), \tag{5.16}$$

for $i = 1,\ 2,\ \ldots,\ 2^{N-p}$, where I_i and I_f are given as in Eq. (5.15), respectively, function $\Phi(\cdot,\cdot)$ is defined in Eq. (5.5), and $\bar{\mathbf{w}}_{2^q}$ is the complex conjugate of vector $\mathbf{w}_{2^q}$ defined in Eq. (5.7). Initially,

$$x_0[\mathbf{I}(k)] = X[k]. \tag{5.17}$$

The recursive process above shows that we can obtain $x_1[\cdot],\ x_2[\cdot], \ldots,\ x_q[\cdot]$ sequentially. Thus, the IFFT computation of $X[k]$ can be distributed into the first $2N$ sampling periods, denoted as stage $\mathcal{N}_0$ in Fig. (5.3). At any n^{th} sampling period of stage $\mathcal{N}_0$ for $n =\ 0,\ 1, \ldots,\ 2N-1$, the following part of the intermediate sequence $x_q[\cdot]$ is obtained,

$$x_q[2^{N-1}s : 2^{N-1}(s+1)-1], \tag{5.18}$$

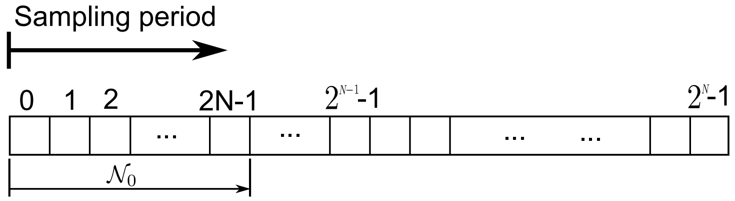

Figure 5.3: Computation stages of the proposed TD-IFFT scheme.

where

$$q = \lceil n/2 \rceil\,, \quad s = \mod (n, 2)$$

and $\lceil \cdot \rceil$ and $\mod (n, m)$ denotes the ceiling function and the modulo function, respectively. Thus, the first half of the sequence $x_q[\cdot]$ is obtained via Eq. (5.16) when n is even, while the second half is obtained via the same equation in the next sampling period. Therefore, at the last sampling period of stage $\mathcal{N}_0$, the whole corresponding time domain sequence $x[\cdot]$ is obtained.

5.2.4 Application: Real-time Iterative Learning Control

In this section, first, an online implementation of the FD-ILC algorithm (5.1) is developed based on the proposed OTD-FFT/TD-IFFT, then the computational complexity of the proposed OTD-FFT/TD-IFFT and FD-ILC algorithms are analyzed.

OTD-FFT/TD-IFFT based Online ILC Implementation

To implement the FD-ILC (5.1) online, we propose to split the required FFT/IFFT computation of each iteration into two parts: First, during the i^{th} iteration, the OTD-FFT is applied to obtain the Fourier transform of the output measured in the i^{th} iteration, $Y_k[\cdot]$, and then at the last sampling period of the i^{th} iteration, the frequency domain input $U_{k+1}[\cdot]$ for the $(i+1)^{\text{th}}$ iteration is obtained via Eq. (5.1); Secondly, during the first $2N$ sampling periods of the $(i+1)^{\text{th}}$ iteration, the proposed TD-IFFT is applied to obtain the time domain input sequence for the $(i+1)^{\text{th}}$ iteration, $u_{k+1}[\cdot]$, and at the $(2N-1)^{\text{th}}$ sampling period, the entire input sequence $\{u_{k+1}[n] \in \Re|\ n = 0,\ 1, \ldots,\ M-1\}$ is obtained. To obtain the iterative control input $u_{i+1}[n]$ for the first

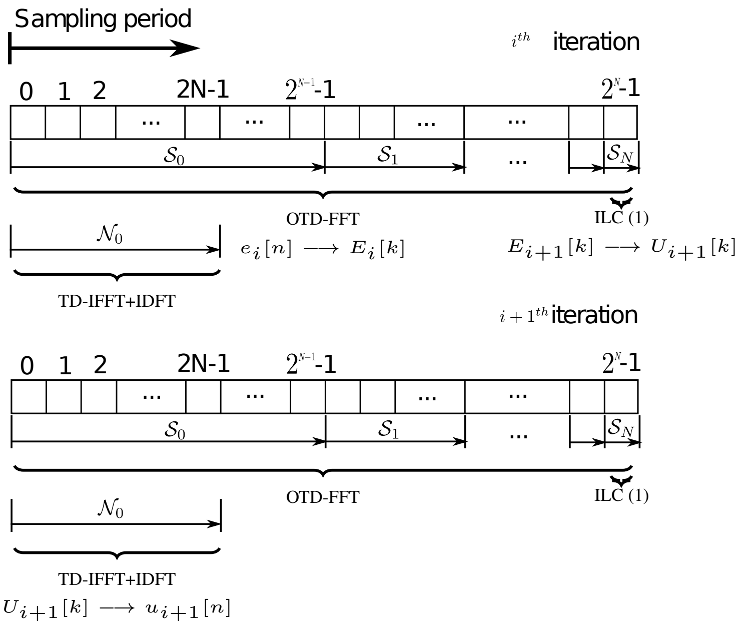

Figure 5.4: Illustration of the proposed FD-ILC computation scheme based on the proposed OTD-FFT/TD-IFFT algorithms.

$2N-1$ sampling period, the following direct IDFT is applied

$$u_{i+1}[n] = \sum_{k=0}^{2^N-1} U_{i+1}[k] w_M^{-kn}, \quad \text{for } n = 0,\ 1, \ldots,\ 2N-1. \tag{5.19}$$

Note that although this portion of the input, $u_{i+1}[n],\ n = 0,\ 1, \ldots,\ 2N-1$, are computed twice, the total per-sampling-period computational complexity of the FD-ILC is still much lower than that when using the conventional FFT/IFFT (see the complexity analysis below). We also note that the computation of the IFFT might be distributed into the first $K < 2N$ sampling periods so that less number of input values need to be computed via IDFT (5.19) directly. As a result, the total computational complexity of the entire iteration trial might be reduced. Such an approach, however, will complicate the computation scheduling, and can increase the per-sampling-period complexity.

To continuously implement this scheme online, in each iteration, the computation in stage $\mathcal{S}_0$ of the OTD-FFT computation is partially overlapped with that of the TD-IFFT in stage $\mathcal{N}_0$ (see Fig. (5.4)). Specifically, the steps to implement the proposed computation algorithm of FD-ILC are given as follows (see Fig. (5.5)):

Step 1 (Initialization): Set the number of sampling periods to 2^N; Compute and store the data that can be computed offline (e.g. $\mathbf{w}_{2^N}$, $I[k]$); And choose the initial input for

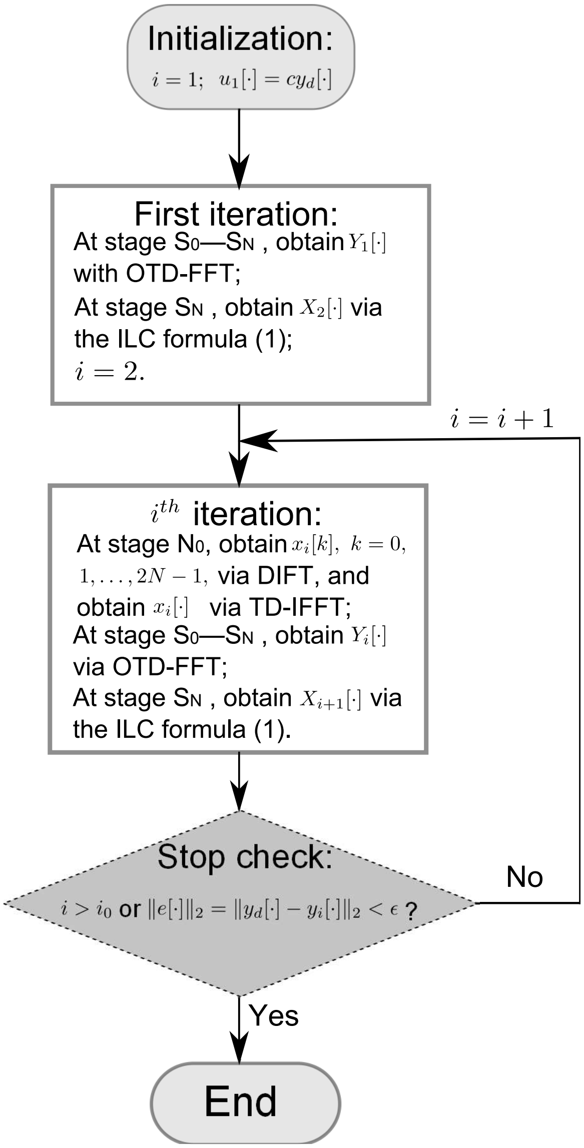
Initialization:
$i = 1;\ \ u_1[\cdot] = cy_d[\cdot]$
First iteration:
At stage S0—SN , obtain $Y_1[\cdot]$ with OTD-FFT;
At stage SN , obtain $X_2[\cdot]$ via the ILC formula (1);
$i = 2.$
$i = i + 1$
i^{th} iteration:
At stage N0, obtain $x_i[k],\ k = 0, 1, \ldots, 2N-1,$ via DIFT, and obtain $x_i[\cdot]$ via TD-IFFT;
At stage S0—SN , obtain $Y_i[\cdot]$ via OTD-FFT;
At stage SN , obtain $X_{i+1}[\cdot]$ via the ILC formula (1).
Stop check:
$i > i_0$ or $\|e[\cdot]\|_2 = \|y_d[\cdot] - y_i[\cdot]\|_2 < \epsilon$?
No
Yes
End

Figure 5.5: Flowchart of the proposed FD-ILC algorithm via OTD-FFT/TD-IFFT.

the first iteration trial, e.g., set $u_1[\cdot] = cy_d[\cdot]$, where $y_d[\cdot]$ is the desired trajectory to be tracked, and c is the DC-gain of the system.

Step 2 (First iteration): During each $1 \leq n < N$ sampling period of the first iteration, apply the corresponding input $u_1[n]$, then execute the sequential computation of the OTD-FFT from stage $\mathcal{S}_1$ to $\mathcal{S}_N$ to obtain the corresponding Fourier transform of the output, $Y_1[\cdot]$; Then, at stage $\mathcal{S}_N$, i.e., the last sampling period, the ILC formula in Eq. (5.1) is executed to obtain the FFT of the input sequence for the next iteration, $U_2[\cdot]$. Go to Step 4.

Step 3 (The i^{th} iteration): In each i^{th} ($i \geq 2$) iteration, during each n^{th} sampling period of stage $\mathcal{N}_0$ for $n = \ 0,\ 1, \ldots,\ 2N-1$, obtain the time domain input $u_i[n]$ via inverse DFT and apply it to the system, and apply the TD-IFFT algorithm until the whole input sequence $u_i[\cdot]$ is obtained at the last sampling period of stage $\mathcal{N}_0$. Then the OTD-FFT is utilized from stage $\mathcal{S}_1$ to $\mathcal{S}_N$ to obtain the frequency domain data $Y_i[\cdot]$; during the state $\mathcal{S}_N$, the ILC formula in Eq. (5.1) is executed to obtain the Fourier transformed sequence of the input for the next iteration, $U_{i+1}[\cdot]$.

Step 4 (Stop condition): If the iteration counts $i \geq i^*$ (i^*: the pre-chosen number of iterations), or the tracking error $e[k] = y_d[k] - y_i[k]$ satisfying $\|e[\cdot]\|_2 \leq \epsilon$ (the pre-chosen desired tracking precision), then stop; otherwise, $i = i+1$, and go back to Step 3.

Computational Complexity Analysis

The analysis next is focused on the per-sampling-period computational complexity of the above algorithms.

The per-sampling-period computational complexity (only consider the complex multiplication and addition in the computation) of the classical radix-2 FFT and IFFT algorithms are, respectively, $2^N \log 2^N$ multiplications and $2^N \log 2^N$ additions [85]. With the proposed OTD-FFT, the per-sampling-period computational complexity in stage $\mathcal{S}_i$, for $i = 0,\ 1, \ldots,\ N$, is reduced to $2^{i+1} - 2$ multiplications and $2^{i+1} - 2$ additions. Therefore, the maximal per-sampling-period computational complexity is at $2^{N+1} - 2$ multiplications and $2^{N+1} - 2$ additions, occurring at the last sampling period of stage $\mathcal{S}_N$. With the TD-IFFT, the computation of IFFT is completed in stage $\mathcal{N}_0$, and the complexity in each sampling period is the same during stage $\mathcal{N}_0$, and equals at 2^{N-1} multiplications and 2^{N-1} additions.

Finally, comparison of the proposed OTD-FFT/TD-FFT with the conventional FFT/IFFT shows that except being distributed across the sampling periods, the computations executed in the OTD-FFT and TD-FFT are, in fact, exactly the same as those in the conventional FFT/IFFT. Thus, the proposed OTD-FFT/TD-FFT does not increase but maintain the total computational complexity.

Table 5.1: Comparison of the maximal per-sampling-period computational complexity.

Algorithm	Multiplications	Additions
FFT/IFFT	$2^N \log 2^N$	$2^N \log 2^N$
OTD-FFT	$2^{N+1} - 2$	$2^{N+1} - 2$
TD-IFFT	2^{N-1}	2^{N-1}
FD-ILC (FFT+IFFT)	$2 \times 2^N \log 2^N + 2^N$	$2 \times 2^N \log 2^N + 2^{N+1}$
FD-ILC (OTD-FFT+TD-IFFT+IDFT)	$3 \times 2^{N+1} - 2$	$2^{N+2} - 2$

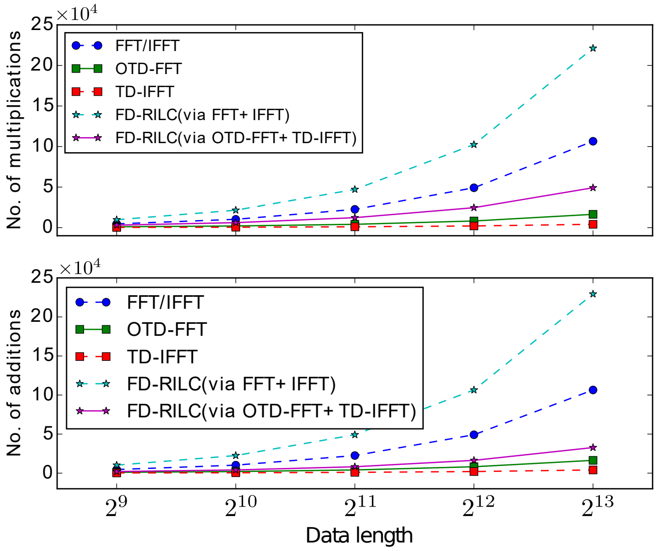

Figure 5.6: Comparison of the maximal per-sampling-period computational complexity for $N = 9, 10, 11, 12, 13$.

To quantify the computational complexity of the proposed FD-ILC algorithm, we note that IDFT is also used to obtain part of the input, $u_i[n]$ for $n = 0,\ 1,\ \ldots,\ 2N - 1$, for immediate implementation, adding another 2^N multiplications and $2^N - 1$ additions in each sampling period of stage $\mathcal{N}_0$. As a result, the total computation in the implementation of the FD-ILC in each iteration trail should account for the computation of the OTD-FFT, TD-IFFT, and the IDFT computation (for those sampling periods in stage $\mathcal{N}_0$), and the frequency domain ILC computation (for stage $\mathcal{S}_N$) all combined. The maximal per-sampling-period computational complexity occurs at the last sampling period of each iteration, and for $Q[j\omega_k] = 1$ in the ILC algorithm (5.1), equals to $3 \times 2^N - 2$ multiplications and $2^{N+2} - 2$ additions—a dramatic reduction from $2 \times 2^N \log 2^N + 2^N$ multiplications and $2 \times 2^N \log 2^N + 2^{N+1}$ additions when using the classic radix-2 FFT and IFFT in the FD-ILC implementation. The comparison of the maximal per-sampling-period computational complexity is shown in Table. 5.1 and Fig. (5.6).

Complexity comparison in Fig. (5.6) clearly reveals the efficacy of the proposed OTD-FFT/TD-FFT for online applications. Particularly, as the data length increases, the maximal per-sampling-period computational complexity increases linearly and thereby, relatively very slow for the proposed OTD-FFT/TD-FFT, whereas increases *exponentially*, and thereby, much faster for the conventional FFT/IFFT. For example, at the data length of 2^{13} ($2^{13} = 8192$), the maximal per-sampling-period complexity of the OTD-FFT is 5.5 times lower than (i.e., 15.4 % of) that of the conventional FFT. As shown in Fig. (5.6), the maximal per-sampling-period computational complexity of the proposed FD-ILC–with both the OTD-FFT, TD-FFT, and partial IDFT computed simultaneously–is still lower than that of the conventional FFT/IFFT alone—1.2 times lower than that of the conventional FFT/IFFT alone at the data length of 2^{13} (8192). Such a dramatic reduction in the maximal per-sampling-period computational complexity is very much needed in online control applications and other signal processing applications on general-purpose microprocessors.

5.3 Experimental Implementation

The experimental implementation consisted of two parts: We first demonstrated the proposed OTD-FFT/TD-FFT technique in real-time signal processing, then further illustrated the implementation of the proposed real-time FD-ILC technique in output tracking control on a piezoactuator of an AFM system.

All the programs employed in the experiments were designed in Matlab/Simulink, and then downloaded and executed online on a PC system (CPU: Intel Xeon @2.4GHz) under the Matlab-xPC-target real-time operation system environment through a data acquisition system (PCI 6259, National Instrument Inc.).

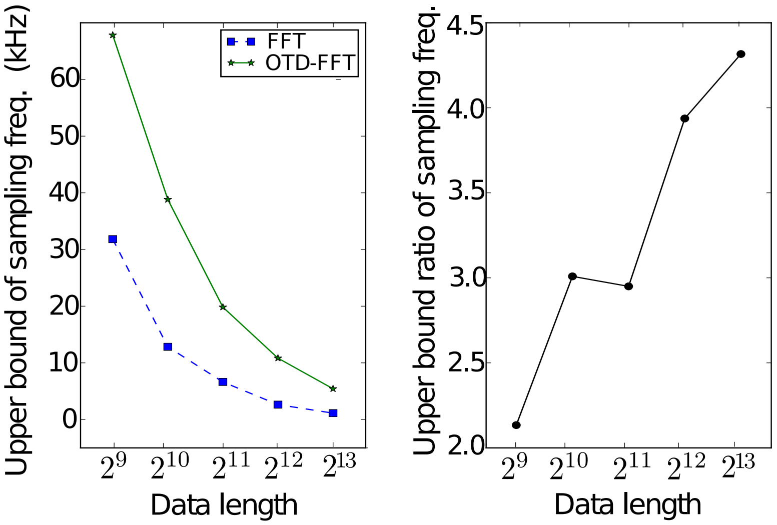

Figure 5.7: (a) Comparison of the upper bound of the sampling frequency (Fs) of the proposed OTD-FFT vs. the conventional FFT measured in the experiment using the white-noise data sequence of different lengths (Dl); and (b) the ratio of upper bound of the sampling frequency (Fs) of the proposed OTD-FFT to that of the conventional FFT.

5.3.1 Implementation of OTD-FFT/TD-IFFT, Results & Discussion

First, we tested and evaluated the online computational efficiency of the proposed OTD-FFT: An a priori generated white-noise of given length, $\{w_o[n] | n = 0,\ 1, \cdots,\ M-1\}$ with $M = 2^N$, was sent out and directly acquired back via the D/A and A/D conversion of the DAQ system, respectively. During these M sampling periods, the proposed OTD-FFT was employed to obtain the Fourier transform of the acquired input signal, $W_i[k],\ k =\ 0,\ 1, \ldots,\ M-1$. For comparison, the classical FFT was also implemented, i.e., the FFT was computed within the sampling period right after the entire discretized sequence of the white-noise signal was sent out and acquired back completely. The upper bounds of the sampling frequency–by keeping the length of the sampled white-noise sequence fixed–were experimentally tested as the supremium at which the CPU was overloaded and the program was halted. The tested upper bounds of the sampling frequency for these two cases are compared in Fig. (5.7) for different lengths of sampled sequence.

The experimental results showed that by using the proposed OTD-FFT technique, the computational efficiency in online signal processing was significantly improved—by using the proposed OTD-FFT algorithm, the upper bounds of the sampling frequencies were substantially higher than those when using the classical FFT. Also, from Fig. (5.7b), we can see that the computational efficiency improvement, measured by the ratio of the sampling frequency upper bound of the proposed OTD-FFT to that of the conventional FFT, became more significant as the length of the data increased—by using the proposed OTD-FFT method, the upper bound of the sampling frequency was increased over 3.3 times when using the proposed OTD-FFT method at the data length of 2^{13} (8192). Such an trend of increase well agreed with the theoretical prediction. Secondly, we further evaluated the online computational efficiency when implementing the proposed OTD-FFT and TD-IFFT together: First, the OTD-FFT was applied to obtain the Fourier transform of the white-noise signal $\{w_o[n]|\ n = 0,\ 1, \ldots,\ M-1\}$, $W_i[\cdot]$ in the same way as in the first experiment. Then in the following $2N$ sampling periods, the proposed TD-IFFT was applied to the obtained Fourier transform sequence $W_i[\cdot]$ to obtain the "converted" time sequence $\{w_i[n]|\ n = 0,\ 1, \ldots,\ M-1\}$. The upper bounds of the sampling frequency for different lengths of the sampled white-noise were experimentally tested as the supremium at which the CPU was overloaded and the program was halted. For comparison, the classical FFT+IFFT was also implemented, i.e., the FFT+IFFT was computed within the sampling period right after the white-noise signal was sent out and acquired back completely. The tested upper bounds of the sampling frequency for these two cases are compared in Fig. (5.8) for different lengths of the sampled sequence. Similarly, the upper bounds of the data length–for fixed sampling frequency–were also experimentally tested, and the tested upper bounds of the data length are compared in Fig. (5.9).

The experimental results showed that by using the proposed OTD-FFT+TD-IFFT algorithms, the computational efficiency in online signal processing was significantly improved. By using the proposed OTD-FFT+TD-IFFT algorithms, the upper bounds of the sampling frequencies were substantially higher than those of the classical FFT+IFFT. For example, the upper bound of the sampling frequency was increased over 3.1 times when using the proposed OTD-FFT+TD-IFFT method for the sampled data of length of 1024 (improved from 7.1kHz to 29kHz, see Fig. (5.8)). Such an improvement can

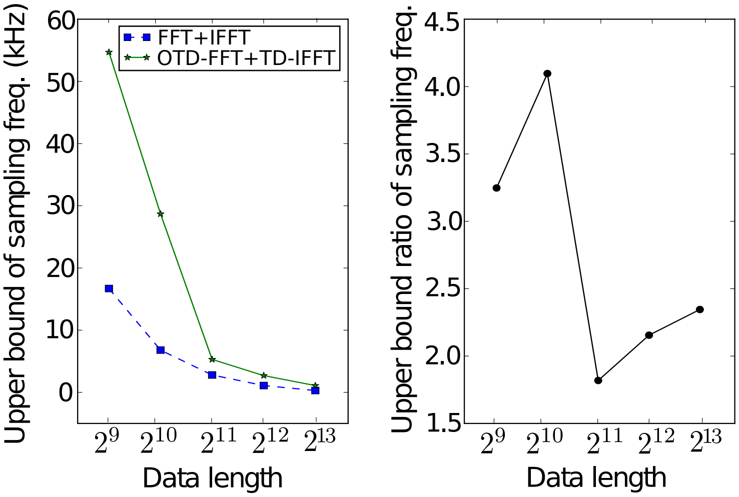

Figure 5.8: (a) Comparison of the upper bound of the sampling frequency (Fs) of the proposed OTD-FFT+TD-IFFT vs. the conventional FFT+IFFT measured in the experiment using the white-noise data sequence of different lengths (Dl); and (b) the ratio of the upper bound of the sampling frequency (Fs) of the proposed OTD-FFT+TD-IFFT to that of the conventional FFT+IFFT.

also be seen from the increase of the upper bounds of the achievable data length for fixed sampling frequency. For example, at the sampling frequency of 16kHz, by using the proposed OTD-FFT+TD-IFFT method, the upper bound of the data length was increased over 3 times (improved from 256 to 1024, see Fig. (5.9)) at the sampling frequency of 16 kHz.

We note that the improvement reduced when the data length was further increased to 2048 (2^{11}), 4096 (2^{12}), and 8192 (2^{13}) (see Fig. (5.8)) or when the sampling frequency was further decreased (see Fig. (5.9)), which was lower than those theoretical predictions (see Fig. (5.6)). Such a difference–between the experimentally-achieved and the theoretically-predicted improvements–might be caused by two factors: First, the theoretical prediction does not take into account the time consumed by other operations involved, including the A/D and D/A conversion and the memory read/write time. The portion of the time spent on these operations can increase significantly as the length of the sampled sequence increased. Secondly, the OTD-FFT+TD-IFFT algorithm was programed and executed via the Simulink environment whereas the FFT+IFFT functions were executed via the built-in program of Matlab–much more

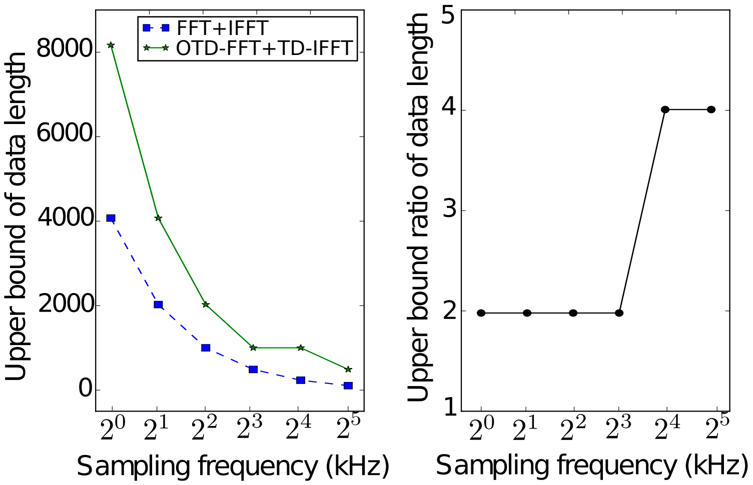

Figure 5.9: (a) Comparison of the upper bound of the data length (Dl) of the proposed OTD-FFT+TD-IFFT vs. the conventional FFT+IFFT measured in the experiment using the white-noise data sequence of different sampling frequency (Fs); and (b) the ratio of the upper bound of the data length (Dl) of the proposed OTD-FFT+TD-IFFT to that of the conventional FFT+IFFT.

efficient than Simulink programs of user-defined script functions. Therefore, the efficacy of the proposed OTD-FFT+TD-IFFT technique is very promising for high-speed online signal-processing applications.

5.3.2 Implementation of Real-time FD-ILC, Results & Discussion

Next, to demonstrate the proposed technique to real-time FD-ILC, the following inversion-based ILC algorithm [78]

$$U_i[j\omega_k] = U_{i-1}[j\omega_k] + G[j\omega_k]^{-1} e_{i-1}[j\omega_k], \tag{5.20}$$

was implemented in high-speed online trajectory tracking on a piezoelectric actuator of an AFM system (Dimension ICON, Bruker-Nano Inc.). The first resonant frequency and the bandwidth of the piezoelectric actuator were at ∼5.0kHz and ∼3.0kHz, respectively, as can be seen from the frequency response of the system measured through experiment, shown in Fig. (5.10). Note that the general ILC in (5.1) is reduced to the above inversion-based iterative learning control (IIC) by choosing $Q[j\omega_k] = 1$ and replacing $L[j\omega_k]$ with the inverse of the frequency response $G[j\omega_k]$.

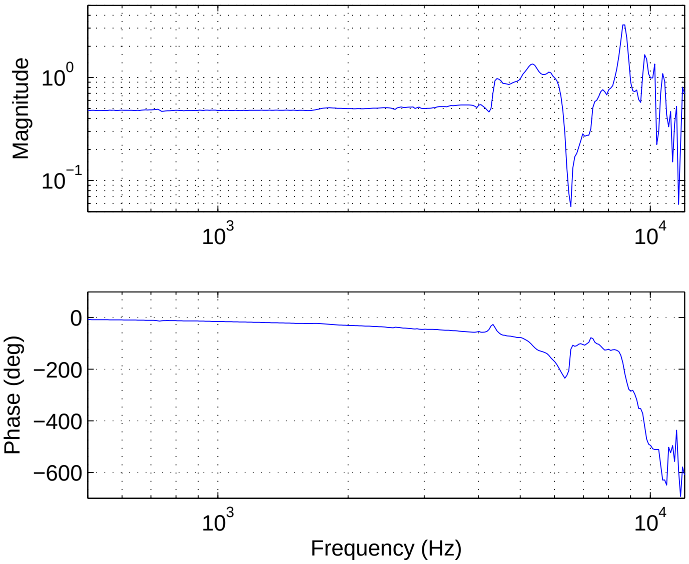

Figure 5.10: The frequency response of the piezoelectric actuator measured through experiment.

The proposed real-time FD-ILC technique as detailed in Sec. 5.2.4 was implemented to track a two-period triangle signal of different triangle rates (500 Hz, 650 Hz, and 800 Hz), where the experimentally measured frequency response (as a sequence of complex numbers) was used in the FD-ILC law (5.20) directly, and the effects of the system variation and disturbances were accounted for by setting $U_i[j\omega_k] = U_{i-1}[j\omega_k]$ at frequencies where the iterative input updated term, $G[j\omega_k]e_{i-1}[j\omega_k]$, was less than the threshold value (which was determined based on the noise/disturbance level). The sequence of the input/output data was kept at the same length of $1024(2^{10})$ for the three triangle rates, and the iterative control input obtained was immediately applied one iteration after the other. The same initial condition for each iteration was maintained by having a long enough zero-extension at both the beginning and the end of the trajectory in all three speeds of the desired trajectories (see Fig. (5.11a), (5.12b), and (5.13a)). The upper-bound of the sampling frequency was measured by keeping on increasing the sampling frequency (until the CPU overload occurred). The iteration was stopped when the tracking error (measured in RMS sense) cannot be further reduced. Then the practically converged tracking results were acquired by using the highest-achievable sampling frequency for the above three triangle rates. For comparison, the upper-bounds of the

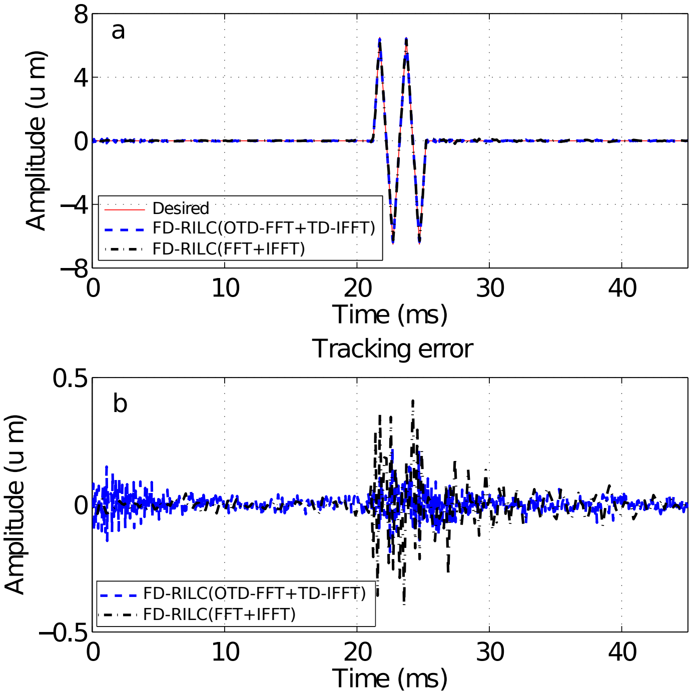

Figure 5.11: Comparison of (a) the tracking results and (b) the tracking error by using the proposed FD-ILC (via OTD-FFT+TD-IFFT) with those by using the FD-ILC (via FFT+IFFT) for the trajectory of triangle rate at 500 Hz.

sampling frequency when using the FD-ILC via the conventional FFT+IFFT to track the above three triangle trajectories were also tested in the experiment, and the converged tracking results under the highest-achievable sampling frequency were acquired. In each cases (the proposed FD-ILC and the conventional FD-ILC), the frequency response measured at the corresponding sampling frequency was used in the computation of the iterative control input via Eq. (5.20).

Table 5.2: Comparison of the relative RMS tracking error by using the two FD-ILC schemes for different triangle rates.

Triangle rate (Hz)	500	650	800
FD-ILC (%) (via OTD-FFT+TD-IFFT)	3.25	5.36	5.44
FD-ILC (%) (via FFT+IFFT)	7.34	24.00	29.62

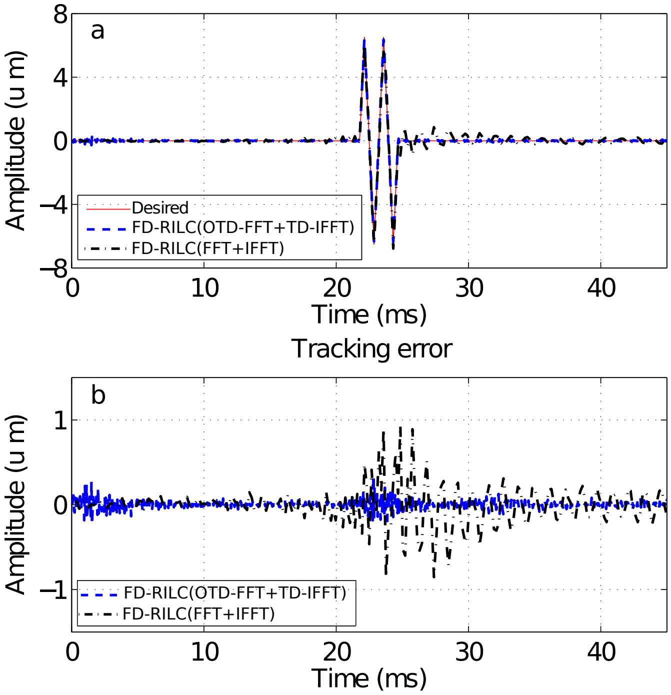

Figure 5.12: Comparison of (a) the tracking results and (b) the tracking error by using the proposed FD-ILC (via OTD-FFT+TD-IFFT) with those by using the FD-ILC (via FFT+IFFT) for the trajectory of triangle rate at 650 Hz.

The experimentally-measured upper bound of the sampling frequency was at 22 kHz and 6.8 kHz for the proposed FD-ILC and the FD-ILC via conventional FFT+IFFT, respectively. The converged tracking results obtained by these two implementation schemes are compared in Fig. (5.11), Fig. (5.12), and Fig. (5.13) for the triangle trajectory at rate of 500 Hz, 650 Hz, and 800 Hz, respectively. The relative RMS tracking errors of the above three triangle trajectories obtained by using these two methods are also compared in Table 5.2.

The experimental tracking results showed that high-speed precision trajectory tracking can be achieved by using the proposed method. As shown in Table 5.2, the tracking performance of the FD-ILC was significantly improved when using the proposed OTD-FFT+TD-IFFT over the conventional FFT+IFFT algorithms. For example, even for the high-speed trajectory of triangle rate at 800Hz, the relative RMS error was only

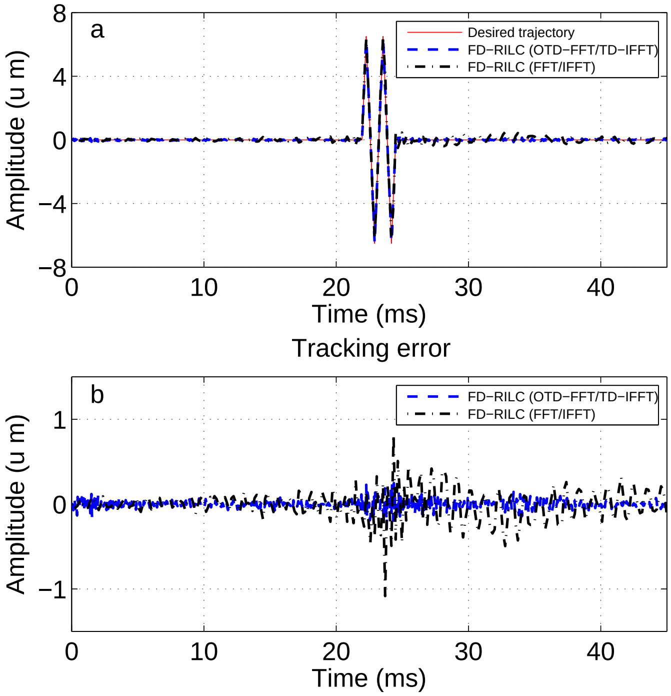

Figure 5.13: Comparison of (a) the tracking results and (b) the tracking error by using the proposed FD-ILC (via OTD-FFT+TD-IFFT) with those by using the FD-ILC (via FFT+IFFT) for the trajectory of triangle rate at 800 Hz.

5.44%, over 4 times lower than that when using the conventional FFT+IFFT. Such an increase of tracking performance was mainly benefited from the higher sampling frequency enabled by using the proposed OTD-FFT+TD-FFT technique, as by using the proposed FD-ILC scheme, the achievable sampling frequency was over 3 times higher than that by using the conventional FFT+IFFT based FD-ILC scheme (22 kHz vs. 6.8 kHz). The higher sampling frequency allowed not only more higher-order harmonic frequencies of the tracking trajectory to be acquired into the DAQ system, but also better (i.e., more accurate) representation of the system dynamics (i.e., the frequency response) in the FD-ILC algorithm (5.20), resulting in better compensation for the tracking error by the proposed FD-ILC. For example, as shown in Fig. (5.14), at the sampling frequency of 22 kHz, or equivalently, the Nyquist frequency of 11 kHz, most of the spectrum of the 800 Hz two-period triangle trajectory can be acquired without aliasing. Whereas at the sampling frequency of 6.8 kHz, a significant part of

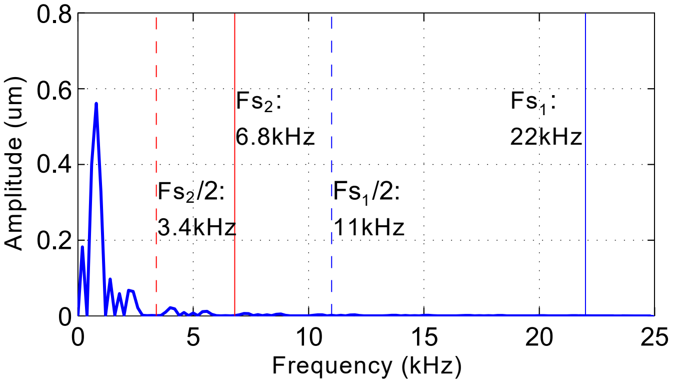

Figure 5.14: The spectrum of the 800 Hz triangle trajectory with respect to the sampling frequency achieved by using the proposed approach (Fs_1: 22 kHz) and the conventional FFT/IFFT-based approach (Fs_2: 6.8 kHz), and the corresponding Nyquist frequencies ($Fs_1/2$ and $Fs_2/2$), respectively.

the frequency components of the trajectory exceeded the corresponding Nyquist frequency of 3.4 kHz, resulting in aliasing issue in both the tracking results acquired. Similarly, such a decrease of Nyquist frequency (from 11 kHz to 3.4 kHz) also led to frequency aliasing in the system frequency response $G[j\omega_k]$ that can be utilized by the FD-ILC algorithm for output tracking. Such an aliasing-caused tracking error (due to the limited sampling frequency achieved) became more pronounced as the triangle rate increased, as shown in Table 5.2. Finally, for trajectories such as triangle trajectory of relatively narrow frequency spectrum, the online computation efficiency can be further improved by combining the Goertzel algorithm [86] with the proposed technique (by distributing the computation for a specific range of frequencies in a similar pattern as the proposed OTD-FFT/TD-IFFT). However, the proposed approach is equally effectively for tracking more general trajectories with relatively broad spectrum. Therefore, the experimental results clearly illustrated the superior performance of the proposed OTD-FFT+TD-FFT over the conventional FFT+IFFT in real-time implementation of the FD-ILC techniques and other real-time signal processing applications.

5.4 Conclusion

The OTD-FFT/TD-IFFT technique was developed to minimize the per-sampling-period computational complexity in online FFT implementation. The proposed OTD-FFT approach distributed the computation load by exploiting the butterfly structure in computing FFT/IFFT to each single sampling period, and minimized the per-sampling-period computational complexity without latency while maintaining the total computational complexity. The proposed OTD-FFT/TD-IFFT algorithm was utilized in online FD-ILC implementation. Moreover, the computational complexity of the proposed OTD-FFT/TD-IFFT and the FD-ILC was analyzed. Experimental implementation of the proposed OTD-FFT/TD-IFFT and its application to the FD-ILC technique demonstrated that the proposed approach was effective in online control applications.

Chapter 6

Conclusion

In this dissertation, optimal and learning-based output tracking and tracking-transition techniques were proposed for applications in high-speed probe-based nanofabrication. Particularly, we proposed an approach to extend the previous work that attains smooth output transition and smooth tracking/transition switching to further achieve amplitude-constrained input-energy minimization and transition time minimization. Then, an offline learning based control technique was developed to compensate for both hysteresis and dynamics in hysteresis-Hammerstein systems. Moreover, for the trajectory decomposition problem arose in the offline learning based control technique, we further developed an asymptotic online trajectory decomposition by only using one type of basis function without truncation. For the computation issue occurred in the real-time implementation of frequency-domain iterative learning control (FD-ILC) in high-speed nanofabrication, a real-time FD-ILC based on an optimal time-distributed FFT/IFFT was developed. The main contributions of this dissertation include:

1. A multi-objective optimization technique to trajectory design and tracking with non-periodic tracking-transition switching was proposed for non-minimum phase systems. The proposed approach extended the previous work on trajectory design and tracking of non-periodic tracking-transition switching to further minimize the input energy under amplitude constraint and then the transition time. It was shown that the amplitude-constrained input-energy minimization problem can be converted to an unconstrained one. Then the optimal parameters in designing the transition output trajectory were sought through an improved conjugate gradient method, and the minimal transition time was further obtained via one dimensional search.

2. The superposition of Hammerstein systems has been studied through the development of the almost superposition principle of Hammerstein systems. It has been shown that the superposition error can be rendered arbitrarily small, i.e., the almost superposition holds for Hammerstein systems, provided that the number of output elements is large enough. A realization of the ASHS for output tracking was proposed by combining uniform B-splines for trajectory decomposition with an inverse Preisach model for input synthesis. Two optimization schemes were further proposed to enhance the implementation of the ASHS in practice. Moreover, the proposed ASHS was implemented for simultaneous compensation of hysteresis and dynamics in precision output tracking. Experiments were performed to control the piezoactuator on an AFM system in different frequency range, and the results demonstrated the efficacy of the proposed method.

3. A trajectory decomposition method using single basis was proposed for the applications in control. First, a trajectory decomposition method based on a least square minimization process was proposed and the decomposition error was proved to be arbitrarily small as the decomposition knot period goes to zero. Then, we showed that by adding a zero period at the beginning and the end of the trajectory, the first few coefficients can be ignored, thus, the truncation of the basis elements can be avoided. Furthermore, we proposed a sectional interactive decomposition algorithm (SID) to decompose a previewed trajectory with only one basis function. The numerical results demonstrated the efficacy of the proposed method.

4. The OTD-FFT/TD-IFFT technique was developed to minimize the per-sampling-period computational complexity in online FFT implementation. The proposed OTD-FFT approach distributed the computation load by exploiting the butterfly structure in computing FFT/IFFT to each single sampling period, and minimized the per-sampling-period computational complexity without latency while maintaining the total computational complexity. The proposed OTD-FFT/TD-IFFT

algorithm was utilized in online FD-ILC implementation. Moreover, the computational complexity of the proposed OTD-FFT/TD-IFFT and the FD-ILC was analyzed. Experimental implementation of the proposed OTD-FFT/TD-IFFT and its application to the FD-ILC technique demonstrated that the proposed approach was effective in online control applications.

References

[1] J. Ren and Q. Zou, “High-speed adaptive contact-mode atomic force microscopy imaging with near-minimum-force,” *Review of Scientific Instruments*, vol. 85, no. 7, p. 073706, 2014.

[2] A. Cavalcanti, “Assembly automation with evolutionary nanorobots and sensor-based control applied to nanomedicine,” *Nanotechnology, IEEE Transactions on*, vol. 2, no. 2, pp. 82–87, 2003.

[3] D. J. Hoelzle, A. G. Alleyne, and A. J. W. Johnson, “Basis task approach to iterative learning control with applications to micro-robotic deposition,” *Control Systems Technology, IEEE Transactions on*, vol. 19, no. 5, pp. 1138–1148, 2011.

[4] A. Gasparetto and V. Zanotto, “A new method for smooth trajectory planning of robot manipulators,” *Mechanism and Machine Theory*, vol. 42, no. 4, pp. 455–471, 2007.

[5] S. Devasia, “Design of feedforward input for output-settling control with dual-stage actuators,” *Mechatronics, IEEE/ASME Transactions on*, vol. 12, no. 6, pp. 670–679, 2007.

[6] H. Perez and S. Devasia, “Optimal output-transitions for linear systems,” *Automatica*, vol. 39, no. 2, pp. 181–192, 2003.

[7] H. Perez, Z. Qingze, and S. Devasia, “Design and control of optimal scan trajectories: Scanning tunneling microscope example,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 126, no. 1, pp. 187–197, 2004.

[8] K. S. Kim, Z. Qingze, and S. Chanmin, “A new approach to scan-trajectory design and track: AFM force measurement example,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 130, no. 5, 2008.

[9] H. Wang, Q. Zou, and H. Xu, “Inversion-based optimal output tracking–transition switching with preview for nonminimum-phase linear systems,” *Automatica*, vol. 48, no. 7, pp. 1364–1371, 2012.

[10] L. Jetto, V. Orsini, and R. Romagnoli, “Accurate output tracking for nonminimum phase nonhyperbolic and near nonhyperbolic systems,” *European Journal of Control*, vol. 20, no. 6, pp. 292–300, 2014.

[11] D. Croft, G. Shed, and S. Devasia, “Creep, hysteresis, and vibration compensation for piezoactuators: atomic force microscopy application,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 1, pp. 35–43, 2001.

[12] G. Song, J. Zhao, X. Zhou, D. Abreu-Garcia *et al.*, "Tracking control of a piezoceramic actuator with hysteresis compensation using inverse Preisach model," *Mechatronics, IEEE/ASME Transactions on*, vol. 10, no. 2, pp. 198–209, 2005.

[13] X. Tan and J. S. Baras, "Adaptive identification and control of hysteresis in smart materials," *Automatic Control, IEEE Transactions on*, vol. 50, no. 6, pp. 827–839, 2005.

[14] S. Xiao and Y. Li, "Modeling and high dynamic compensating the rate-dependent hysteresis of piezoelectric actuators via a novel modified inverse Preisach model," *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 5, pp. 1549–1557, 2013.

[15] W. Lei, J. Mao, and Y. Ma, "A new modeling method for nonlinear rate-dependent hysteresis system based on LS-SVM," in *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on.* IEEE, 2008, pp. 1442–1446.

[16] P. K. Wong, Q. Xu, C. M. Vong, and H. C. Wong, "Rate-dependent hysteresis modeling and control of a piezostage using online support vector machine and relevance vector machine," *Industrial Electronics, IEEE Transactions on*, vol. 59, no. 4, pp. 1988–2001, 2012.

[17] M. Al Janaideh and P. Krejci, "Inverse rate-dependent Prandtl–Ishlinskii model for feedforward compensation of hysteresis in a piezomicropositioning actuator," *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 5, pp. 1498–1507, 2013.

[18] K. K. Leang and S. Devasia, "Design of hysteresis-compensating iterative learning control for piezo-positioners: Application to atomic force microscopes," *Mechatronics*, vol. 16, no. 3, pp. 141–158, 2006.

[19] Y. Wu and Q. Zou, "Iterative control approach to compensate for both the hysteresis and the dynamics effects of piezo actuators," *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 5, pp. 936–944, 2007.

[20] C. Ong, Y. Wong, H. Loh, and X. Hong, "An optimization approach for biarc curve-fitting of B-spline curves," *Computer-Aided Design*, vol. 28, no. 12, pp. 951–959, 1996.

[21] W. Ma and J. P. Kruth, "Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces," *Computer-Aided Design*, vol. 27, no. 9, pp. 663–675, 1995.

[22] H. Kano, H. Nakata, and C. F. Martin, "Optimal curve fitting and smoothing using normalized uniform B-splines: a tool for studying complex systems," *Applied Mathematics and Computation*, vol. 169, no. 1, pp. 96–128, 2005.

[23] J. Bolder, B. Lemmen, S. Koekebakker, T. Oomen, O. Bosgra, and M. Steinbuch, "Iterative learning control with basis functions for media positioning in scanning inkjet printers," in *Intelligent Control (ISIC), 2012 IEEE International Symposium on.* IEEE, 2012, pp. 1255–1260.

[24] D. J. Hoelzle and K. Barton, “Flexible iterative learning control using a library based interpolation scheme,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on.* IEEE, 2012, pp. 3978–3984.

[25] J. Bolder and T. Oomen, “Rational basis functions in iterative learning control with experimental verification on a motion system,” *Control Systems Technology, IEEE Transactions on*, vol. 23, no. 2, pp. 722–729, 2015.

[26] H. Wang, K. Kim, and Q. Zou, “B-spline-decomposition-based output tracking with preview for nonminimum-phase linear systems,” *Automatica*, vol. 49, no. 5, pp. 1295–1303, 2013.

[27] E. Cetin, R. Morling, and L. Kale, “An integrated 256-point complex FFT processor for real-time spectrum analysis and measurement,” in *Instrumentation and Measurement Technology Conference, 1997. IMTC/97. Proceedings. Sensing, Processing, Networking., IEEE*, vol. 1. IEEE, 1997, pp. 96–101.

[28] S. He and M. Torkelson, “Design and implementation of a 1024-point pipeline FFT processor,” in *Custom Integrated Circuits Conference, 1998. Proceedings of the IEEE 1998.* IEEE, 1998, pp. 131–134.

[29] A. Saeed, M. Elbably, G. Abdelfadeel, and M. Eladawy, “Efficient FPGA implementation of FFT/IFFT processor,” *International Journal of Circuits, Systems and Signal Processing*, vol. 3, no. 3, pp. 103–110, 2009.

[30] T. Hiyama, N. Suzuki, and T. Funakoshi, “On-line identification of power system oscillation modes by using real time FFT,” in *Power Engineering Society Winter Meeting, 2000. IEEE*, vol. 2. IEEE, 2000, pp. 1521–1526.

[31] J. Hurchalla, “A time distributed FFT for efficient low latency convolution,” in *Audio Engineering Society Convention 129.* Audio Engineering Society, 2010.

[32] J. Salt and M. Tomizuka, “Hard disk drive control by model based dual-rate controller. computation saving by interlacing,” *Mechatronics*, vol. 24, no. 6, pp. 691–700, 2014.

[33] J. C. Gilbert and J. Nocedal, “Global convergence properties of conjugate gradient methods for optimization,” *SIAM Journal on Optimization*, vol. 2, no. 1, pp. 21–42, 1992.

[34] Y. Yan, H. Wang, and Q. Zou, “A decoupled inversion-based iterative control approach to multi-axis precision positioning:3D nanopositioning example,” *Automatica*, vol. 48, no. 1, pp. 167–176, 2012.

[35] K. S. Kim and Q. Zou, “A modeling-free inversion-based iterative feedforward control for precision output tracking of linear time-invariant systems,” *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 6, pp. 1767–1777, 2013.

[36] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control 3rd Ed.* Wiley.com, 2012.

[37] M. A Lau and L. Y Pao, “Input shaping and time-optimal control of flexible structures,” *Automatica*, vol. 39, no. 5, pp. 893–900, 2003.

[38] S. Devasia, D. Chen, and B. Paden, "Nonlinear inversion-based output tracking," *Automatic Control, IEEE Transactions on*, vol. 41, no. 7, pp. 930–942, 1996.

[39] Q. Zou, "Optimal preview-based stable-inversion for output tracking of nonminimum-phase linear systems," *Automatica*, vol. 45, no. 1, pp. 230–237, 2009.

[40] L. Zaccarian and A. R. Teel, *Modern anti-windup synthesis: control augmentation for actuator saturation.* Princeton University Press, 2011.

[41] J. Descusse and C. Moog, "Dynamic decoupling for right-invertible nonlinear systems," *Systems & Control Letters*, vol. 8, no. 4, pp. 345–349, 1987.

[42] L. M. Silverman, "Inversion of multivariable linear systems," *Automatic Control, IEEE Transactions on*, vol. 14, no. 3, pp. 270–276, 1969.

[43] S. Devasia, "Nonlinear minimum-time control with pre-and post-actuation," *Automatica*, vol. 47, no. 7, pp. 1379–1387, 2011.

[44] Q. Zou and S. Devasia, "Preview-based stable-inversion for output tracking of linear systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 121, no. 4, pp. 625–630, 1999.

[45] E. Zattoni, "Geometric methods for invariant-zero cancellation in linear multivariable systems with application to signal rejection with preview," *Asian Journal of Control*, 2014.

[46] K. F. Riley, M. P. Hobson, and S. J. Bence, *Mathematical methods for physics and engineering.* Cambridge University Press, 2006.

[47] S. Wright and J. Nocedal, *Numerical optimization.* Springer New York, 1999, vol. 2.

[48] S. Sastry, *Introductory methods of numerical analysis 4th Ed.* PHI Learning Pvt. Ltd., 2005.

[49] I. Zelinka, V. Snasel, and A. Abraham, *Handbook of optimization: from classical to modern approach.* Springer, 2012, vol. 38.

[50] T. Kailath, *Linear systems.* Prentice-Hall Englewood Cliffs, NJ, 1980, vol. 1.

[51] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[52] J. F. Carinena, J. Grabowski, and G. Marmo, "Superposition rules, Lie theorem, and partial differential equations," *Reports on Mathematical Physics*, vol. 60, no. 2, pp. 237–258, 2007.

[53] E. Witten, "Dynamical breaking of supersymmetry," *Nuclear Physics B*, vol. 188, no. 3, pp. 513–554, 1981.

[54] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4, pp. 667–682, 1999.

[55] X. Gao and S. Mishra, "An iterative learning control algorithm for portability between trajectories," in *American Control Conference (ACC), 2014*. IEEE, 2014, pp. 3808–3813.

[56] L. Menini and A. Tornambè, "Nonlinear superposition formulas for two classes of non-holonomic systems," *Journal of Dynamical and Control Systems*, vol. 20, no. 3, pp. 365–382, 2014.

[57] L. Menini and A. Tornambe, "Nonlinear superposition formulas: Some physically motivated examples." in *CDC-ECE*, 2011, pp. 1092–1097.

[58] P. Ge and M. Jouaneh, "Tracking control of a piezoceramic actuator," *Control Systems Technology, IEEE Transactions on*, vol. 4, no. 3, pp. 209–216, 1996.

[59] I. D. Mayergoyz, *Mathematical models of hysteresis and their applications.* Academic Press, 2003.

[60] Y. Xu and G. Basset, "Sequential virtual motion camouflage method for nonlinear constrained optimal trajectory control," *Automatica*, vol. 48, no. 7, pp. 1273–1285, 2012.

[61] J. Liu and Q. Zou, "On single-basis online asymptotic trajectory decomposition for control applications," in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2016, pp. 1291–1296.

[62] H. Akima, "A new method of interpolation and smooth curve fitting based on local procedures," *Journal of the ACM (JACM)*, vol. 17, no. 4, pp. 589–602, 1970.

[63] B. W. Silverman, "Some aspects of the spline smoothing approach to non-parametric regression curve fitting," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–52, 1985.

[64] Q. Chen, N. Huang, S. Riemenschneider, and Y. Xu, "A B-spline approach for empirical mode decompositions," *Advances in Computational Mathematics*, vol. 24, no. 1-4, pp. 171–195, 2006.

[65] R. Tóth, P. S. Heuberger, and P. M. Van den Hof, "Asymptotically optimal orthonormal basis functions for LPV system identification," *Automatica*, vol. 45, no. 6, pp. 1359–1370, 2009.

[66] L. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *Neural Networks, IEEE Transactions on*, vol. 3, no. 5, pp. 807–814, 1992.

[67] J. Liu and Q. Zou, "A decomposition-based learning approach to hysteresis-dynamics system control: Piezoelectric actuator example," in *American Control Conference (ACC), 2015*. IEEE, 2015, pp. 2649–2654.

[68] D. Pavković, S. Polak, and D. Zorc, "PID controller auto-tuning based on process step response and damping optimum criterion," *ISA Transactions*, vol. 53, no. 1, pp. 85–96, 2014.

[69] W. Rudin, *Real and complex analysis.* Tata McGraw-Hill Education, 1987.

[70] S. Demko, W. F. Moss, and P. W. Smith, “Decay rates for inverses of band matrices,” *Mathematics of Computation*, vol. 43, no. 168, pp. 491–499, 1984.

[71] J. K. Hunter and B. Nachtergaele, *Applied analysis.* World Scientific, 2001.

[72] D. A. Bristow, “Frequency domain analysis and design of iterative learning control for systems with stochastic disturbances,” in *American Control Conference, 2008.* IEEE, 2008, pp. 3901–3907.

[73] J. R. Choi, S. B. Park, D. S. Han, and S. H. Park, “A 2048 complex point FFT architecture for digital audio broadcasting system,” in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, vol. 5. IEEE, 2000, pp. 693–696.

[74] R. Cutler and L. S. Davis, “Robust real-time periodic motion detection, analysis, and applications,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 781–796, 2000.

[75] I. S. Uzun, A. Amira, and A. Bouridane, “FPGA implementations of fast Fourier transforms for real-time signal and image processing,” in *Vision, Image and Signal Processing, IEE Proceedings*, vol. 152, no. 3. IET, 2005, pp. 283–296.

[76] B. Boashash and P. J. Black, “An efficient real-time implementation of the Wigner-Ville distribution,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 35, no. 11, pp. 1611–1618, 1987.

[77] J. Ren, Q. Zou, B. Li, and Z. Lin, “High-speed atomic force microscope imaging: Adaptive multiloop mode,” *Physical Review E*, vol. 90, no. 1, p. 012405, 2014.

[78] Y. Wu and Q. Zou, “An iterative-based feedforward-feedback control approach to high-speed atomic force microscope imaging,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 131, no. 6, p. 061105, 2009.

[79] D. Bristow, M. Tharayil, A. G. Alleyne *et al.*, “A survey of iterative learning control,” *Control Systems, IEEE*, vol. 26, no. 3, pp. 96–114, 2006.

[80] Z. Wang and Q. Zou, “A modeling-free differential-inversion-based iterative control approach to simultaneous hysteresis-dynamics compensation: High-speed large-range motion tracking example,” in *2015 American Control Conference (ACC).* IEEE, 2015, pp. 3558–3563.

[81] K. T. Teng and T. C. Tsao, “A comparison of inversion based iterative learning control algorithms,” in *American Control Conference (ACC), 2015.* IEEE, 2015, pp. 3564–3569.

[82] M. Fallahpour and D. Megías, “Robust high-capacity audio watermarking based on FFT amplitude modification,” *IEICE Transactions on Information and Systems*, vol. 93, no. 1, pp. 87–93, 2010.

[83] M. K. I. Molla and K. Hirose, “Single-mixture audio source separation by subspace decomposition of Hilbert spectrum,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 3, pp. 893–900, 2007.

[84] A. Antoniou, *Digital signal processing.* McGraw-Hill Toronto, Canada, 2006.

[85] A. V. Oppenheim, R. W. Schafer, J. R. Buck *et al.*, *Discrete-time signal processing.* Prentice-hall Englewood Cliffs, 1989, vol. 2.

[86] N. Dahnoun, *Digital signal processing implementation using the TMS320C6000 DSP platform.* Addison-Wesley Longman Publishing Co., Inc., 2000.