Multi-Objective Iterative Learning Control: An Advanced ILC Approach for Application Diversity

by

Ingyu Lim

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2016

Doctoral Committee:

Assistant Professor Kira L. Barton, Chair
Assistant Research Scientist Tulga Ersal
Assistant Professor Dimitra Panagou
Professor Dawn Tilbury

To my wife, Jaewon

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Prof. Kira Barton, for providing me the opportunity to learn and work in a wonderful research environment during my Ph.D. Her encouragement, guidance and support enabled me to develop an understanding of the research and gave me strength to pursue my studies during these years. I am honored to have been her student and deeply grateful for all I have learned from her. I would also like to convey my warm and sincere thanks to my committee members, Prof. Tilbury, Dr. Tulga, and Prof. Panagou, for their valuable support and advice. I truly appreciate their time and effort they volunteered to guide this dissertation work. I believe that their insightful comments have led this dissertation to be more thorough and complete. I also wish to express my love and gratitude to all Barton Research Group members and all my friends at University of Michigan for giving me friendly environment during the course of my studies. I would like to recognize support from the Rackham Graduate School, the University of Michigan College of Engineering and Department of Mechanical Engineering.

I am thankful of my family for their unconditional and perennial support. Lastly, I would like to thank Jaewon for being who she is, the most encouraging, considerate, and loving person I know. Without her love and support, I would not have completed this Ph.D. journey.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Multi-Objective Iterative Learning Control:
An Advanced ILC Approach for Application Diversity

by

Ingyu Lim

Chair: Assistant Professor Kira Barton

While ILC has been applied to repetitive applications in manufacturing, chemical processing, and robotics, several key assumptions limit the extension of ILC to various applications. Conventional ILC focuses on improving the performance of a single metric, such as tracking performance through iterative updates of the time domain control input. The application range is limited to systems that satisfy the assumption of iteration invariance of the plant, reference signal, initial conditions, and disturbances.

We aim to relax this assumption to gain significant advantages. More specifically we focus on relaxing the strict reference tracking requirement to address multiple performance metrics and define the stability bounds across temporal and spatial domains. The aim of this research is expanding the application space of ILC towards non-traditional applications.

Chapter III presents an initial framework to provide the foundation for the multi-objective ILC. This framework is validated by simulation and experimental tests with

a wheeled mobile robot.

Chapter IV extends the initial framework from the temporal domain to the spatial domain. The initial framework is generalized to address four classifications of performance objectives. Stability and performance analysis for each classification is provided. Simulation results on a high-resolution additive manufacturing system validate the extended framework.

For the generalized framework, we present a distributed approach in which additional objectives are considered separately. Chapter V evaluates the difference between this distributed approach, and a centralized approach in which the objectives are combined into a single matrix depending on the classification.

Chapter VI extends the multi-objective ILC to incorporate a region-based tracking problem in which reference uncertainty is addressed through the development of a bounded region. A multi-objective region-to-region ILC is developed and validated by a simulation of a surveillance problem with an UAV and multiple unattended ground sensors. Comparisons with point-to-point ILC, region-to-region ILC, and multi-objective region-based ILC demonstrate the performance flexibility that can be achieved when leveraging the regions.

This dissertation provides new approaches for relaxing the classical assumption of iteration invariant reference tracking. New stability and convergence analysis is provided, resulting in a design methodology for multi-objective ILC. These approaches are validated by simulation and experimental results.

CHAPTER I

Introduction

The concept of learning control can initially be attributed to human learning. Humans exhibit an innate ability to apply learning towards the achievement of a goal. For instance, a basketball player shooting a ball at a fixed position can improve his ability to score by practicing the shot repeatedly (Fig. 1.1). During each shot, the basketball player observes the trajectory of the ball and consciously plans an alteration in the shooting motion for the next attempt. As the player continues to practice, the correct motion is learned so that the player can repeatedly shoot the ball with an optimal path, thus iteratively improving the shooting accuracy. Similarly, control engineers can apply the concept of learning, or more broadly adaptation,

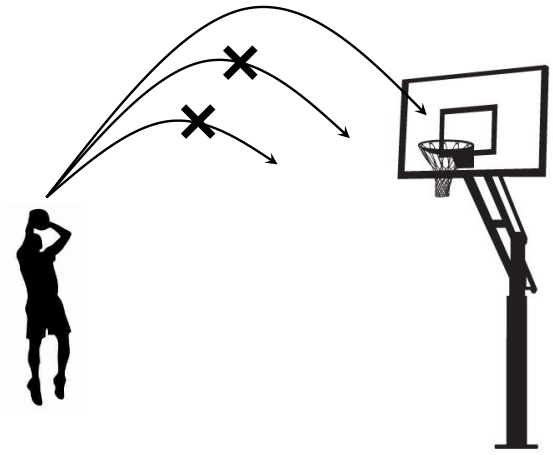


Figure 1.1: Basketball player can improve shooting accuracy by repetitive practice.

towards the improvement of a repetitive system.

There exist many different learning control approaches, such as adaptive control, repetitive control, and iterative learning control (ILC) [1]. Adaptive control approaches modify the controllers themselves, whereas ILC modifies the control input signals [2, 3]. ILC is similar to repetitive control, although the initial conditions are different [4, 5]. In ILC, the initial conditions are assumed to be iteration invariant. The system returns to its initial position after finishing its task, then repeats the task again. However, in repetitive control, the initial conditions are determined by the final conditions of the previous trial. The new trial starts from the position where the last trial finishes.

ILC can improve the performance of systems which repeat a given task multiple times. ILC has been successfully applied to repetitive applications in manufacturing [6, 7, 8], chemical processing [9, 10], and robotics [11, 12, 13] because the systems or applications of these traditional fields can be well defined under the assumptions of iteration invariance. The key assumptions of iteration invariance include,

(1) The stable plant does not change from iteration to iteration.

(2) Initial conditions are identical for each trial.

(3) The finite reference signal repeats for every iteration.

(4) Disturbances are repetitive and small.

Iteration invariance enables the concept of learning control to be applied towards the improvement of specified metrics. However, based on these assumptions, ILC is applied to a limited class of systems and application areas. The goal of this research is to investigate the feasibility of relaxing the requirement of iteration invariance to increase application diversity. In particular, we focus on relaxing the fundamental assumption related to the reference signal in order to extend ILC into new application domains.

Returning to the basketball player example; to score points, the only criteria is

that the basketball must fall through the hoop. The trajectory of the ball does not directly affect the performance; that is, the solution is not singular as there are many different arcs that will satisfy the performance requirement. This freedom enables the player to make a shot that may optimize alternative considerations such as shooting higher to avoid a blocked shot.

Mapping the basketball player example to a more standard control problem, there exist systems or applications that focus on improving the performance only at select points of interest that comprise a subset of the reference signal. For instance, pick-and-place robots primarily focus on the pick-up and drop-off locations [14]. By relaxing the strict reference tracking constraint, additional feed-forward control flexibility can be obtained during the interim times. We leverage this control flexibility to address additional performance objectives.

In addition to discrete, selected points of interest, we also consider the case where the locations of these points may vary within a bounded uncertainty region. For example, consider surveillance missions in which unmanned autonomous vehicles (UAVs) are tasked with obtaining data from unattended ground sensors (UGSs) [15, 16, 17, 18]. In order to meet the mission requirements, the UAV must gather data from all of the UGSs. While the standard practice relies on known drop locations of the UGSs, environmental and external disturbances may shift the drop location. To address this uncertainty in the reference locations, we develop a region-based learning framework that guarantees data transfer, and enables the controller to address additional performance metrics such as overall distance traveled. Distance is a particularly important metric for surveillance where fuel savings from shorter distances may lead to fewer refueling tasks, and the ability to respond more quickly in the presence of information that requires decisive actions. To address this scenario, we derive a multi-objective learning framework that modifies the selected points to regions of interest that guarantee data transfer in the presence of the uncertainty in

the reference.

The validity of these modified learning control frameworks towards the implementation of ILC on alternative application spaces is validated through a combination of simulation and experimental tests. An extension of the temporal ILC framework to a spatial framework provides an appropriate domain for considering spatially dependent applications such as additive manufacturing. Applications ranging from autonomous vehicles to additive manufacturing will be considered.

The remainder of this chapter provides a brief introduction to the core iterative learning control frameworks utilized in this research.

1.1 Conventional ILC

ILC is an adaptive feedforward control approach for improving the performance of systems which repeat the same task over a finite interval. Figure 1.2 shows a standard progression in the error and control signal over several iterations with the use of ILC. Before the start of each iteration, ILC algorithms use the error signal and control signal from previous iterations to generate an updated control signal for the current iteration to improve system performance [1, 8, 19]. Over several iterations, the feedforward control signal is optimized and the error signal is minimized.

ILC has been successfully applied to repetitive applications in manufacturing [6, 7, 8], chemical processing [9, 10], and robotics [11, 12, 13]. In majority of these applications, ILC was implemented to improve the trajectory tracking performance of the system in the time domain through iterative updates to the control signal.

For this approach, a standard time-domain update law is defined by [22]

$$\mathbf{u}_{j+1} = \mathbf{L}_u \mathbf{u}_j + \mathbf{L}_e \mathbf{e}_j. \tag{1.1}$$

In Eq. (1.1), $\mathbf{u}$ represents a vector containing all time domain input signals across

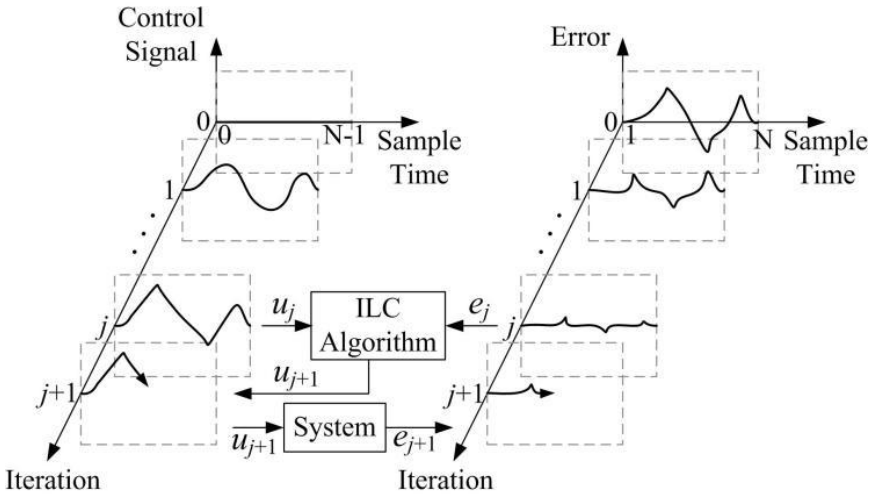


Figure 1.2: A standard progression in the error and control signal over several iterations with the use of ILC [1]

Figure 1.3: Standard applications in manufacturing and robotics in which ILC has shown significant performance improvements [1, 13, 20, 21].

the entire cycle, and $\mathbf{L}_{u,e}$ represent learning filters in the form of matrices containing the time domain parameters. This structure is known as the lifted domain [23]. Eq. (1.1) is derived by minimizing the following cost function with respect to $\mathbf{u}_{j+1}$ [22].

$$J = \mathbf{e}_{j+1}^T \mathbf{Q} \mathbf{e}_{j+1} + \mathbf{u}_{j+1}^T \mathbf{S} \mathbf{u}_{j+1} + (\mathbf{u}_{j+1} - \mathbf{u}_j)^T \mathbf{R} (\mathbf{u}_{j+1} - \mathbf{u}_j).$$

The control objective of conventional ILC requires precise tracking of the entire trajectory. By relaxing this strict tracking constraint, we can obtain additional control flexibility to address additional performance metrics. In order to relax the constraint, point-to-point ILC is first introduced before segueing into multi-objective ILC.

1.2 Point-to-point ILC

As an alternative to using the complete error signal, a point-based controller focuses on improving the error at discrete locations or times for performance enhancements in applications such as, robotic pick n' place tasks [14], patient stroke rehabilitation [24], and reconnaissance missions with UAVs [25]. In these application examples, specific locations (e.g. the start and end positions for pick n' place robots) are critical to the success of the task, while the motion profile between the locations is irrelevant. Recent work by Freeman and Rogers [26] has resulted in an ILC algorithm termed point-to-point learning control that focuses on specific times or locations of a predetermined motion profile. In point-to-point ILC, the selected points define a subset of the motion profile, $\chi(n_i) \subseteq y_d(k)$, where n_i are the selected points for all $i = 1, \ldots, M$, $y_d(k)$ defines the motion profile, and k is the time index. The learning controller only applies a feedforward update to these specified points, $\chi(n_i)$. By removing the unnecessary constraint of a predefined path between the points, additional feed-forward control freedom can be obtained and redirected towards achieving multiple performance objectives, Figure 1.4.

For this approach, the update law defined in (1.1) can also be applied, although the definitions of $\mathbf{L}_u$ and $\mathbf{L}_e$ have been modified to address the subset of points identified in $\chi(n_i)$. The new cost function can be written as,

$$J = (\mathbf{\Psi e}_{j+1})^T \mathbf{Q} (\mathbf{\Psi e}_{j+1}) + \mathbf{u}_{j+1}^T \mathbf{S} \mathbf{u}_{j+1} + (\mathbf{u}_{j+1} - \mathbf{u}_j)^T \mathbf{R} (\mathbf{u}_{j+1} - \mathbf{u}_j),$$

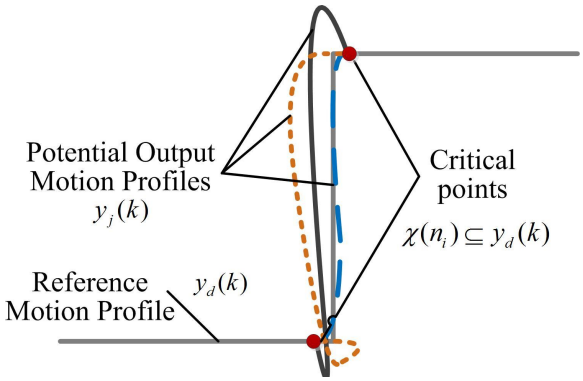


Figure 1.4: Illustration of the point-to-point tracking problem.

where $\mathbf{\Psi}$ is a matrix identifying the indices of the selected points $\chi(n_i)$.

$$\mathbf{\Psi} = \begin{bmatrix} \Psi(0,0) & & 0 \\ & \ddots & \\ 0 & & \Psi(N-1,N-1) \end{bmatrix}$$

$$\mathbf{\Psi}(i,j) = \begin{cases} 1, & i = j = n_k, \quad k = 1, 2, \ldots, M \\ 0, & \text{otherwise} \end{cases}$$

The additional flexibility from the algorithm results in multiple potential solutions that can be optimized for additional metrics such as speed, energy usage, attack angle, and robustness.

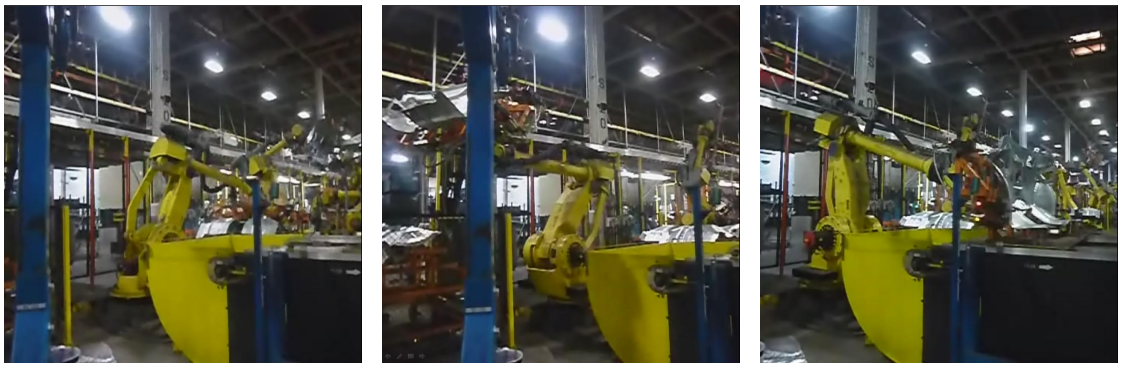
1.3 Multi-objective ILC

Conventional ILC focuses on improving the performance of a single objective such as trajectory tracking. However, there exist applications where the system behavior is governed by two or more objectives. Examples of applications that perform repetitive tasks with multiple performance metrics can be found in manufacturing [28] (objectives: throughput, part quality, material waste/usage) and robotics [13, 29, 30] (objectives: speed, path following, energy consumption, sensor transmission strength).

Focusing on two or more performance objectives provides a greater degree of design and control flexibility that leverages underutilized control actuation that can result from meeting the performance requirements of different objectives [13].

For example, consider the manufacturing process illustrated in Fig. 1.5. In this process, an adhesive is applied at discrete locations on the part. These locations, along with the initial and final positions are the selected points of interest. The pathway between each of these locations can be designed such that another performance metrics is addressed such as speed, vibration damping, avoidance of another machine, etc.

Pareto optimization is a commonly employed multi-objective approach in which two or more conflicting objectives are weighted within a single framework [31]. Solutions to this class of problems require a trade-off in the performance objectives based on desired design criteria. Trade-offs within a control design are frequently made as a trade-off between performance and robustness [32, 33], or as a single performance objective optimized within a constrained system [34]. Figure 1.6 presents a simplified example of the performance trade-offs that may occur within a pareto optimization. If we consider transportation, we can define two (amongst many) competing objectives; energy savings (fuel usage) versus time savings. Figure 1.6 presents the pareto



(a) Initial part location. (b) Applying the ashesion to selected locations. (c) Final location of the part.

Figure 1.5: Example manufacturing process that could leverage a multi-objective control design. [27]

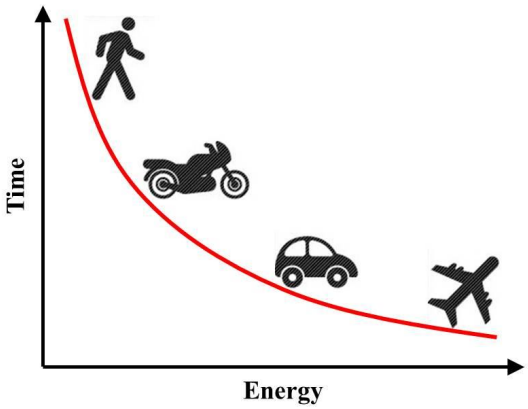


Figure 1.6: Example of Pareto optimization; energy savings vs. time savings.

trade-off between these two objectives. Clearly the decision to walk is the most energy efficient with respect to fuel savings; however, this will require a significant amount of time that may not be reasonable for the user. The extreme choices (extrema on the left and right) are often too polarized in one direction to be realized. However, the options within the middle of the pareto optimization often provide a more reasonable trade-off between the different objectives, such as taking a car or motorcycle in this case.

To enable the learning algorithm to address multiple performance metrics, a more flexible control structure must be defined such that underutilized control can be applied towards multiple performance metrics. To achieve this control flexibility, the primary objective (e.g., typically a tracking requirement) is modified to include a select group of discrete states based on the point-to-point framework described in the previous subsection, $\chi(n_p) \subseteq g_d(k)$, where n_p are the selected points for all $p = 1, ..., \bar{M}$, $g_d(k)$ defines the desired performance profile, and k is the time index. Once the primary objective has been defined, additional performance metrics can be identified. These metrics fall into three main categories: complementary metrics that contribute to the overall weighting on the primary objective, and competing metrics that result in a trade-off scenario between the different objectives as a function of either the

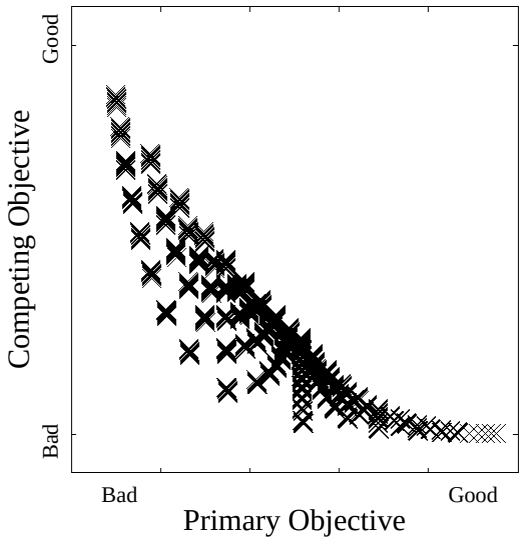


Figure 1.7: Example trade-off between two key performance objectives: primary versus competing.

input or the output, Fig. 1.7.

Recently published work in the area of multi-objective learning [35, 36, 37] focuses on the use of a two-step approach to optimizing system performance. Step 1 aims at optimizing the control effort to achieve zero steady-state trajectory tracking (the primary objective in these applications). In step 2, the framework seeks to optimize the performance of an additional objective through the use of a cost function that considers the additional objective while simultaneously minimizing the difference between a new control input and the optimal control signal determined in step 1. This iterative learning sequence involves multiple steps, while bounding the range of the new solution to be arbitrarily close to the initial optimal input.

Our preliminary work [13, 38] introduces a multi-objective learning control framework that optimizes multiple performance objectives simultaneously. As a result of our one-step optimization approach, the relationship between the primary objective and additional objectives can be clearly observed. Additionally, by eliminating the constraint on the control signal that is imposed in the two step approach, the optimization search is implemented over a broader set of potential solutions. This enables

a greater variety of possible outcomes. In Chapter III, we extend the research from [13, 38] in several key areas:

1 Chapter III provides a baseline framework for multi-objective ILC with stability and convergence analysis.

2 A comparison study of the effects of weighting alternative performance metrics is presented in Chapter III.

3 Simulation and experimental results validate the proposed approach.

For a multi-objective learning approach, the update law is defined by

$$\mathbf{u}_{j+1} = \mathbf{L}_u \mathbf{u}_j + \mathbf{L}_e \mathbf{e}_j + \sum_i \mathbf{L}_i \boldsymbol{\varepsilon}_{i,j},$$

where $\boldsymbol{\varepsilon}_{i,j}$ is the i_{th} additional performance metric for the j_{th} trial. With this novel algorithm, ILC can be extended to address multiple performance objectives.

1.4 Spatial ILC

The conventional ILC framework was constructed in the temporal domain. However, certain applications such as additive or subtractive manufacturing can exhibit system behavior defined spatially rather than temporally. Additive manufacturing has been used to develop functional parts for aerospace applications, the biomedical industry, and flexible electronics [39, 40, 41], Fig. 1.8. A unique challenge in additive manufacturing is the correlation between device functionality and the spatial fidelity of material deposition. As such, with the use of additive manufacturing steadily increasing, the need for a spatial learning framework has increased.

Figure 1.9 introduces points A and B that are distant in time along a raster trajectory, but close in space. In the temporal domain, the system behavior and subsequently the learning algorithm is defined as a 1D function in time (t). A response at one point in time only affects the outputs of those instances that are temporally

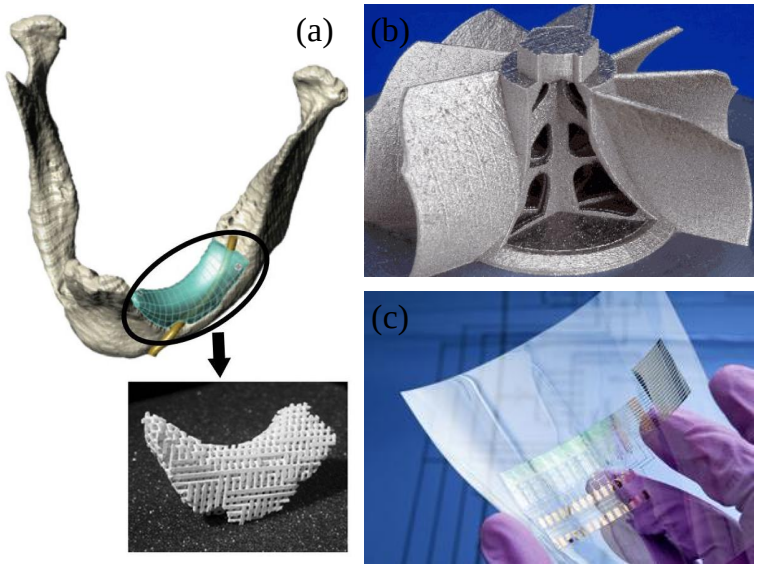


Figure 1.8: Functional Parts for a) biomedical industry [39], b) aerospace application [40], and c) flexible electronics [41].

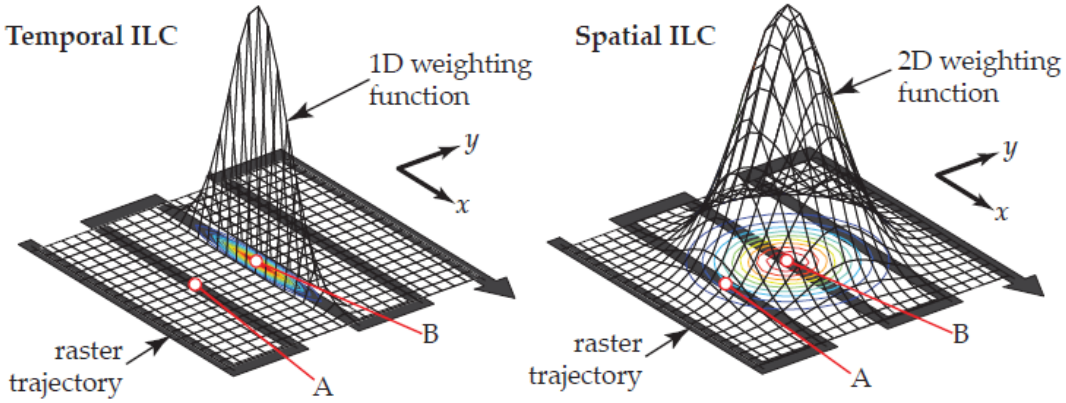


Figure 1.9: Comparison of a temporal ILC weighting function and the proposed spatially dependent scheme [42]

close, not necessarily spatially close. In the temporal domain, the response at B is not considered as an impact on the output of A. In the spatial domain, system behavior is defined as a 2D function in the x-y plane. Behavior at one location in space (x_1, y_1) can affect the outputs in the surrounding 2D region. In this framework, the spatial response at B can be shown to have an impact on the output of A. For these systems, a 1D temporal based learning algorithm does not contain the necessary information

to ensure sufficient performance improvement within the 2D space. In order to ensure the interactions between two points, one must use a 2-D spatial approach rather than 1-D temporal approach.

Examples of spatial ILC algorithms can be found in literature [42, 43, 44, 45, 46, 47, 48]. Many of these approaches utilize known mappings between time and space, indirectly incorporating a time component into the framework. Recently, [42] introduced a spatial learning framework that eliminates the notion of time (but not iteration) from the algorithm development. In this research, we extend the spatial framework to formulate a multi-objective learning framework in the spatial domain. A multi-objective spatial learning framework is analogous to the multi-objective temporal learning framework in [13]. The key challenge stems from the definition and subsequent derivation of the performance objectives in the lifted framework; a structure requiring the use of convolution with 2D spatial signals. Simulation results based on a micro-additive manufacturing process demonstrate the use of the multi-objective spatial ILC framework for optimized device design and fabrication.

In this dissertation, we extend the research from [13, 38, 49] in several key areas:

1 Chapter IV provides a complete description of Spatial Multi-Objective ILC with full stability, convergence, and robustness analysis and a simulation study from an empirically derived model, all of which were not provided in [13, 38].

2 Chapter IV details three classifications of additional performance metrics: complementary, competitive, and domain specific with robust performance and stability analysis provided for all three classification categories.

3 Chapter IV provides an updated design methodology aimed at simplifying the design of a multi-objective iterative learning controller.

4 A high-fidelity additive manufacturing simulation example has been provided for controller validation.

5 Chapter V defines a single weighting matrix that combines primary and complementary performance metrics into a single matrix. It then compares the performance flexibility of the single versus individual weighting gain designs.

For the spatial approach, the update law is defined as,

$$\mathbf{u}_{j+1} = \mathbf{L}_u \mathbf{u}_j + \mathbf{L}_e \mathbf{e}_j.$$

The equations look analogous to conventional ILC in the temporal domain; however, the system dynamics have been mapped by a 2D spatial impulse rather than a 1D temporal impulse.

For a spatial framework with multiple performance objectives, the update law is defined as,

$$\mathbf{u}_{j+1} = \mathbf{L}_u \mathbf{u}_j + \mathbf{L}_e \mathbf{e}_j + \sum_i \mathbf{L}_i \boldsymbol{\varepsilon}_{i,j},$$

where $\boldsymbol{\varepsilon}_{i,j}$ is the i_{th} additional performance metric for the j_{th} trial. With this architecture, ILC is extended to applications dominated by spatial dynamics rather than time dynamics.

1.5 ILC with Reference Uncertainty

Conventionally, ILC has been applied to systems with the same reference signal over numerous iterations. However, there may exist tasks in which the reference exhibits uncertainty from pass-to-pass. Consider the example of a surveillance mission in which a UAV is tasked with collecting data from a series of UGSs. If the locations of the UGSs contain uncertainty, this can directly impact the performance capability of the system [15, 16, 17, 18].

To address this challenge, we estimate bounded regions in which the system can ensure data transfer from the UGS to the UAV. The bounded regions are a function of the bounded uncertainty in the reference locations. Once these regions have

been identified, we introduce a region-to-region iterative learning control framework that aims to maximize the tracking performance within the regions. Similar to the framework presented in Chapters III and IV, a multi-objective framework will be considered.

For the region-to-region learning framework, the update law can be written as,

$$\mathbf{u}_{j+1} = \mathbf{L}_u \mathbf{u}_j + \mathbf{L}_e \hat{\mathbf{e}}_j,$$

where $\hat{\mathbf{e}}_j$ defines the error in reaching the bounded region. Stability and convergence analysis will show the parallels of this approach with the conventional point-to-point and multi-objective ILC frameworks presented in Chapters III and IV.

CHAPTER II

Problem Statement

Despite the successful application of ILC to a range of systems in manufacturing, robotics, and chemical processing [1], the key assumption of iteration invariance described in Chapter 1 has limited the extension of ILC to a more diverse class of systems.

The goal of this dissertation is to develop various methods for relaxing the fundamental assumption of iteration invariant reference signals with full reference signal requirements. The methods are designed to leverage advantages that result from allowing reference variations from pass to pass, while focusing on a subset of key reference points. Modifications to the conventional norm optimal ILC framework are provided for both the temporal and spatial domains in order to apply ILC to new application spaces in both autonomous vehicles and additive manufacturing.

To achieve this goal, the following scientific questions are addressed:

(1) Can the requirement for strict reference tracking constraint be relaxed in order to leverage the newly available control bandwidth towards the improvement of multiple performance metrics?

(2) What knowledge from temporal ILC can be applied towards the development of spatial ILC? What are the differences that drive new design methodologies or stability analysis?

(3) Can strict performance be guaranteed for iteration varying references with a certain level of uncertainty?

2.1 Multi-objective Iterative Learning Control in Temporal Domain

In this task, we develop a multi-objective learning control framework that leverages additional (unused) control bandwidth to address multiple performance metrics. A rigorous yet simplified stability analysis is provided along with an investigation of trade-off considerations for the competing objectives.

Scientific contributions in this work stem from the following:

(1) The development of a new learning framework for multiple performance metrics.
(2) Identifying the structure of the additional performance metrics that must be applied in order for the new metric to be integrated into the cost function and update law.
(3) Providing a rigorous stability analysis of the modified framework.
(4) Providing a systematic design methodology for the multi-objective controller.

The application for this task is a UAV tracking system with multiple performance metrics being identified as (1) set point tracking, and (2) energy usage. Simulation and experiment validations are provided.

Publications related to this research include:

I. Lim and K. L. Barton, "Pareto optimization-based iterative learning control," in *American Control Conference (ACC), 2013*, pp. 5171–5176, IEEE, 2013

I. Lim and K. L. Barton, "Pareto iterative learning control: Optimized control for multiple performance objectives," *Control Engineering Practice*, vol. 26, pp. 125–135, 2014

2.2 Multi-objective Iterative Learning Control in the Spatial Domain

In this task, we extend the multi-objective learning control framework to the spatial domain. Conventionally, ILC has been constructed in the temporal domain to primarily address trajectory tracking problems. However, the new focus on additive manufacturing necessitates a controller that addressed performance metrics defined in space rather than time.

We will show that a spatial learning framework is analogous to the temporal learning framework, with the main differences stemming from (1) a 2D impulse response format, and (2) the construction of circulant rather than Toeplitz lifted matrices. Stability and convergence analyses are provided.

Scientific contributions from this research stems from the following:

(1) Extension of the multi-objective learning framework into the spatial domain.

(2) Defining the additional performance metrics as a function of input signal with the use of 2D spatial dynamics.

(3) Rigorous stability and convergence analysis of the modified framework.

The target application for this task is an ink deposition system with multiple performance metrics defined as (1) image/part quality, and (2) material usage. A simulation validation is provided.

Published work related to this research includes:

I. Lim, K. L. Barton, and D. Hoelzle, "Spatial ILC for multi-objective systems," in *Dynamic System and Control Conference (DSCC), 2014*, ASME, 2014

I. Lim, K. L. Barton, and D. Hoelzle, "A multi-objective iterative learning control approach for additive manufacturing applications," *Control Engineering Practice, submitted*, 2015

2.3 Iterative Learning Control with Reference Uncertainty

In this task, we develop a learning framework with reference uncertainty. Conventionally, ILC has been applied to applications or tasks with iteration invariant references. However, there may exist tasks where the reference contains a bounded uncertainty. To address this uncertainty, we develop a region-to-region learning control framework. The regions are estimated using the bounded uncertainty of the reference signal.

Scientific contributions in this work stem from the following:

(1) Estimation of the target regions based on the bounds of the reference uncertainty.

(2) The development of a region-to-region learning framework that addresses the reference uncertainty.

(3) Comparison of the proposed approach to existing learning-based methods to assess the advantages of leveraging the bounded regions.

The application for this task is a UAV tracking system with reference uncertainty. Simulation validations are provided.

CHAPTER III

Multi-Objective Iterative Learning Control in Temporal Domain

In this chapter, we present a generalized multi-objective learning control framework for systems that require the optimization of multiple performance objectives simultaneously. To address the performance requirements for these types of systems, the control objectives are posed as a multi-objective learning problem where the controller seeks to optimize a cost function containing multiple performance objectives.

To demonstrate the validity of this approach, we implement the generalized multi-objective learning control framework on a wheeled mobile robot. We identify the following key performance metrics: (1) tracking of a select subset of locations/points, and (2) energy savings throughout the task as a function of distance travelled. Through the analysis, and simulation and experimental results, the trade-off between trajectory tracking and energy savings is clearly observed. Additionally, the optimization search is implemented over a broad set of potential solutions; thus enabling a greater variety of possible outcomes.

Simulation and experimental results validate the controller performance and provide a means for verifying trends in the system behavior. A performance trade-off analysis is also included to evaluate potential design choices for energy reduction.

3.1 Class of Systems

For clarity of exposition, the class of systems considered for this work include linear, causal, discrete-time single-input, single-output (SISO) systems, given as,

$$\mathbf{H} \triangleq \begin{cases} \mathbf{x}_j(k+1) = \mathbf{A}\mathbf{x}_j(k) + \mathbf{B}u_j(k) \\ y_j(k) = \mathbf{C}\mathbf{x}_j(k) \end{cases} \tag{3.1}$$

where $\mathbf{x}(k) \in \mathbb{R}^p$ are the system states, $u(k) \in \mathbb{R}$ is the control input, $y(k) \in \mathbb{R}$ is the output, $k \in \mathbb{Z}^{N+1}$ is the time index, and $j = 1, 2, \ldots$ is the iteration index. $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ are appropriately sized iteration-invariant real-valued matrices. It is assumed that $\mathbf{x}_j(0) = \mathbf{x}_0$ for all j. As defined by the matrices $(\mathbf{A}, \mathbf{B}, \mathbf{C})$, $\mathbf{H}$ is time-invariant over a single profile and iteration-invariant from trial-to-trial. In the lifted-domain, the discrete-time behavior of the system is represented by its convolution matrix using impulse response data $\mathbf{H}_{m,n}$. The lifted-system representation transforms a two-dimensional (time and iteration) system into an one-dimensional (iteration) system. The lifted system representation is given by,

$$\mathbf{H} = \begin{bmatrix} H_{0,0} & & 0 \\ \vdots & \ddots & \\ H_{N-1,0} & \cdots & H_{N-1,N-1} \end{bmatrix}$$

For LTI systems, $\mathbf{H}_{m,n}$ contains the impulse response data and can be derived using the matrices in (3.1)

$$\mathbf{H}_{m,n} : \left\{ \mathbf{C}\mathbf{A}^{m-n}\mathbf{B}, \quad m \geq n \right.$$

While the results presented in this paper are for an LTI system, the same design process can be applied to LTV systems. In the case of LTV systems, $\mathbf{H}_{m,n}$ is of the

form

$$\mathbf{H}_{m,n}: \begin{cases} \mathbf{C}(n)\mathbf{B}(n), & m = n \\ \mathbf{C}(m)\mathbf{A}(m-1)\mathbf{A}(m-2)\dots\mathbf{A}(n)\mathbf{B}(n), & m > n \end{cases}$$

3.2 Norm Optimal ILC

This work adopts the widely used norm optimal iterative learning control (NOILC) approach [23, 52, 53]. The norm optimal approach was chosen for its monotonic convergence guarantees and design trade-off abilities, such as the intuitive weighting structure and modal architecture that enables weighting of multiple objectives. The general and point-to-point based norm optimal frameworks are briefly described here, and will be extended to enable design modifications for systems with multiple performance objectives.

3.2.1 Conventional ILC

A well-known norm optimal ILC update law is adopted [54].

$$\mathbf{u}_{j+1} = \mathbf{L}_u\mathbf{u}_j + \mathbf{L}_e\mathbf{e}_j$$

where $\mathbf{e}_j = \mathbf{y}_d - \mathbf{y}_j = \mathbf{y}_d - \mathbf{H}\mathbf{u}_j$ with $\mathbf{e}_j = \begin{bmatrix} e_j^T(1) & e_j^T(2) & \cdots & e_j^T(N) \end{bmatrix}^T$ and $\mathbf{u}_j = \begin{bmatrix} u_j^T(0) & u_j^T(1) & \cdots & u_j^T(N-1) \end{bmatrix}^T$.

The norm optimal ILC algorithm is designed to minimize a quadratic optimization problem, in which the objective is to minimize a cost function [55],

$$J = \mathbf{e}_{j+1}^T\mathbf{Q}\mathbf{e}_{j+1} + \mathbf{u}_{j+1}^T\mathbf{S}\mathbf{u}_{j+1} + (\mathbf{u}_{j+1} - \mathbf{u}_j)^T\mathbf{R}(\mathbf{u}_{j+1} - \mathbf{u}_j).$$

$(\mathbf{Q}, \mathbf{S}, \mathbf{R})$ are symmetric positive definite matrices with a common form given as $(\mathbf{Q}, \mathbf{S}, \mathbf{R}) \triangleq (q\mathbf{I}, s\mathbf{I}, r\mathbf{I})$. Minimizing the cost function J with respect to $\mathbf{u}_{j+1}$ yields

the norm optimal ILC update algorithm filters with respect to the weighting matrices $(\mathbf{Q}, \mathbf{S}, \mathbf{R})$ and the plant $\mathbf{H}$ [53].

$$\mathbf{L}_u = \left(\mathbf{H}^T\mathbf{Q}\mathbf{H} + \mathbf{S} + \mathbf{R}\right)^{-1}\left(\mathbf{H}^T\mathbf{Q}\mathbf{H} + \mathbf{R}\right)$$
$$\mathbf{L}_e = \left(\mathbf{H}^T\mathbf{Q}\mathbf{H} + \mathbf{S} + \mathbf{R}\right)^{-1}\mathbf{H}^T\mathbf{Q}$$

Note that for $(\mathbf{L}_u, \mathbf{L}_e)$ to ensure convergence, $\mathbf{H}^T\mathbf{Q}\mathbf{H} + \mathbf{S} + \mathbf{R}$ must be positive definite [8].

3.2.2 Point-to-point ILC

Point-to-point ILC relaxes the trajectory tracking constraint along the entire trajectory and focuses on minimizing the tracking error only at specific points (see Figure 3.1) [26]. Eliminating the full trajectory tracking constraint results in a significant reduction in the feedforward control effort along the cycle, Figure 3.2. This control freedom can be utilized to address additional performance metrics. Figure 3.1 provides a comparison of 3 control schemes. Figure 3.2 presents the feedforward and combined feedforward and feedback control efforts for traditional and point-to-point ILC for the example in Figure 3.1. Note the reduction in feedforward control effort

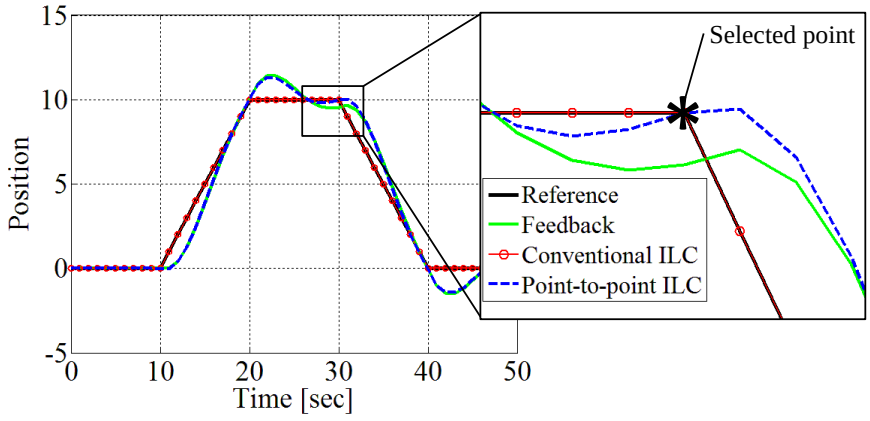


Figure 3.1: Tracking results of feedback control, traditional ILC, and point-to-point ILC with a selected point at 30 seconds.

for the point-to-point ILC approach (Figure 3.2 (a)), resulting in additional design flexibility for this control approach. While the combined feedforward and feedback signals do not exhibit the same reduction in control effort (Figure 3.2 (b)); it is important to note that the feedback controller dominates the signal composition in the point-to-point case. This effort is not being employed towards improved performance; therefore, it can be leveraged for additional performance gains using a targeted feedforward signal designed to address a particular objective.

In point-to-point ILC, the selected points define a subset of the motion profile, $\chi(n_i) \subseteq y_d(k)$, where n_i are the selected points for all $i = 1, \ldots, M$, $M < N$, and N defines the number of samples in a given cycle. Reformulation of the norm optimal framework to incorporate point-to-point tracking yields a modified cost function and learning filters. The modified cost function for point-to-point ILC is given as [26],

$$J=(\mathbf{\Psi e}_{j+1})^T\mathbf{Q}(\mathbf{\Psi e}_{j+1}) + \mathbf{u}_{j+1}^T\mathbf{S}\mathbf{u}_{j+1} + (\mathbf{u}_{j+1} - \mathbf{u}_j)^T\mathbf{R}(\mathbf{u}_{j+1} - \mathbf{u}_j) \tag{3.2}$$

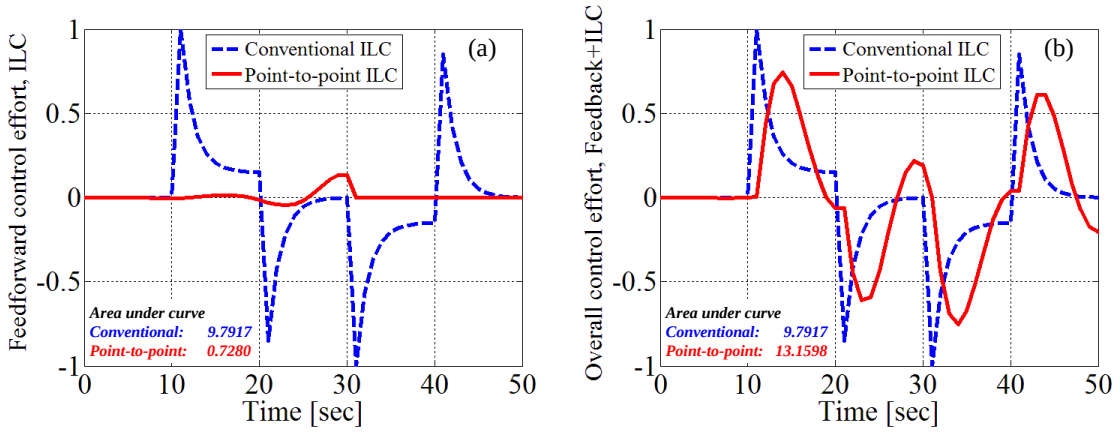


Figure 3.2: Control effort for traditional ILC and point-to-point ILC with a selected point at 30 seconds. a) Feedfoward control effort. b) Feedback and feedforward control effort.

where Ψ is a matrix identifying the indices of the selected points $\chi(n_i)$,

$$\mathbf{\Psi} = \begin{bmatrix} \Psi(0,0) & & 0 \\ & \ddots & \\ 0 & & \Psi(N-1,N-1) \end{bmatrix}$$

$$\mathbf{\Psi}(i,j) = \begin{cases} 1, & i=j=n_k, \quad k=1,2,\ldots,M \\ 0, & otherwise \end{cases}$$

From (3.2), the point-to-point ILC update filters are given as,

$$\mathbf{L}_u = \left((\mathbf{\Psi H})^T \mathbf{Q}(\mathbf{\Psi H}) + \mathbf{S} + \mathbf{R}\right)^{-1} \left((\mathbf{\Psi H})^T \mathbf{Q}(\mathbf{\Psi H}) + \mathbf{R}\right) \tag{3.3}$$

$$\mathbf{L}_e = \left((\mathbf{\Psi H})^T \mathbf{Q}(\mathbf{\Psi H}) + \mathbf{S} + \mathbf{R}\right)^{-1} \left((\mathbf{\Psi H})^T \mathbf{Q}\mathbf{\Psi}\right). \tag{3.4}$$

3.3 Multi-objective ILC

A Pareto optimization-based ILC framework incorporates additional performance metrics into a single controller design. In this manner, the controller is tasked with optimizing multiple, often competing, objectives simultaneously. This leads to a trade-off scenario in which the control objectives are weighted according to desired performance requirements.

3.3.1 Control Framework Assumptions

A Pareto multi-objective learning control framework requires that the following assumptions be satisfied:

(A1) The system is stable or stabilizable.

(A2) Only the points identified by $\mathbf{\Psi}$ will be included in the tracking performance metric.

(A3) Measured output $\mathbf{z}_{i,j}$ is controllable by $\mathbf{u}_j$.

(A1) is necessary for ILC in general. (A2) enables the use of the point-to-point ILC approach, standard update law, and learning filters. (A3) enables $\mathbf{z}_{i,j}$ to be improved through iterative updates to the control input $\mathbf{u}_{j+1}$.

3.3.2 Multi-objective Learning Framework

The modified ILC algorithm and cost function for the Pareto optimization problem are given as,

$$\mathbf{u}_{j+1} = \mathbf{L}_u \mathbf{u}_j + \mathbf{L}_e \mathbf{e}_j + \sum_i \mathbf{L}_i \boldsymbol{\varepsilon}_{i,j} \tag{3.5}$$

$$\begin{aligned} J =& (\boldsymbol{\Psi}\mathbf{e}_{j+1})^T \mathbf{Q} (\boldsymbol{\Psi}\mathbf{e}_{j+1}) + \mathbf{u}_{j+1}^T \mathbf{S} \mathbf{u}_{j+1} + (\mathbf{u}_{j+1} - \mathbf{u}_j)^T \mathbf{R} (\mathbf{u}_{j+1} - \mathbf{u}_j) \\ &+ \sum_i \boldsymbol{\varepsilon}_{i,j+1}^T \mathbf{W}_i \boldsymbol{\varepsilon}_{i,j+1} \end{aligned} \tag{3.6}$$

In (3.5) and (3.6), $\boldsymbol{\varepsilon}_{i,j}$ denotes the error signal of an additional i_{th} control metric, $\mathbf{W}_i$ is the weighting matrix applied to the i_{th} control metric error signal, and $\mathbf{L}_i$ is a new learning filter related to the i_{th} control metric error signal.

The error of the additional control metric is defined as a function of the difference between the desired value and the measured value. Similarly, the error signal is defined as a function of the reference signal and the measured output signal. The error of the additional control metric is written as,

$$\begin{aligned} \boldsymbol{\varepsilon}_{i,j} &= \mathbf{z}_{i,d} - \mathbf{z}_{i,j} \\ &= \mathbf{z}_{i,d} - \mathbf{X}_i \mathbf{u}_j \end{aligned} \tag{3.7}$$

where $\boldsymbol{\varepsilon}_{i,j} \in \mathbb{R}^{N\times 1}$, $\mathbf{z}_{i,d} \in \mathbb{R}^{N\times 1}$ and $\mathbf{z}_{i,j} \in \mathbb{R}^{N\times 1}$ are the desired signal and the

measured output signal for i_{th} objective, $\mathbf{u}_j \in \mathbb{R}^{N\times 1}$ is the input signal, and $\mathbf{X}_i \in \mathbb{R}^{N\times N}$ is a matrix that maps $\mathbf{u}_j$ to the $\mathbf{z}_{i,j}$.

Note that the additional performance metric $\mathbf{z}_{i,j}$ is defined as a function of the input signal $\mathbf{u}_j$. To obtain the update law from the cost function, we differentiate the cost function with respect to $\mathbf{u}_{j+1}$. If $\mathbf{z}_{i,j}$ is not a function of the input signal, the update law will not incorporate a component in the input signal based on this additional metric.

Differentiating the modified cost function with respect to $\mathbf{u}_{j+1}$ yields the learning filters $\mathbf{L}_u$, $\mathbf{L}_e$ and $\mathbf{L}_i$,

$$\begin{aligned}
\mathbf{L}_u &= \mathbf{C}_F \left\{ (\mathbf{\Psi H})^T \mathbf{Q} (\mathbf{\Psi H}) + \mathbf{R} + \sum_i \mathbf{X}_i^T \mathbf{W}_i \mathbf{X}_i \right\} \\
\mathbf{L}_e &= \mathbf{C}_F \left\{ (\mathbf{\Psi H})^T \mathbf{Q} \mathbf{\Psi} \right\} \\
\mathbf{L}_i &= \mathbf{C}_F \left\{ \mathbf{X}_i^T \mathbf{W}_i \right\} \\
\text{where,} \quad \mathbf{C}_F &= \{ (\mathbf{\Psi H})^T \mathbf{Q} (\mathbf{\Psi H}) + \mathbf{S} + \mathbf{R} + \sum_i \mathbf{X}_i^T \mathbf{W}_i \mathbf{X}_i \}^{-1}.
\end{aligned}$$

Importantly, the $\mathbf{L}_i$ filters are not found in traditional or point-to-point ILC as they relate to the additional performance metrics introduced into the multi-objective learning framework. For $(\mathbf{L}_u, \mathbf{L}_e, \mathbf{L}_i)$ to ensure convergence, the following condition must hold true,

$$((\mathbf{\Psi H})^T \mathbf{Q} (\mathbf{\Psi H}) + \mathbf{S} + \mathbf{R} + \sum_i \mathbf{X}_i^T \mathbf{W}_i \mathbf{X}_i) > \mathbf{0}.$$

3.3.3 Asymptotic Stability

Substituting $\mathbf{e}_j = \mathbf{y}_d - \mathbf{H}\mathbf{u}_j$ and (3.7) into (3.5) sets up a recursion with the closed-loop iteration dynamics

$$\mathbf{u}_{j+1} = \left(\mathbf{L}_u - \mathbf{L}_e \mathbf{H} - \sum \mathbf{L}_i \mathbf{X}_i \mathbf{H} \right) \mathbf{u}_j + \mathbf{L}_e \mathbf{y}_d + \sum_i \mathbf{L}_i \mathbf{z}_{i,d}. \tag{3.8}$$

From (3.8), the iterative learning controller is asymptotic stable if and only if

$$\left|\lambda_k\left(\mathbf{L}_u - \mathbf{L}_e\mathbf{H} - \sum \mathbf{L}_i\mathbf{X}_i\mathbf{H}\right)\right| < 1 \tag{3.9}$$

for $k = 1, 2, ..., N$, for where $\lambda_k(\bullet)$ is the k^{th} eigenvalue of $(\bullet)$ [54]. For the recursion in (3.8), satisfying (3.9) guarantees a bounded input signal, $\mathbf{u}_{j+1}$, as per the properties of linear systems.

Theorem III.1. ***Output Boundedness:*** *An output of the form* $\mathbf{y}_{j+1} = \mathbf{H}\mathbf{u}_{j+1}$ *is bounded for all* k *if the ILC control input* $\mathbf{u}_{j+1}$ *is bounded.*

Proof of Theorem III.1 : *By the properties of matrix norm [56],*

$$\|A\|_p = \sup_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p}$$

where $A \in R^{m\times n}$, $x \in R^n$, *and* $p \geq 1$, *the following inequality of the norm is derived as,*

$$\|Ax\|_p \leq \|A\|_p \|x\|_p .$$

$\mathbf{H}$ *is a stable linear system, and let* $\mathbf{u}_{j+1}$ *be bounded by satisfying* (3.9), *then*

$$\| \mathbf{y}_{j+1} \|_2 = \| \mathbf{H}\mathbf{u}_{j+1} \|_2 \leq \| \mathbf{H} \|_2 \| \mathbf{u}_{j+1} \|_2 < \infty$$

Thus, the output is bounded for all k *by the boundedness of* $\mathbf{u}_{j+1}$ *and the properties of linear systems.* □

3.4 Simulation Validation

3.4.1 Simulation setup

To evaluate the performance potential of the proposed approach, this section provides the design and analysis of a multi-objective learning controller for a model

of the wheeled mobile robot used in the experimental testing. The details of the model will be provided in Section 3.5.1. The discrete time closed-loop transfer functions for each axis (assumed to be dynamically decoupled) are given as,

$$H_X(z)=\frac{0.002838z^{-1}-9.966e^{-05}z^{-2}-0.002576z^{-3}}{1-2.782z^{-1}+2.576z^{-2}-0.7935z^{-3}}$$
$$H_Y(z)=\frac{0.002374z^{-1}-0.0001206z^{-2}-0.002227z^{-3}}{1-2.829z^{-1}+2.664z^{-2}-0.8348z^{-3}}$$

with a sampling time of 0.05 seconds. H_X and H_Y are the closed-loop models of the x and y-axis of the wheeled mobile robot used in the experimental testing. As such, the results allow one to verify trends between simulation and experimentation.

The desired output trajectory is given in Figure 3.3. As a demonstrative example for multi-objective ILC, competing performance objectives are selected: 1) the

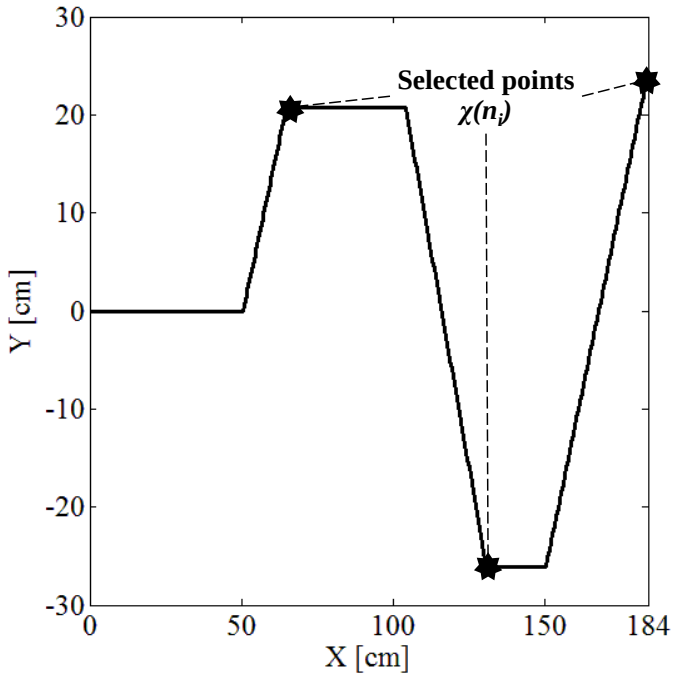


Figure 3.3: Reference trajectory for demonstrative example.

tracking performance of the selected points $\chi(n_i)$ defined as the minimization of,

$$\mathbf{\Psi e}_j = \mathbf{\Psi}(\mathbf{y}_d - \mathbf{y}_j) = \mathbf{\Psi}(\mathbf{y}_d - \mathbf{H u}_j),$$

and 2) distance and acceleration metrics related to the energy usage for a single cycle or iteration;

$$\boldsymbol{\varepsilon}_{l,j} = \mathbf{l}_d - \mathbf{l}_j = \mathbf{l}_d - \mathbf{X}_l \mathbf{u}_j,$$
$$\boldsymbol{\varepsilon}_{a,j} = \mathbf{a}_d - \mathbf{a}_j = \mathbf{a}_d - \mathbf{X}_a \mathbf{u}_j,$$

where $\boldsymbol{\varepsilon}_{l,j}$ and $\boldsymbol{\varepsilon}_{a,j}$ are the error signals for the additional metrics for the distance and the acceleration, respectively. $\mathbf{l}_d$ and $\mathbf{l}_j$ are the desired distance and the measured distance, respectively. $\mathbf{a}_d$ and $\mathbf{a}_j$ are the desired acceleration and the measured acceleration, respectively. For the simulation setup, $\mathbf{l}_d = \mathbf{O}(N,1)$ and $\mathbf{a}_d = \mathbf{O}(N,1)$, where $\mathbf{O}(N,1)$ is a $N \times 1$ column vector with all its entries being 0. To define these metrics, the mapping matrices $\mathbf{X}_l$ and $\mathbf{X}_a$ are of the form,

$$\mathbf{X}_l = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & -1 & 1 \end{pmatrix} \mathbf{H} \quad \text{and} \quad \mathbf{X}_a = \frac{1}{T_s^2} \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & -1 & 1 \end{pmatrix}^2 \mathbf{H}.$$

Commonly, the energy required to run a given system is calculated as the sum of the square of the input control signal at each point in time, $\sum_{k=1}^{N} (u_j(k))^2$, where the control input is typically related to an input current. However, standard and point-to-point norm optimal ILC apply a weighting norm to the control input to reduce the effects of model uncertainty. Increasing this gain has been shown to reduce the effects

of model uncertainty at the expense of significant performance degradation [54]. This trade-off has been demonstrated in Table 3.2.

Comparing the square of the control input and kinetic energy with respect to time in Figures 3.4 and 3.5, one can clearly identify the similarities between the two trends. As such, kinetic energy has been selected as a suitable performance metric for energy use. w_l and w_a denote the additional weighting gains for distance and acceleration, respectively. The gains for the simulation are provided in Table 3.1.

For the purpose of this example, kinetic energy has been defined as,

$$E_j(k) = \frac{1}{2} m \left(v_j(k)\right)^2$$

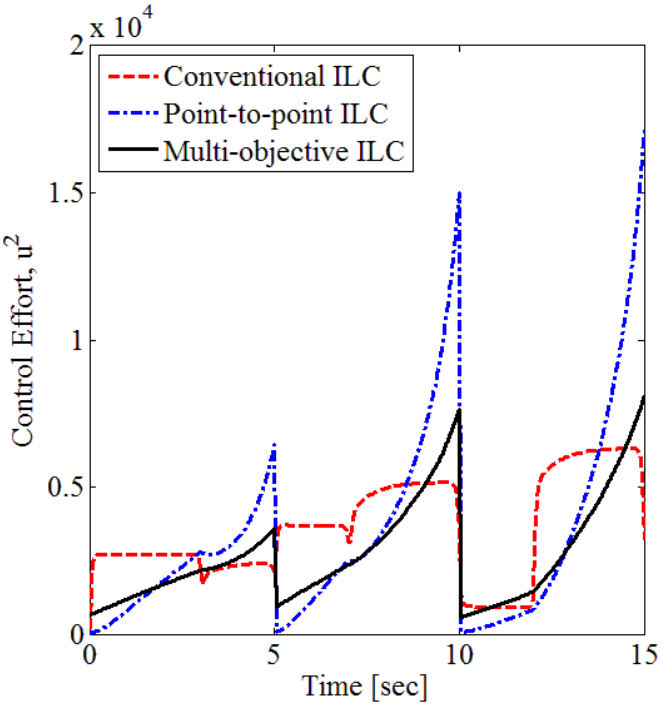
where $E_j(k)$ is the kinetic energy at a given point in time, m defines the mass of the system, and $v_j(k)$ is the translational velocity. Note that the system workspace is assumed to be planar and perpendicular to gravity, thus potential energy has been ignored in the derivation of required energy usage for this system.

When considering kinetic energy, the key parameter is velocity. Importantly, velocity can be defined with respect to acceleration and distance; providing two independent variables that can be used to evaluate energy optimization across a given cycle. The relationships between these variables and velocity are given as:

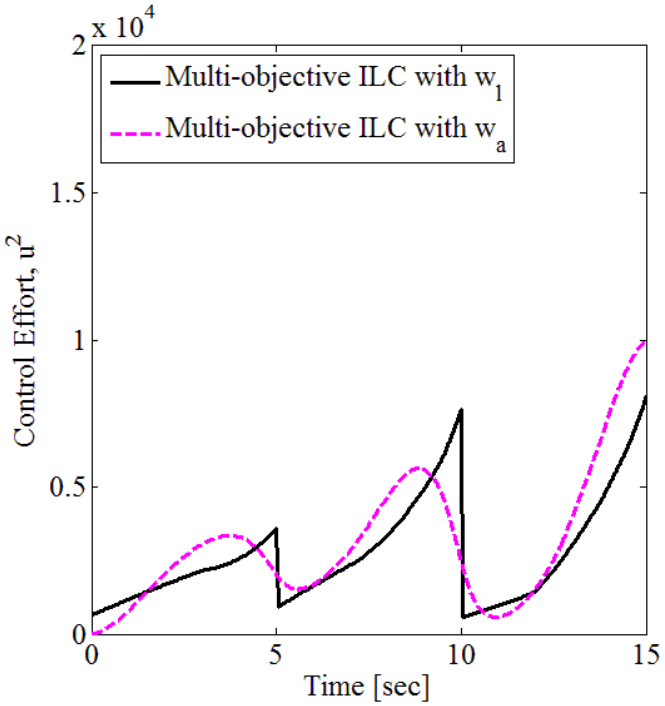
$$l_j(k) = v_j(k) \cdot T_s \tag{3.10}$$

$$v_j(k) = a_j(k) \cdot T_s + v_j(k-1) \tag{3.11}$$

In (3.10) and (3.11), $l_j(k)$ is the distance traveled per time step, $a_j(k)$ is the acceleration per time step, and T_s is the sampling time of the system. Assuming a constant time step and cycle time, changes in distance are directly correlated to changes in velocity and thus kinetic energy. Similarly, changes in acceleration result in a rate of change for the velocity; requiring a modification in the energy usage as compared to

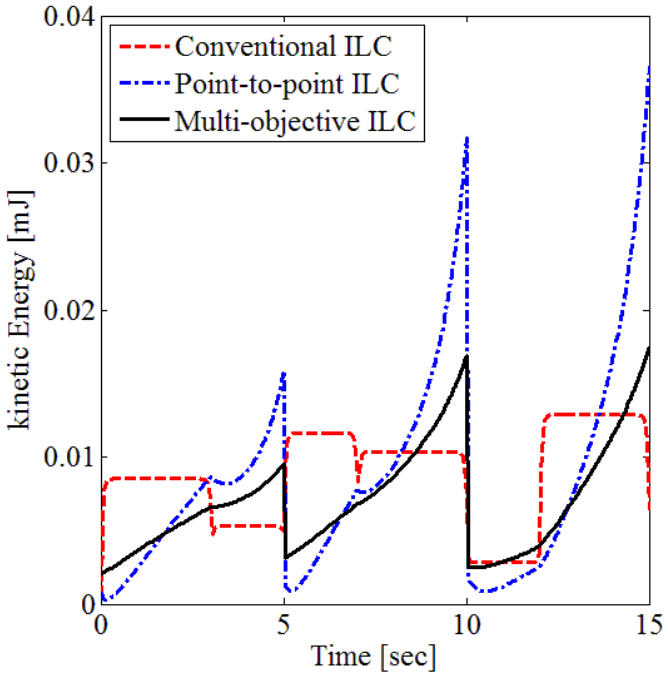


(a) Conventional, point-to-point, and multi-objective ILC.

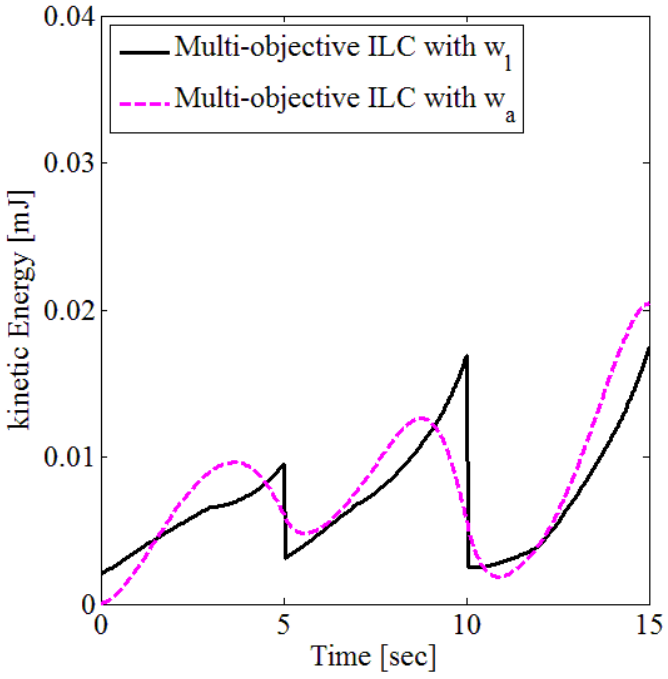


(b) Multi-objective ILC with weighted distance and acceleration, w_l and w_a.

Figure 3.4: A comparison of control effort squared for various ILC designs.



(a) Conventional, point-to-point, and multi-objective ILC.



(b) Multi-objective ILC with weighted distance and acceleration, w_l and w_a.

Figure 3.5: A comparison of calculated kinetic energy for various ILC designs.

Table 3.1: Weighting matrix gains for the simulation testing

	q	s	r	w_l	w_a
Conventional ILC	1	$5e^{-4}$	$1e^{-8}$	0	0
Point-to-point ILC	1	$5e^{-4}$	$1e^{-8}$	0	0
Multi-objective ILC, w_l	1	$5e^{-4}$	$1e^{-8}$	1	0
Multi-objective ILC, w_a	1	$5e^{-4}$	$1e^{-8}$	0	1
Point-to-point ILC, $s = 6e^{-3}$	1	$6e^{-3}$	$1e^{-8}$	0	0
Point-to-point ILC, $s = 1$	1	1	$1e^{-8}$	0	0

constant velocity energy use. The effects of these two parameters on kinetic energy are evaluated in simulation with results provided in Table 3.2.

3.4.2 Simulation Results

For the simulation validation, distance and acceleration were included as additional performance metrics within the multi-objective ILC framework. The q weighting gain was fixed at 1 for all the cases; emphasizing trajectory tracking within the cost function. To simplify the trade-off analysis, noise and model uncertainty were not included in the simulation model. As such, the r weighting gain was selected as $1e^{-8}$; a value slightly larger than zero to mitigate computational inaccuracies that sometimes accompany the selection of $r = 0$. The s weighting gain was selected as $5e^{-4}$ for the nominal cases; a heuristically determined value that ensured convergence with appropriate system robustness in the experimental testing. To evaluate the impact of weighting the control input with an additional performance metric aimed at energy reduction, two additional values of s, $\{s = 1, s = 6e^{-3}\}$, were selected and tested using the point-to-point ILC framework. In addition to the s gain, the weighting gains associated with the distance and acceleration parameters, $\{w_l, w_a\}$, were evaluated at values of 0 and 1. An overview of all gain selections for the different controllers is provided in Table 3.1.

The multi-objective learning controller resulted in a trade-off between trajectory tracking, energy use, and overall distance traveled. Table 3.2 provides an overview

of several performance metrics for different learning controllers: the RMS trajectory tracking error (3.12); overall distance traveled (3.13); average kinetic energy usage (3.14); and the sum of the squared control inputs or control effort (3.15).

$$E_{j,RMS}=\sqrt{\frac{1}{N}\sum_{k=1}^{N}E_{x,j}^2(k)+\frac{1}{N}\sum_{k=1}^{N}E_{y,j}^2(k)} \tag{3.12}$$

$$L_j=\sum_{k=1}^{N}\sqrt{l_{x,j}^2(k)+l_{y,j}^2(k)} \tag{3.13}$$

$$\overline{KE}_j=\frac{1}{N}\left(\frac{1}{2}m\sum_{k=1}^{N}\sqrt{v_{x,j}^2(k)+v_{y,j}^2(k)}\right) \tag{3.14}$$

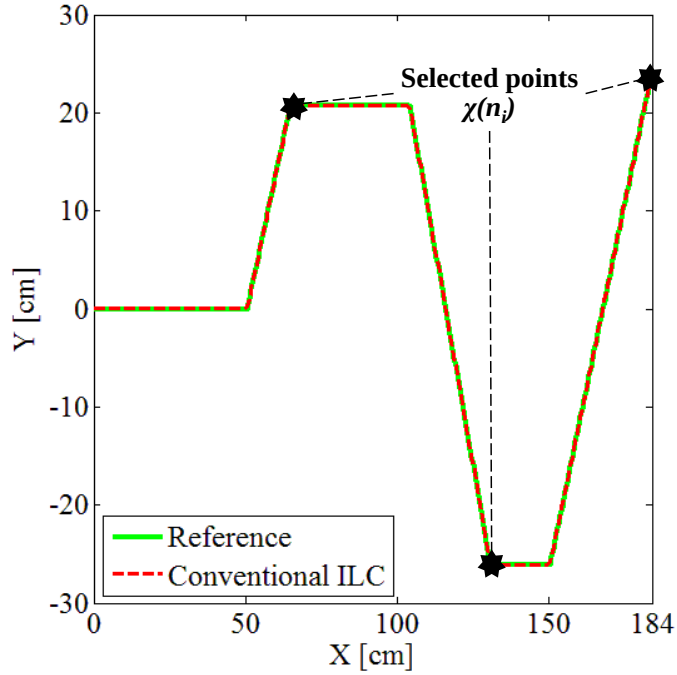
$$U_j=\sum_{k=1}^{N}\sqrt{u_{x,j}^2(k)+u_{y,j}^2(k)} \tag{3.15}$$

As can be identified from this table and illustrated in Figure 3.6, an emphasis on distance or acceleration results in a small degradation in the tracking performance. However, this is achieved at the reduction of energy usage and overall distance traveled; an acceptable trade-off for many systems.

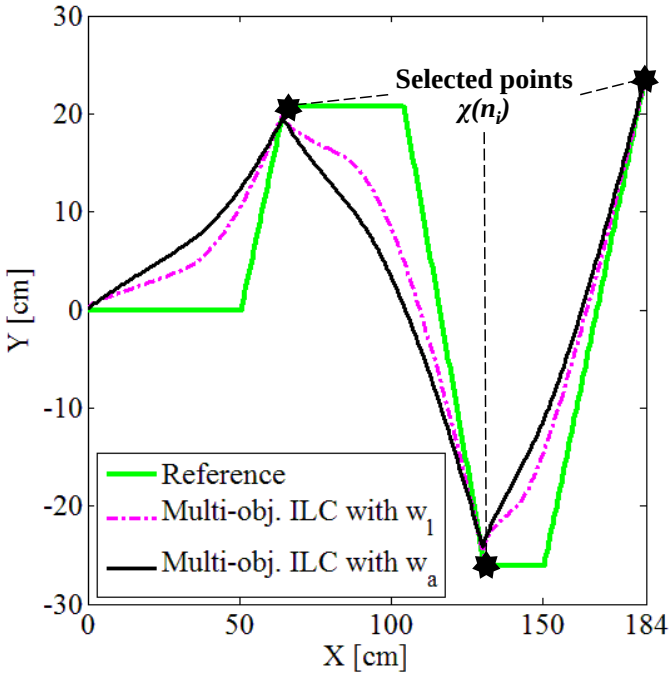
For the example system, weighting the distance traveled resulted in an 18% reduction in kinetic energy use and a 3% reduction in overall distance traveled over point-to-point ILC.

As a comparison for the distance example provided above, kinetic energy use due to acceleration was also investigated. For the given robotic system, weighting acceleration resulted in an 8% reduction in kinetic energy use and virtually no reduction in distance traveled as compared to point-to-point ILC. With similar tracking performances, the improvement in the kinetic energy use as a function of weighting the acceleration did not provide satisfactory performance trade-offs to warrant use in this example.

To evaluate the impact of weighting the control signal directly, consider the two cases provided in Table 3.2. A selection of $s = 1$ provided results that were well



(a) Trajectory tracking with feedback and conventional ILC.



(b) Tracking performance for multi-objective ILC designs weighting distance and acceleration, w_l and w_a.

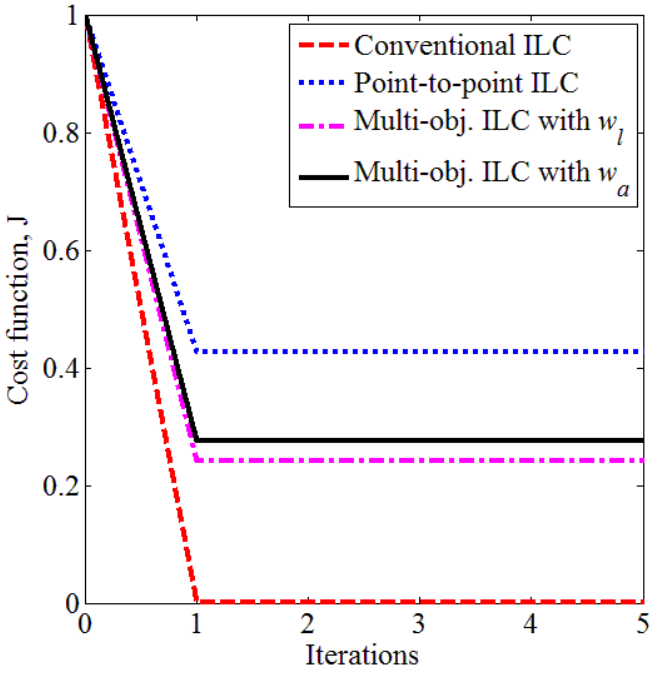
Figure 3.6: Simulation results.

Table 3.2: Comparison of performance metrics for various controllers: simulation

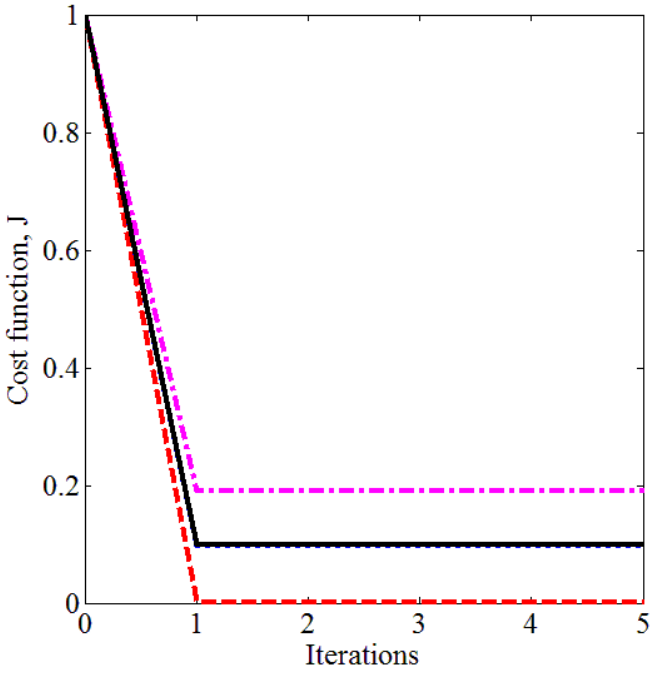
	RMS error [cm]	**Distance [cm]**	**Average K.E. [mJ]**	**Control Effort**
Conventional ILC	0.33	247.83	8.85	$1.1e^6$
Point-to-point ILC	0.89	227.00	8.50	$1.0e^6$
Multi-objective ILC, w_l	1.59	219.21	7.00	$8.1e^5$
Multi-objective ILC, w_a	1.27	226.93	7.78	$9.4e^5$
Point-to-point ILC, $s = 6e^{-3}$	7.15	209.99	7.02	$8.2e^5$
Point-to-point ILC, $s = 1$	17.79	188.24	5.07	$5.5e^5$

outside the acceptable range for improved trajectory tracking, despite the significant improvements observed in the kinetic energy and control effort. To provide a more appropriate comparison, the s gain was tuned until similar kinetic energy and control effort results were seen as compared to the multi-objective ILC controller with the w_l weighting gain. For this example, a point-to-point ILC controller with $s = 6e^{-3}$ resulted in comparable energy metrics, but a significant degradation in the trajectory tracking metric; over 4 times larger as compared to the multi-objective learning controller with weighting on the distance traveled. These results clearly indicate the effects of using the control signal as the performance metric for optimized energy. Importantly, the control effort (sum of the squared control inputs) trends match the trends observed with the average kinetic energy. For both calculations, the optimal controller in terms of reduced energy use while maintaining significant trajectory tracking improvements was the multi-objective ILC controller with weighting on the distance traveled.

An important consideration in ILC controller design is convergence over the iteration domain. Figure 3.7 illustrates the convergence characteristic of the cost function over iteration space. The cost function was numerically calculated from the simulation results. As can be seen from the figure, the four main controllers demonstrate monotonic convergence properties for the cost function. This indicates that the system does not exhibit any significant transient behavior in the search for an optimal



(a) For X-axis



(b) For Y-axis

Figure 3.7: Convergence properties of the cost function

solution; an important robustness consideration.

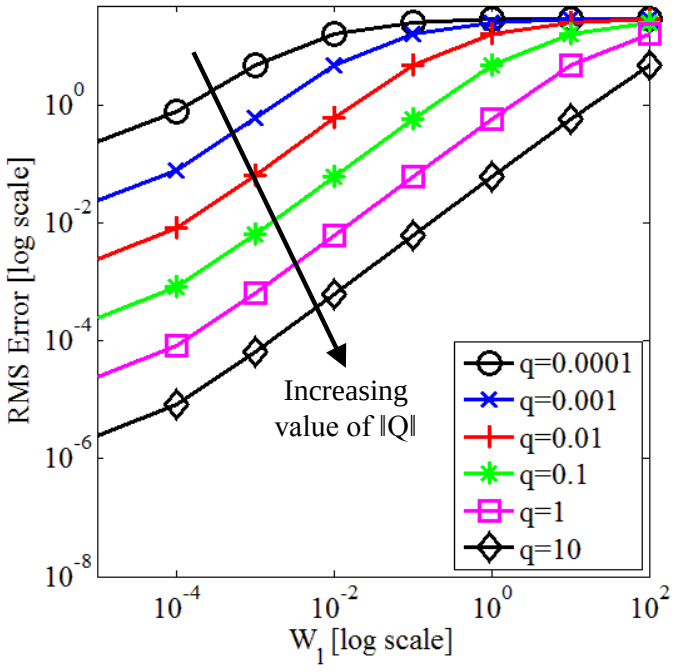
3.4.3 Trade-off Analysis Between Tracking Performance and Energy

To evaluate the trade-off between performance and kinetic energy usage as a function of distance traveled and acceleration, two case studies were considered. In case study 1, the weighting gains s and r were fixed at $1e^{-8}$ based on the assumption of no noise or model uncertainty. To evaluate the trade-off between trajectory tracking and weighting the distance traveled, q and w_l were varied over the ranges $q \in [1e^{-4}, 10]$ and $w_l \in [1e^{-5}, 100]$, while w_a was set to $w_a = 0$.

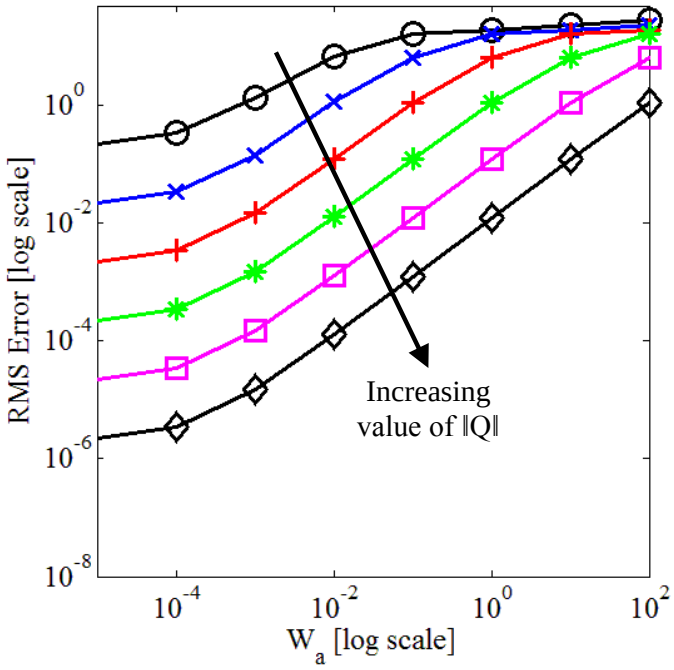
Case 2 evaluated the trade-off between trajectory tracking and weighting acceleration. Similar to case 1, s and r were fixed at $1e^{-8}$. The weighting gains q and w_a were varied over the ranges $q \in [1e^{-4}, 10]$ and $w_a \in [1e^{-5}, 100]$, while w_l was set to $w_l = 0$.

Figures 3.8 (a) and (b) present the effects of varying $\mathbf{W}_l$ and $\mathbf{W}_a$ gain matrices on RMS trajectory tracking errors at the selected points. The different lines represent varying values for $\mathbf{Q}$. For example, a selection of $\|\mathbf{Q}\| = 0$ (top black lines with circle markers) with varying $\mathbf{W}_l$ or $\mathbf{W}_a$ gain matrices results in minimal changes to the RMS error. However, as $\|\mathbf{Q}\|$ increases (i.e. increased emphasis on tracking at the selected points), varying the $\mathbf{W}_l$ or $\mathbf{W}_a$ gain matrices significantly affects the RMS error. As predicted, increasing the energy based weighting gains results in reduced energy usage, Figures 3.9 (a) and (b), with varying rates of decay based on the $\mathbf{Q}$ gain matrix.

To evaluate the combined performance metrics, Figures 3.10 (a) and (b) illustrate the tracking performance and energy usage as a function of varying $\mathbf{Q}$, $\mathbf{W}_l$, and $\mathbf{W}_a$ gain matrices. As can be seen from these figures, to maximize the performance metrics simultaneously, the q gain should be set to 1, while the w_l or w_a gains should start small and increase until an energy plateau is achieved. After this

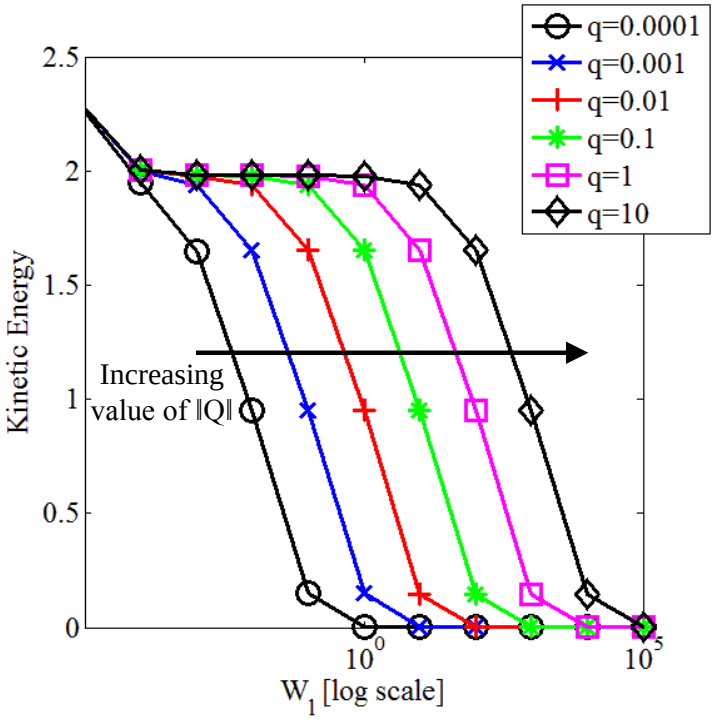


(a) Varying $\mathbf{Q}$ and $\mathbf{W}_l$

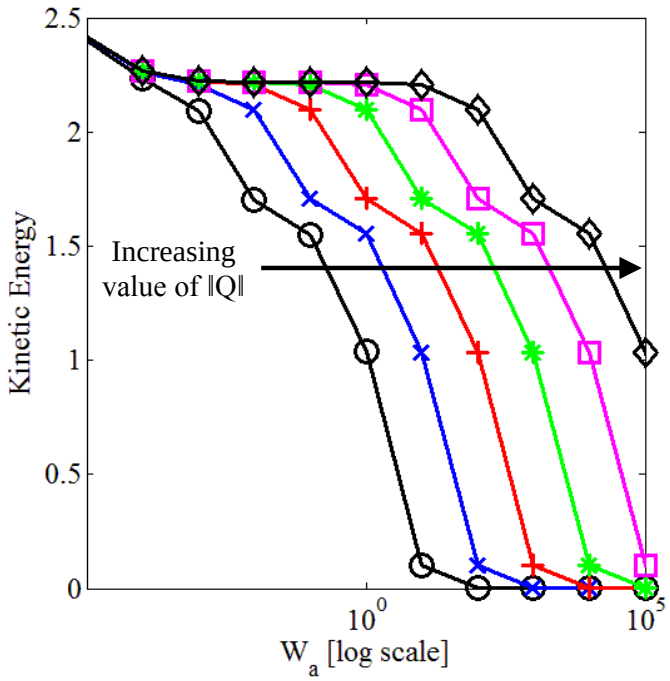


(b) Varying $\mathbf{Q}$ and $\mathbf{W}_a$

Figure 3.8: RMS error

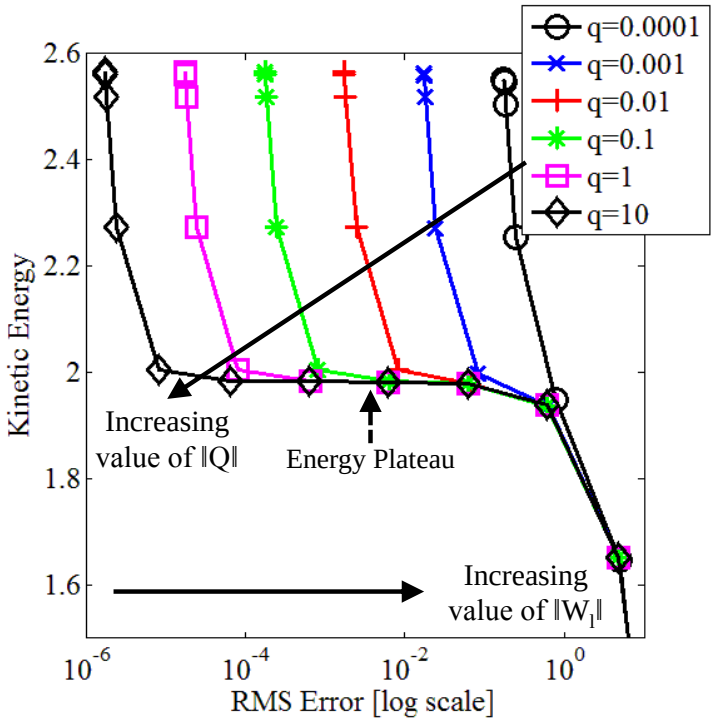


(a) Varying $\mathbf{Q}$ and $\mathbf{W}_l$

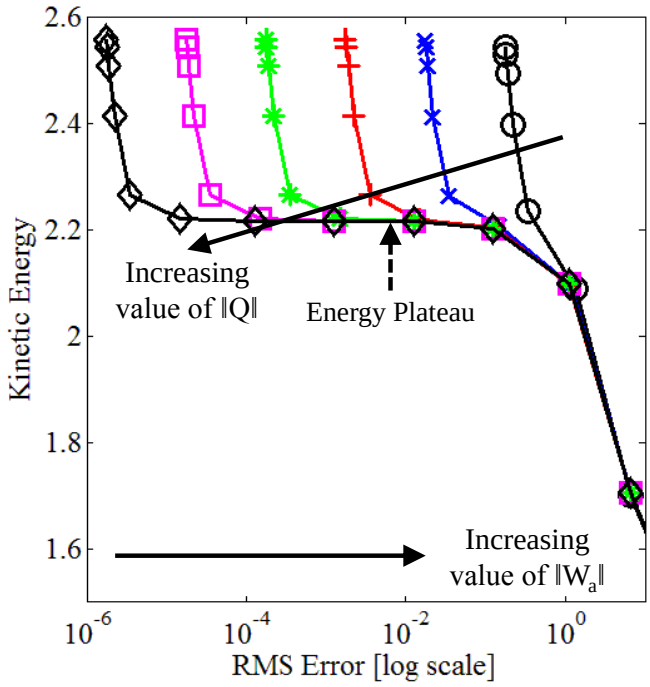


(b) Varying $\mathbf{Q}$ and $\mathbf{W}_a$

Figure 3.9: Energy usage



(a) Varying $\mathbf{Q}$ and $\mathbf{W}_l$



(b) Varying $\mathbf{Q}$ and $\mathbf{W}_a$

Figure 3.10: Tradeoffs between tracking and energy

point, decreased energy usage can only be achieved with significant degradation of the tracking performance.

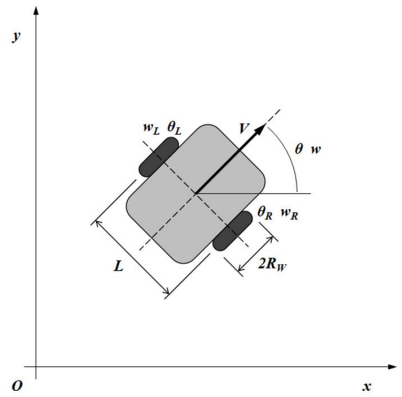
3.5 Experimental Validation

3.5.1 Experimental Setup

To validate the proposed learning framework, experimental tests with a wheeled mobile robot were conducted. The robot moves on two wheels driven by independent electric motors. A gyroscope is mounted at the center of the wheels to measure position and heading angle of the robot. Figure 3.11(a) shows the wheeled mobile robot used in this study. For additional details regarding the wheeled mobile robot please see Appendix A.

A kinematic model is used to describe the dynamics of the mobile robot used in these experiments, Figure 3.11(b). This model describes the relationship between the derivatives of the robot position and orientation with the robot's linear and angular speeds [57, 58, 59, 60, 61, 62, 63]. The vehicle velocity, position, and heading angle

(a) Unicycle mobile wheeled robot, LEGO NXT Mindstorm



(b) Planar schematic of the vehicle

Figure 3.11: Experimental setup

are determined using the given differential equations [58].

$$V = R_w \frac{(\omega_R + \omega_L)}{2} \tag{3.16}$$

$$\dot{\theta} = \omega = R_w \frac{(\omega_R - \omega_L)}{L} \tag{3.17}$$

$$\dot{x} = V \cos\theta = R_w \frac{(\omega_R + \omega_L)}{2} \cos\theta \tag{3.18}$$

$$\dot{y} = V \sin\theta = R_w \frac{(\omega_R + \omega_L)}{2} \sin\theta \tag{3.19}$$

In Eqns (3.16)-(3.19) and Figure 3.11(b), R_w is the wheel radius, L is the distance between wheels, θ is the heading angle of the vehicle, θ_R is the rotation angle of the right wheel, θ_L is the rotation angle of the left wheel, ω is the angular velocity of the vehicle, ω_R is the angular velocity of the right wheel, ω_L is the angular velocity of the left wheel, and V is the vehicle speed.

Closed-loop models of the x and y-axis were experimentally identified using a series of step inputs and the Matlab function 'ident'. Desired x and y input signals were compared with the x and y output positions derived from the measured angular velocities of the wheels, the wheel radius, and the distance between the wheels using the transformations presented in (3.16)-(3.19). In practice, a MIMO system will best represent the system as,

$$H(z) = \begin{bmatrix} H_{XX}(z) & H_{YX}(z) \\ H_{XY}(z) & H_{YY}(z) \end{bmatrix}.$$

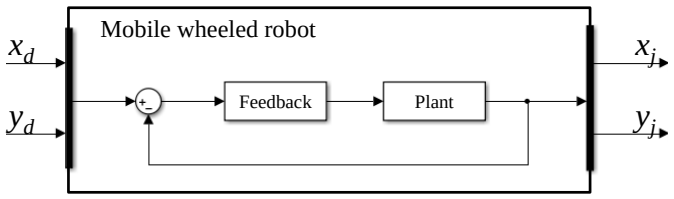


Figure 3.12: Simplified block diagram of the wheeled mobile robot.

However, $H_{YX}(z)$ and $H_{XY}(z)$ are relatively small compared to $H_{XX}(z)$ and $H_{YY}(z)$. Therefore, to simplify the system for the sake of modeling and control design we ignored the off-axis functions. From the experimental system identification, the dynamically decoupled discrete time closed-loop transfer function for each axis is given as,

$$H_X(z) = H_{XX}(z) = \frac{0.002838z^{-1} - 9.966e^{-05}z^{-2} - 0.002576z^{-3}}{1 - 2.782z^{-1} + 2.576z^{-2} - 0.7935z^{-3}}$$
$$H_Y(z) = H_{YY}(z) = \frac{0.002374z^{-1} - 0.0001206z^{-2} - 0.002227z^{-3}}{1 - 2.829z^{-1} + 2.664z^{-2} - 0.8348z^{-3}}$$

with a sampling time of 0.05 seconds.

For the experimental validation, distance and acceleration were included as additional performance metrics within the multi-objective ILC framework. The s and r weighting gains were heuristically tuned, then fixed at $5e^{-4}$ and $1e^{-4}$, respectively, while $\{q, w_l, w_a\}$ were selected as 0 or 1 depending on the desired controller. The individual gains for each controller are provided in Table 3.3.

3.5.2 Experimental Results

As predicted by the analysis and simulation results, a multi-objective learning controller resulted in a trade-off between trajectory tracking, average kinetic energy usage, and overall distance traveled. Table 3.4 lists the RMS error, average kinetic energy usage, control effort (the sum of the square of the control signal), and overall distance traveled for several control approaches. As predicted by the simulation

Table 3.3: Weighting matrix gains for the experimental testing

	q	s	r	w_l	w_a
Conventional ILC	1	$5e^{-4}$	$1e^{-8}$	0	0
Point-to-point ILC	1	$5e^{-4}$	$1e^{-8}$	0	0
Multi-objective ILC, w_l	1	$5e^{-4}$	$1e^{-8}$	1	0
Multi-objective ILC, w_a	1	$5e^{-4}$	$1e^{-8}$	0	1

results and presented in this table and Figure 3.13, an emphasis on distance or acceleration results in a degradation in the tracking performance. However, this is achieved at the reduction of energy usage and overall distance traveled.

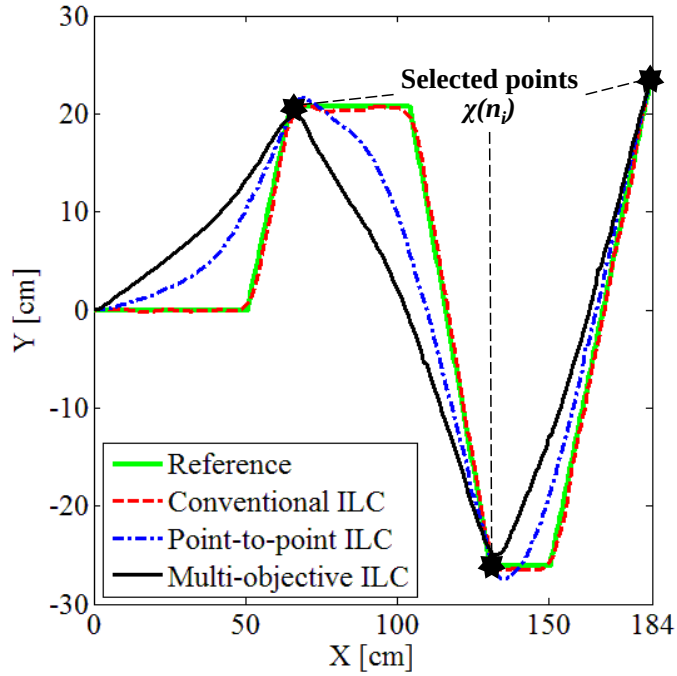
For the example system, the weighting on the distance traveled resulted in a 26% reduction in average kinetic energy usage (32% reduction in control effort) and a 5% reduction in overall distance traveled over point-to-point ILC. Assuming that the mobile robot runs over 10 hours with a single charge, this reduction in energy translates to an additional 2.1 hours of running time for this particular robotic system, a significant increase that could lead to longer deployment time or additional iterations of a given task.

As a comparison for the distance example provided above, energy usage due to acceleration was also investigated. For the given robotic system, weighting acceleration resulted in only a 17% reduction in kinetic energy use (9% reduction in control effort) as compared to point-to-point ILC, translating to just over 0.8 hour of additional running time. As the simulation results predicted, the optimal design choice for this experimental testbed is a multi-objective weighting on the trajectory tracking and distance travelled.

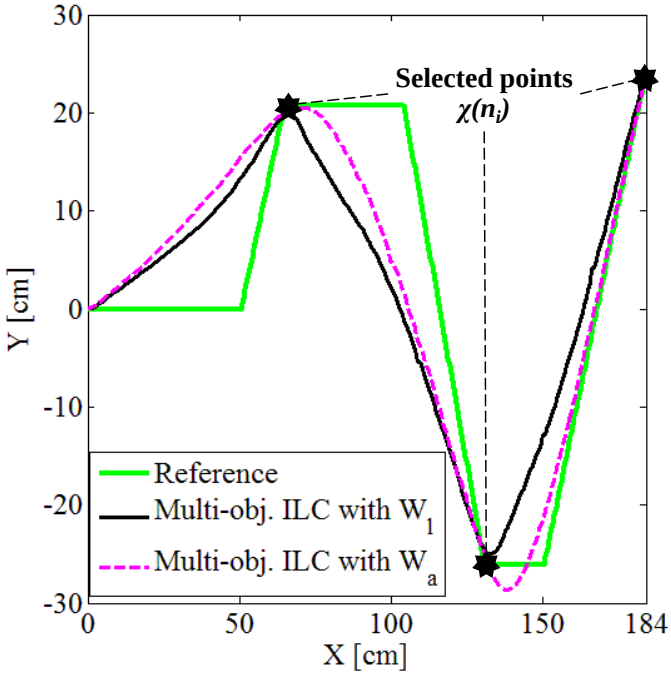
As a final evaluation, the cost function convergence for the experimental testbed was determined. Figure 3.14 presents the convergence pattern for the four main learning controllers. As can be seen from this figure, the cost function converges in a near monotonic manner. Slight discrepancies are attributed to non-repetitive

Table 3.4: Comparison of performance metrics for various control approaches.

	RMS error [cm]	**Distance [cm]**	**Average K.E. [mJ]**	**Control Effort**
Conventional ILC	1.13	246.67	8.94	$1.1e^6$
Point-to-point ILC	1.02	232.62	10.04	$1.3e^6$
Multi-objective ILC, w_l	1.59	221.57	7.40	$8.8e^5$
Multi-objective ILC, w_a	1.22	230.53	8.30	$1.0e^6$

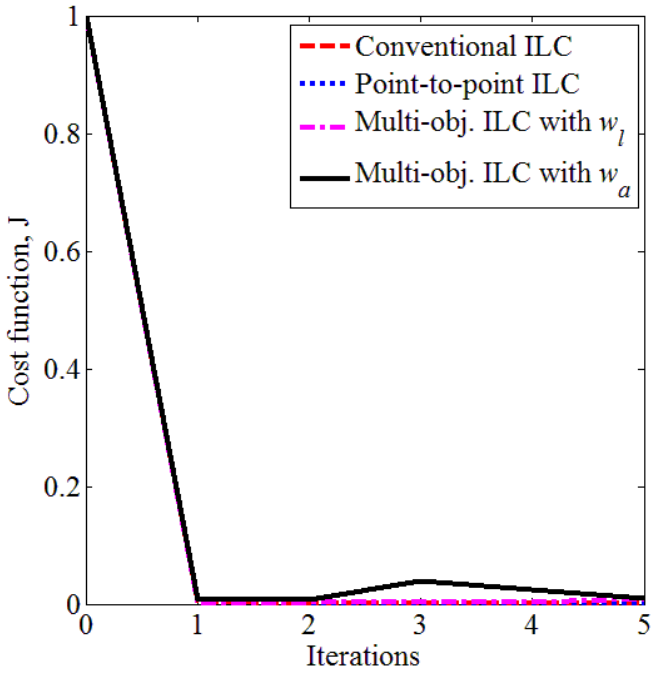


(a) Conventional ILC, Point-to-point ILC, and Multi-objective ILC with w_l

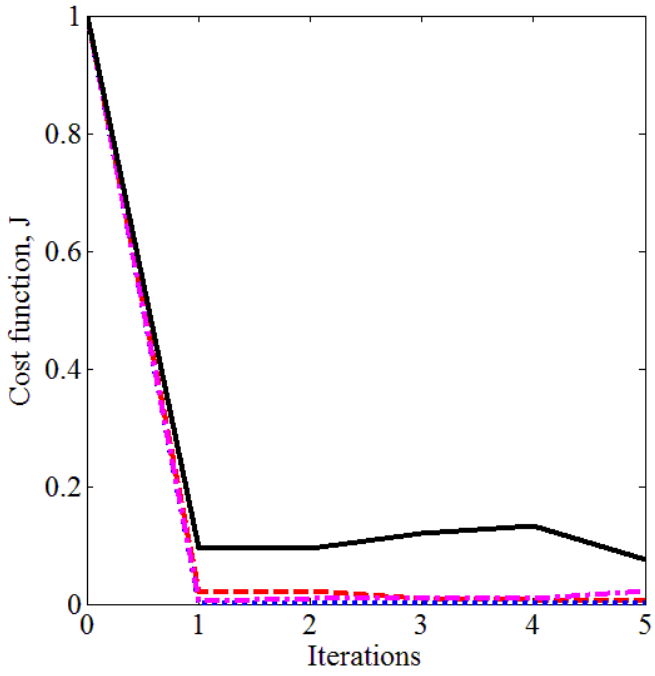


(b) Multi-objective ILC with w_l and w_a

Figure 3.13: Experimental results



(a) For X-axis



(b) For Y-axis

Figure 3.14: Experimental convergence properties of the cost function

disturbances and noise that have been shown to alter monotonicity [54].

3.6 Concluding Remarks

This chapter presents the extension of ILC to the application of a mobile wheeled robot system. The specific performance objectives were identified as (1) selected points of tracking, and (2) energy savings. By considering the trade-offs between the performance objectives, the results for several different controller designs were used to derive comparisons between the different cases. The specific contributions of this chapter include: (1) the development of an initial framework for considering multiple performance objectives within an iterative learning control framework, (2) an analytical discussion of the trade-off benefits of the multi-objective learning approach for a specific set of performance objectives, and (3) simulation and experimental validations of the new control approach on a robotic test platform.

The experimental testing demonstrated the trade-off benefits of the multi-objective learning approach with a 26% reduction in energy usage over point-to-point ILC methods, while still maintaining relatively low tracking errors at the selected points. As illustrated by the experimental example, the flexibility in the proposed framework enables the user to personalize the control design to specific cases, a necessary requirement for generalizing the framework across various applications. While these performance trade-offs are demonstrated with a simple robotic platform, they indicate the potential for this framework to achieve substantial cost, time, or energy usage savings by leveraging the additional control freedom available in point-based learning problems.

CHAPTER IV

Multi-Objective Iterative Learning Control in Spatial Domain

While conventional ILC has been constructed in the temporal domain, a new emphasis in additive manufacturing (AM) has driven the need for a spatially focused learning framework. AM defines a process of laying down material, layer by layer. As such, spatial interactions within the deposited material dominate system behavior.

In this chapter, we extend the multi-objective iterative learning control framework into the spatial domain. As such, the description of system dynamics and critical performance metrics is modified from a 1D temporal impulse response to a 2D spatial impulse response. The target application in this chapter is an ink deposition system. The key performance metrics are identified as (1) image quality, and (2) material usage. A simulation validation is provided.

4.1 Preliminaries

4.1.1 Spatial Convolution

In this chapter, the authors investigate multi-objective iterative learning control in the spatial domain. The authors introduced the concept of a spatial iterative learning controller in [42]. A complete description of spatial ILC can be found in

more recent work in [49]. Here we present only the salient details from [49] necessary for the introduction of a multi-objective learning controller in the spatial domain. For additional details and a more thorough discussion of the spatial learning framework, the readers are invited to see [42, 49].

A 2D spatial input map $u(x, y)$ and a plant operator H yield a spatial output map $g(x, y)$,

$$g(x, y) = Hu(x, y), \tag{4.1}$$

where $u(x, y) \in \mathbb{R}^{A\times B}$, $g(x, y) \in \mathbb{R}^{A\times B}$, and x and y are integer valued coordinates that discretize the spatial map; $x = 0, 1, \ldots, A-1$ and $y = 0, 1, \ldots, B-1$. The plant operator H is represented by a spatial impulse response centered around a point (m, n) [64]:

$$H\delta(x - m, y - n) = h(x - m, y - n),$$

where $h(x-m, y-n) \in \mathbb{R}^{C\times D}$. Assuming a spatially-invariant operator H, (4.1) can be computed using the 2D convolution sum,

$$\begin{aligned} g_e(x, y) &= u_e(x, y) * h_e(x, y) \\ &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} u_e(m, n) h_e(x + M^{\triangle} - m, y + N^{\triangle} - n), \end{aligned} \tag{4.2}$$

where the subscript e denotes that each spatial map is an extended version of the original map that has been appended by zeros. The sizes of the extended maps are $u_e(x, y) \in \mathbb{R}^{M\times N}$, $g_e(x, y) \in \mathbb{R}^{M\times N}$, and $h_e(x, y) \in \mathbb{R}^{M\times N}$, with M and N defined as,

$$\begin{aligned} M &= A + C - 1 + (A + C) \bmod 2 \\ N &= B + D - 1 + (B + D) \bmod 2. \end{aligned} \tag{4.3}$$

Equation (4.3) is used to set the size of M and N and ensure that they are odd numbers such that each function has a defined center coordinate; we define the center

coordinate as $M^{\triangle} = \lfloor M/2 \rfloor + 1$ and $N^{\triangle} = \lfloor N/2 \rfloor + 1$, where $\lfloor \cdot \rfloor$ denotes a floor operation. The derivations of these extended forms can be found in [49]. From here forward we assume that all 2D functions have been extended and will remove the subscript e for brevity.

The spatial variables from (4.1) can equivalently be written as $M \times N$ complex matrices. Conversion of these matrices to $MN \times 1$ column vectors is commonly referred to as vectorization.

Definition 1. *[49] An $A \times B$ 2D complex matrix can be converted to an $AB \times 1$ column vector using the vectorization operator $\mathcal{V}$.*

$$\begin{bmatrix} p(0,0) & p(0,1) & \cdots & p(0,B-1) \\ p(1,0) & p(1,1) & \cdots & p(1,B-1) \\ \vdots & \vdots & \ddots & \vdots \\ p(A-1,0) & p(A-1,1) & \cdots & p(A-1,B-1) \end{bmatrix}$$

$$\mathbf{p} \triangleq \mathcal{V}(p) \triangleq \begin{bmatrix} p(0,0) \\ \vdots \\ p(0,B-1) \\ p(1,0) \\ \vdots \\ p(A-1,B-1) \end{bmatrix}$$

The reverse of vectorization will be referred to as matricization and denoted $\mathcal{V}^{-1}$: this notation is used for brevity and does not satisfy the properties of an inverse.

4.1.2 Circulant Matrices

The matrix plant operator $\mathbf{H}$ is a Block circulant matrix composed of circulant blocks (BCCB matrix) [49, 64, 65],

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}^c_{M^\triangle} & \mathbf{H}^c_{M^\triangle-1} & \cdots & \mathbf{H}^c_0 & \mathbf{H}^c_{M-1} & \cdots & \mathbf{H}^c_{M^\triangle+1} \\ \mathbf{H}^c_{M^\triangle+1} & \mathbf{H}^c_{M^\triangle} & \cdots & \mathbf{H}^c_1 & \mathbf{H}^c_0 & \cdots & \mathbf{H}^c_{M^\triangle+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ \mathbf{H}^c_{M-1} & \mathbf{H}^c_{M-2} & \cdots & \mathbf{H}^c_{M^\triangle} & \mathbf{H}^c_{M^\triangle-1} & \cdots & \mathbf{H}^c_0 \\ \mathbf{H}^c_0 & \mathbf{H}^c_{M-1} & \cdots & \mathbf{H}^c_{M^\triangle+1} & \mathbf{H}^c_{M^\triangle} & \cdots & \mathbf{H}^c_1 \\ \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}^c_{M^\triangle-1} & \mathbf{H}^c_{M^\triangle-2} & \cdots & \mathbf{H}^c_{M-1} & \mathbf{H}^c_{M-2} & \cdots & \mathbf{H}^c_{M^\triangle} \end{bmatrix}, \tag{4.4}$$

where each $\mathbf{H}^c_k$ is circulant and contains the spatial impulse response data,

$$\mathbf{H}^c_k = \begin{bmatrix} h(j,N^\triangle) & h(j,N^\triangle-1)\cdots h(j,N^\triangle+1) \\ h(j,N^\triangle+1) & h(j,N^\triangle) & \cdots h(j,N^\triangle+2) \\ \vdots & \vdots & \ddots & \vdots \\ h(j,N^\triangle-1) h(j,N^\triangle-2)\cdots & h(j,N^\triangle) \end{bmatrix}.$$

Circulant matrices have interesting properties that enable a significant reduction in the computational cost of calculating matrix products and norms [65, 66, 67].

- A generic circulant matrix $\mathbf{H}^c \in \mathbb{C}^{N\times N}$ is defined by first row $[\, h(0,0),\ h(0,1),\ \cdots,\ h(0,N-1)\,]$.
- All circulant matrices are diagonalizable, $\mathbf{H}^c = \mathbf{W}^{-1}\mathbf{\Lambda}\mathbf{W}$, where $\mathbf{W} = \frac{1}{\sqrt{N}}[\mathbf{w}^{(0)}, \mathbf{w}^{(1)}, \cdots, \mathbf{w}^{(N-1)}]$ and $\mathbf{\Lambda} = diag\,(\,\lambda_0,\,\lambda_1,\,\cdots,\,\lambda_{N-1}\,)$.
- All eigenvector matrices consist of the exact same set of eigenvectors $\mathbf{w}^{(n)}$ for $n = 0, 1, \cdots, N-1$ and λ_n are the corresponding eigenvalues calculated from the first row of $\mathbf{H}^c$.
- Two circulant matrices $\mathbf{H}^c_1$ and $\mathbf{H}^c_2$ commute, $\mathbf{H}^c_1\mathbf{H}^c_2 = \mathbf{H}^c_2\mathbf{H}^c_1$.
- The sum and product of two circulant matrices, $\mathbf{H}^c_1$ and $\mathbf{H}^c_2$, are circulant.

The properties of circulant matrices can also be applied to BCCB matrices. All BCCB matrices are diagonalizable and can be written in the form of $\mathbf{H} = \mathbf{W}^{-1}\mathbf{\Lambda}\mathbf{W}$

[49]. In this format, $\mathbf{\Lambda}$ is a block diagonal eigenvalue matrix, $\mathbf{\Lambda} = diag(\Lambda_0, \Lambda_1, \cdots, \Lambda_{M-1})$, where each block is a diagonal matrix of a subset of the eigenvalues of $\mathbf{H}$, and the eigenvector matrix $\mathbf{W}$ is symmetric and unitary, $\mathbf{W}^{-1} = \mathbf{W}^*$, where * denotes the Hermitian conjugate.

We use the operator $\rho(\cdot)$ to denote the spectral radius and $\bar{\sigma}(\cdot)$ to denote the maximum singular value of a 2D function or matrix.

Definition 2. *[49] The Frobenius norm of a function p is equivalent to the vector 2-norm of the vectorized function* $\mathbf{p}$*,*

$$\|p\|_F = \|\mathbf{p}\|_2$$

Lemma IV.1. *[49] Given a BCCB matrix* $\mathbf{H}$*, the eigenvalues of* $\mathbf{H}$ *satisfy the equality* $\rho(\mathbf{H}) = \bar{\sigma}(\mathbf{H})$*.*

Proof of Lemma IV.1 : *This proof comes from the previous works in [42, 49]. Given a BCCB matrix* $\mathbf{H}$*, the maximum singular value of* $\mathbf{H}$ *can be defined as,*

$$\begin{aligned}\bar{\sigma}(\mathbf{H}) &= \sqrt{\rho(\mathbf{H}^*\mathbf{H})} \\ &= \sqrt{\mathbf{W}^{-1}\mathbf{\Lambda}^*\mathbf{\Lambda}\mathbf{W}} \\ &= \rho(\mathbf{H})\end{aligned}$$

by the diagonalizability of BCCB matrices (see Section 4.1) and the property that $\mathbf{W}^{-1} = \mathbf{W}^*$ *[66, 67].* □

4.1.3 Spatial ILC

Consider the 2D spatial map in (4.1) with the addition of noise η. Applying a convolution mapping from (4.2) and the vectorization operator $\mathcal{V}$, a lifted system

representation of the 2D spatial system can be given [49],

$$\mathbf{g} = \mathbf{H}\mathbf{u} + \boldsymbol{\eta}.$$

This format is equivalent to the lifted-form typically used in temporal ILC [23]. The system spatial dynamics are captured in the block circulant matrix $\mathbf{H}$ of the form given in (4.4).

4.1.4 Performance metric classifications

To consider multiple performance metrics, we define four different classifications that will be formalized in the manuscript.

Definition 3. *The primary performance metric is the main metric used to quantify performance of a system. For example, the performance of a positioning system is often evaluated by analysis of the trajectory tracking error. For these systems, trajectory tracking would be defined as the primary performance metric.*

Definition 4. *Classification I metrics are complementary performance metrics that can be written as a function of the primary performance metric. Weighting classification I performance metrics will results in a direct emphasis on the primary performance metric.*

Definition 5. *Classification II defines competing metrics, where the metric is written as a function of the system output. Competing objectives apply a minimization(maximization) requirement that competes against the primary objective. The use of these metrics requires a trade-off analysis to be performed between the competing performance metrics.*

Definition 6. *Classification III metrics are domain specific objectives that are defined with respect to the input signal. As with classification II, these objectives apply a*

minimization(maximization) requirement that competes with the primary objective. The use of these metrics requires a trade-off analysis to be performed between the competing performance metrics.

4.2 Multi-Objective ILC

In this section, a multi-objective iterative learning control framework is presented for the four categories of performance metrics defined above: primary, and Classifications I-III. Robust performance and convergence analyses are provided.

4.2.1 Initial formulation

We apply the following assumptions towards the design of a multi-objective iterative learning controller:

A1 The system governed by spatial dynamics is stable or stabilizable.

A2 The primary performance metric is defined as a subset of the output signal, $\chi(n_i) \subseteq g_d(k)$.

A3 The additional performance metrics are reachable; i.e. the reachability subspace $\mathcal{R} = \mathcal{Z}$ where the additional performance metrics $z_d(x, y) \in \mathcal{Z}$.

Assumption A1 is a standard requirement for ILC and will be applied in this framework. Assumption A2 focuses the primary objective on a set of discrete points, thus generating a mechanism for underutilized control effort to be present in the system. This control effort can then be applied towards the improvement of alternative performance metrics. While A2 is not a necessary condition for multi-objective ILC, it provides additional controller flexibility for enhanced performance of multiple objectives. Assumption A3 ensures that there exists an input that can drive the initial state of the additional performance metric to the desired final state.

A generalized cost function and ILC algorithm for the multi-objective optimization problem can be given as,

$$J=(\mathbf{\Psi e}_{j+1})^T\mathbf{Q}(\mathbf{\Psi e}_{j+1})+\mathbf{u}_{j+1}^T\mathbf{S}\mathbf{u}_{j+1}+(\mathbf{\Delta u}_{j+1})^T\mathbf{R}(\mathbf{\Delta u}_{j+1})+\sum_i \boldsymbol{\varepsilon}_{i,j+1}^T\mathbf{W}_i\boldsymbol{\varepsilon}_{i,j+1} \quad (4.5)$$

$$\mathbf{u}_{j+1} = \mathbf{L}_u\mathbf{u}_j + \mathbf{L}_e\mathbf{e}_j + \sum_i \mathbf{L}_i\boldsymbol{\varepsilon}_{i,j} \quad (4.6)$$

In (4.5), $\mathbf{\Psi}$ is used to identify the points of interest for the primary performance objective where $\mathbf{\Psi}$ is a diagonal matrix defined as [13]

$$\mathbf{\Psi}(k,k) = \begin{cases} 1, & k = n_k \\ 0, & otherwise. \end{cases}$$

In (4.5), $\mathbf{\Delta u}_{j+1} = (\mathbf{u}_{j+1} - \mathbf{u}_j)$ defines the changes in the control signal from iteration to iteration, $\boldsymbol{\varepsilon}_{i,j+1}$ denotes the error signal for the additional performance metrics, j denotes the iteration, and $\mathbf{W}_i$ are the weighting matrices applied to the additional performance metrics. For simplicity, we assume diagonal weighting matrices of the form $\mathbf{Q} = q\mathbf{I}$, $\mathbf{S} = s\mathbf{I}$, $\mathbf{R} = r\mathbf{I}$ and $\mathbf{W}_i = w_i\mathbf{I}$ where $\{q, s, r, w_i\}$ are positive scalars. $\mathbf{L}_i$ is a learning filter applied to the additional metrics within the update law (4.6).

4.2.2 Performance metrics

To organize the controller structure, the performance objectives are classified as primary and additional performance metrics. Performance evaluation is derived from the error term; i.e., the difference between the desired and measured values.

4.2.2.1 Primary performance metric

As denoted in assumption A2, the primary performance metric is defined as a set of discrete points along the full output signal; thus the selected points are defined as,

$$\begin{aligned}\boldsymbol{\Psi}\mathbf{e}_j &= \boldsymbol{\Psi}(\mathbf{g}_d - \mathbf{g}_j) \\ &= \boldsymbol{\Psi}(\mathbf{g}_d - \mathbf{H}\mathbf{u}_j).\end{aligned}$$

The additional performance metrics are classified into three categories defined in subsection 4.1.4 and mathematically derived below.

4.2.2.2 Classification I: Complementary metrics

The error signal for the complementary metrics is defined as a linear mapping of the baseline error,

$$\boldsymbol{\varepsilon}_{i,j} = \mathbf{z}_{i,d} - \mathbf{z}_{i,j} = \mathbf{X}_i\mathbf{g}_d - \mathbf{X}_i\mathbf{H}\mathbf{u}_j = \mathbf{X}_i\mathbf{e}_j, \tag{4.7}$$

where $\boldsymbol{\varepsilon}_{i,j} \in \mathbb{R}^{N\times 1}$. In (4.7), $\mathbf{z}_{i,d} \in \mathbb{R}^{N\times 1}$ and $\mathbf{z}_{i,j} \in \mathbb{R}^{N\times 1}$ are the desired and measured output signals for the i^{th} objective, and $\mathbf{X}_i \in \mathbb{R}^{M\times M}$ is a linear operator that maps the output signal $\mathbf{g}_j = \mathbf{H}\mathbf{u}_j$ to the performance objective $\mathbf{z}_{i,j}$. Optimizing the cost function (4.5) with respect to $\mathbf{u}_{j+1}$ and (4.7) yields the modified update law (4.6) with learning filters,

$$\mathbf{C}_L = \Big\{(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{H}) + \sum_i (\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H}) + \mathbf{S} + \mathbf{R}\Big\}^{-1} \tag{4.8}$$

$$\mathbf{L}_u = \mathbf{C}_L\Big\{(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{H}) + \sum_i (\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H}) + \mathbf{R}\Big\} \tag{4.9}$$

$$\mathbf{L}_e = \mathbf{C}_L\Big\{\,(\boldsymbol{\Psi}\mathbf{H})^T\,\mathbf{Q}\boldsymbol{\Psi}\Big\} \tag{4.10}$$

$$\mathbf{L}_i = \mathbf{C}_L\Big\{\,(\mathbf{X}_i\mathbf{H})^T\,\mathbf{W}_i\Big\}. \tag{4.11}$$

From (4.7), one can note that design trade-offs for classification I metrics will stem from the classical performance versus robustness trade-off observed in conventional control design. Augmenting the cost function (4.5) to include complementary metrics will require a similar increase in the robustness weighting to maintain robustness properties. This trade-off will be illustrated in the robustness analysis presented in section 4.3.2.

4.2.2.3 Classification II: Competing metrics

In the second classification, the additional performance metrics are defined as functions of the system output, $\mathbf{z}_{i,j} = \mathbf{X}_i\mathbf{g}_j$; however, the desired value is not a function of the desired system output, $\mathbf{z}_{i,d} \neq \mathbf{X}_i\mathbf{g}_d$. For these additional metrics, the goal is to optimize the output value.

$$\boldsymbol{\varepsilon}_{i,j} = \mathbf{z}_{i,d} - \mathbf{z}_{i,j} = \mathbf{z}_{i,d} - \mathbf{X}_i\mathbf{H}\mathbf{u}_j \tag{4.12}$$

Optimizing (4.5) with respect to (4.6) and (4.12) yields the same learning filters from case I, (4.8)-(4.11). Note that the design of w_i will directly impact the primary performance metric; classification I will augment the primary performance, while classification II will compete against the primary performance.

4.2.2.4 Classification III: Domain specific metrics

In this classification, the additional performance metrics are defined as functions of the input dynamics not the output dynamics.

$$\boldsymbol{\varepsilon}_{i,j} = \mathbf{z}_{i,d} - \mathbf{z}_{i,j} = \mathbf{z}_{i,d} - \mathbf{T}_i\mathbf{u}_j \tag{4.13}$$

where $\mathbf{T}_i \in \mathbb{R}^{N\times N}$ is a linear operator that maps $\mathbf{u}_j$ to $\mathbf{z}_{i,j}$.

Substituting (4.13) into the cost function (4.5) yields the learning filters,

$$\mathbf{C}_L = \Big\{(\mathbf{\Psi H})^T \mathbf{Q}(\mathbf{\Psi H}) + \sum_i \mathbf{T}_i^T \mathbf{W}_i \mathbf{T}_i + \mathbf{S} + \mathbf{R}\Big\}^{-1} \tag{4.14}$$

$$\mathbf{L}_u = \mathbf{C}_L \Big\{(\mathbf{\Psi H})^T \mathbf{Q}(\mathbf{\Psi H}) + \sum_i \mathbf{T}_i^T \mathbf{W}_i \mathbf{T}_i + \mathbf{R}\Big\} \tag{4.15}$$

$$\mathbf{L}_e = \mathbf{C}_L \Big\{ (\mathbf{\Psi H})^T \mathbf{Q} \mathbf{\Psi} \Big\} \tag{4.16}$$

$$\mathbf{L}_i = \mathbf{C}_L \Big\{ \mathbf{T}_i^T \mathbf{W}_i \Big\}. \tag{4.17}$$

4.3 Controller Analysis and Design

In this section, convergence and performance analyses for the multi-objective controller are provided.

4.3.1 Nominal convergence

Many practical systems may experience large transient learning growth during asymptotic convergence: rapid growth in the error signal over many iterations before convergence [5]. To eliminate this phenomena, we apply the more conservative requirement for monotonic convergence: guaranteed performance improvements from iteration to iteration [1].

Theorem IV.1. *Given the update law* (4.6)*, learning filters* (4.8)-(4.11)*, and physical system* $\mathbf{g}_j = \mathbf{H}\mathbf{u}_j$*, monotonic convergence of the system error,*

$$\Big\|\mathbf{e}_\infty - \mathbf{e}_{j+1}\Big\|_2 \leq \gamma \Big\|\mathbf{e}_\infty - \mathbf{e}_j\Big\|_2,$$

is guaranteed if and only if the convergence rate, $\gamma < 1$*, where*

$$\gamma \triangleq \Big\| \mathbf{H} \Big\{ (\mathbf{\Psi H})^T \mathbf{Q}(\mathbf{\Psi H}) + \sum_i (\mathbf{X}_i \mathbf{H})^T \mathbf{W}_i (\mathbf{X}_i \mathbf{H}) + \mathbf{S} + \mathbf{R} \Big\}^{-1} \mathbf{R} \mathbf{H}^{-1} \Big\|_2$$

$$\leq \frac{\bar{\sigma}(\mathbf{H})}{\underline{\sigma}(\mathbf{H})} \frac{r}{\underline{\sigma}\Big(q(\mathbf{\Psi H})^T(\mathbf{\Psi H}) + \sum_i w_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\Big) + s + r}.$$

Proof of Theorem IV.1 : *Theorem IV.1 is a statement of the contractibility of the system error. Rearranging* (4.6) *yields the mapping:*

$$\|\mathbf{e}_\infty - \mathbf{e}_{j+1}\|_2 \leq \gamma \|\mathbf{e}_\infty - \mathbf{e}_j\|_2,$$

which is a contraction mapping in terms of the 2-norm of the system error, when Theorem IV.1 is satisfied [68]. The convergence rate, γ is defined as

$$\begin{aligned}
\gamma &\triangleq \Big\| \mathbf{H}\Big\{ (\mathbf{\Psi H})^T \mathbf{Q} (\mathbf{\Psi H}) + \sum_i (\mathbf{X}_i\mathbf{H})^T \mathbf{W}_i (\mathbf{X}_i\mathbf{H}) + \mathbf{S} + \mathbf{R} \Big\}^{-1} \mathbf{R}\mathbf{H}^{-1} \Big\|_2 \\
&\leq \Big\| \mathbf{H} \Big\|_2 \Big\| \Big\{ (\mathbf{\Psi H})^T \mathbf{Q} (\mathbf{\Psi H}) + \sum_i (\mathbf{X}_i\mathbf{H})^T \mathbf{W}_i (\mathbf{X}_i\mathbf{H}) + \mathbf{S} + \mathbf{R} \Big\}^{-1} \mathbf{R} \Big\|_2 \Big\| \mathbf{H}^{-1} \Big\|_2 \\
&\leq \frac{\bar{\sigma}(\mathbf{H})}{\underline{\sigma}(\mathbf{H})} \frac{r}{\underline{\sigma}\Big(q(\mathbf{\Psi H})^T(\mathbf{\Psi H}) + \sum_i w_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\Big) + s + r}.
\end{aligned}$$

□

Corollary 1. *For additional performance metrics defined under classification III, monotonic convergence is guaranteed for $\gamma < 1$, where,*

$$\begin{aligned}
\gamma &\triangleq \Big\| \mathbf{H}\Big\{ (\mathbf{\Psi H})^T \mathbf{Q} (\mathbf{\Psi H}) + \sum_i \mathbf{T}_i^T \mathbf{W}_i \mathbf{T}_i + \mathbf{S} + \mathbf{R} \Big\}^{-1} \mathbf{R}\mathbf{H}^{-1} \Big\|_2 \\
&\leq \frac{\bar{\sigma}(\mathbf{H})}{\underline{\sigma}(\mathbf{H})} \frac{r}{\underline{\sigma}\Big(q(\mathbf{\Psi H})^T(\mathbf{\Psi H}) + \sum_i w_i \mathbf{T}_i^T \mathbf{T}_i\Big) + s + r}.
\end{aligned}$$

To evaluate the impact of the w_i gain, let convergence rate be written as $\gamma \leq C_1 \frac{r}{f(q,w_i)+r}$, where $f(q, w_i) = \underline{\sigma}\Big(q(\mathbf{\Psi H})^T(\mathbf{\Psi H}) + \sum_i w_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\Big) + s$ and $C_1 = \frac{\bar{\sigma}(\mathbf{H})}{\underline{\sigma}(\mathbf{H})}$. Assuming constant gains for everything but $[r, w_i]$, one can validate that increasing w_i results in a decrease in γ; faster convergence at the expense of increased

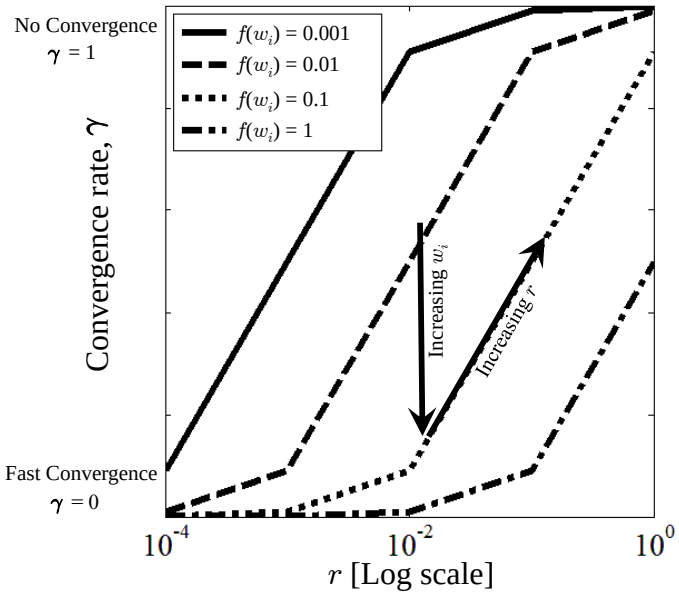


Figure 4.1: **Convergence Analysis:** For $\gamma \leq C_1 \frac{r}{f(q,w_i)+r}$: Increasing w_i reduces the convergence rate, leading to faster convergence at the expense of decreased noise attenuation. Increasing the r gain recovers the original trade-off between convergence rate and noise attenuation.

noise in the converged signal. To compensate for the effects of w_i and recover the value of γ, the r gain must be increased, see Fig. 4.1.

4.3.2 Robust convergence

For robust convergence, we consider the true system to correspond to the nominal model $\mathbf{H}$ with additive uncertainty $\mathbf{\Delta}$: $\mathbf{H_t} = \mathbf{H} + \hat{\mathbf{H}}\mathbf{\Delta}$, where $\hat{\mathbf{H}}$ is a lifted domain representation of a known spatial invariant operator and $\|\mathbf{\Delta}\|_2 \leq 1$. With the true system $\mathbf{H_t}$, the error is written as,

$$\mathbf{e}_j = \mathbf{g}_d - \mathbf{H_t}\mathbf{u}_j.$$

The requirement for robust convergence remains $\gamma < 1$, where

$$\gamma = \max_{\mathbf{\Delta}} \left\| \mathbf{H} \Big\{ (\mathbf{\Psi H})^T \mathbf{Q} (\mathbf{\Psi H}) + \sum_i (\mathbf{X}_i \mathbf{H})^T \mathbf{W}_i (\mathbf{X}_i \mathbf{H}) + \mathbf{S} + \mathbf{R} \Big\}^{-1} \right.$$

$$\left\|\mathbf{R}-\left((\mathbf{\Psi H})^T\mathbf{Q\Psi}+\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i\mathbf{X}_i\right)\hat{\mathbf{H}}\mathbf{\Delta}\right\}\mathbf{H}^{-1}\Big\|_2. \tag{4.18}$$

While (4.18) shows the requirement for robust monotonic convergence of systems with additional performance objectives defined under classification I and II, the presence of the uncertainty $\mathbf{\Delta}$ requires additional investigation.

Lemma IV.2. *Consider* (4.18) *with* $\|\mathbf{R}\|_2 = 0$*, then robust monotonic convergence is guaranteed if*

$$\Big\|\mathbf{H}\Big\|_2\Big\|\Big\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H})+\mathbf{S}\Big\}^{-1}$$
$$\Big\{(\mathbf{\Psi H})^T\mathbf{Q\Psi}+\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i\mathbf{X}_i\Big\}\hat{\mathbf{H}}\Big\|_2\Big\|\mathbf{H}^{-1}\Big\|_2<1.$$

Proof of Lemma IV.2 : *[8] Following directly from* (4.18) *and the inequality,*

$$\begin{aligned}
&\Big\|\mathbf{H}\Big\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\textstyle\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H})+\mathbf{S}\Big\}^{-1}\Big\{(\mathbf{\Psi H})^T\mathbf{Q\Psi}+\textstyle\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i\mathbf{X}_i\Big\}\hat{\mathbf{H}}\mathbf{\Delta}\mathbf{H}^{-1}\Big\|_2\\
&\le\Big\|\mathbf{H}\Big\|_2\Big\|\Big\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\textstyle\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H})+\mathbf{S}\Big\}^{-1}\Big\{(\mathbf{\Psi H})^T\mathbf{Q\Psi}+\textstyle\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i\mathbf{X}_i\Big\}\hat{\mathbf{H}}\Big\|_2\Big\|\mathbf{\Delta}\Big\|_2\Big\|\mathbf{H}^{-1}\Big\|_2\\
&\le\Big\|\mathbf{H}\Big\|_2\Big\|\Big\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\textstyle\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H})+\mathbf{S}\Big\}^{-1}\Big\{(\mathbf{\Psi H})^T\mathbf{Q\Psi}+\textstyle\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i\mathbf{X}_i\Big\}\hat{\mathbf{H}}\Big\|_2\Big\|\mathbf{H}^{-1}\Big\|_2\\
&<1.
\end{aligned}$$

□

Lemma IV.3. *Consider* (4.18) *with*

$$\Big\|\mathbf{H}\Big\|_2\Big\|\Big\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H})+\mathbf{S}\Big\}^{-1}$$
$$\Big\{(\mathbf{\Psi H})^T\mathbf{Q\Psi}+\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i\mathbf{X}_i\Big\}\hat{\mathbf{H}}\Big\|_2\Big\|\mathbf{H}^{-1}\Big\|_2<1.$$

Assume $(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H})+\mathbf{S}$ *to be invertible and positive definite.*

Then robust convergence is guarantee for all $\mathbf{R} = r\mathbf{I}$.

Proof of Lemma IV.3 : *[8]. Given* $(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H}) + \sum_i (\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H}) + \mathbf{S}$ *as an invertible and positive definite matrix, its singular value decomposition is*

$$(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H}) + \sum_i (\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H}) + \mathbf{S} = \mathbf{U\Sigma U}^T,$$

where $\mathbf{U}$ *a unitary matrix, and* $\mathbf{\Sigma}$ *is a diagonal matrix of full rank with diagonal elements* σ_i. *Furthermore, define* $\mathbf{Z} \triangleq \left(\mathbf{U\Sigma U}^T\right)^{-1}\left\{(\mathbf{\Psi H})^T\mathbf{Q\Psi} + \sum_i (\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i\mathbf{X}_i\right\}\hat{\mathbf{H}}$ *and* $\|\mathbf{Z}\|_2 \triangleq \alpha < 1$. *Therefore, the following derivation can be performed similar to those found in [8, 69],*

$$\begin{aligned}
&\max_{\Delta}\left\|\mathbf{H}\Big\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H}) + \textstyle\sum_i (\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H}) + \mathbf{S} + \mathbf{R}\Big\}^{-1}\Big\{\mathbf{R} - \Big((\mathbf{\Psi H})^T\mathbf{Q\Psi} + \textstyle\sum_i (\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i\mathbf{X}_i\Big)\hat{\mathbf{H}}\mathbf{\Delta}\Big\}\mathbf{H}^{-1}\right\|_2 \\
&\quad \triangleq \left\|\mathbf{H}\Big\{\mathbf{U\Sigma U}^T + r\mathbf{I}\Big\}^{-1}\Big\{r\mathbf{I} + \Big((\mathbf{\Psi H})^T\mathbf{Q\Psi} + \textstyle\sum_i (\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i\mathbf{X}_i\Big)\hat{\mathbf{H}}\mathbf{\Delta}\Big\}\mathbf{H}^{-1}\right\|_2 \\
&\quad \triangleq \left\|\mathbf{H}\Big\{\mathbf{U\Sigma U}^T + r\mathbf{I}\Big\}^{-1}\Big\{r\mathbf{I} + \mathbf{U\Sigma U}^T\mathbf{Z\Delta}\Big\}\mathbf{H}^{-1}\right\|_2 \\
&\quad \triangleq \left\|\mathbf{H}\Big\{\mathbf{\Sigma} + r\mathbf{I}\Big\}^{-1}\Big\{r\mathbf{U}^T + \mathbf{\Sigma U}^T\mathbf{Z\Delta}\Big\}\mathbf{H}^{-1}\right\|_2 \\
&\quad \leq \Big\|\mathbf{H}\Big\|_2\Big\|\Big\{\mathbf{\Sigma} + r\mathbf{I}\Big\}^{-1}\Big\{r\mathbf{U}^T + \mathbf{\Sigma U}^T\mathbf{Z\Delta}\Big\}\Big\|_2\Big\|\mathbf{H}\Big\|_2^{-1} \\
&\quad \leq \|\mathbf{H}\|_2\|(\mathbf{\Sigma} + r\mathbf{I})^{-1}(r\mathbf{I} + \alpha\mathbf{\Sigma})\|_2\|\mathbf{H}\|_2^{-1} \\
&\quad = \max_i \tfrac{\alpha\sigma_i + r}{\sigma_i + r}\tfrac{\bar{\sigma}(\mathbf{H})}{\underline{\sigma}(\mathbf{H})} < 1, \forall r \in \mathbf{R} \geq 0.
\end{aligned}$$

□

From Lemma IV.3, we can conclude that the weighting matrix $\mathbf{R} = r\mathbf{I}$ does not influence the robust convergence properties of the ILC controlled system. Because $\mathbf{Q}$ and $\mathbf{W}_i$ weight the performance objectives, $\mathbf{S}$ should be designed such that the robust convergence condition in Lemma IV.2 holds. Similar statements and conclusions have been provided in [69] and [8].

Similarly, the requirement for robust convergence of systems with additional per-

formance metrics defined under classification III is $\gamma < 1$ where,

$$\gamma = \max_{\boldsymbol{\Delta}} \Big\| \mathbf{H} \Big\{ (\boldsymbol{\Psi}\mathbf{H})^T \mathbf{Q} (\boldsymbol{\Psi}\mathbf{H}) + \sum_i \mathbf{T}_i^T \mathbf{W}_i \mathbf{T}_i + \mathbf{S} + \mathbf{R} \Big\}^{-1} \Big\{ \mathbf{R} - \Big((\boldsymbol{\Psi}\mathbf{H})^T \mathbf{Q} \boldsymbol{\Psi} + \sum_i \mathbf{T}_i^T \mathbf{W}_i (\mathbf{T}_i \mathbf{H}^{-1}) \Big) \hat{\mathbf{H}} \boldsymbol{\Delta} \Big\} \mathbf{H}^{-1} \Big\|_2. \tag{4.19}$$

A direct extension of Lemmas IV.2 and IV.3 using (4.19) concludes that the weighting matrix $\mathbf{R}$ does not influence robust convergence for these class of systems. Therefore, $\mathbf{S}$ should be designed to mitigate the effects of model uncertainty.

4.3.3 Performance analysis

To investigate converged performance, we split the discussion into two segments.

4.3.3.1 Classification I

For complementary performance metrics, we focus the performance analysis on the full error signal. Given an update law (4.6) and learning filters (4.8)-(4.11) that satisfy Theorem IV.1, the norm of the bounded converged error is given as,

$$\lim_{j\to\infty} \Big\| \mathbf{e}_j \Big\|_2 = \Big\| \mathbf{e}_\infty \Big\|_2 = \Big\| \mathbf{g}_d - \mathbf{H}\mathbf{u}_\infty \Big\|_2 \leq \frac{\bar{\sigma}(\mathbf{H})}{\underline{\sigma}(\mathbf{H})} \frac{s\bar{\sigma}(\mathbf{g}_d)}{\underline{\sigma}\Big(q(\boldsymbol{\Psi}\mathbf{H})^T (\boldsymbol{\Psi}\mathbf{H}) + \sum_i w_i (\mathbf{X}_i\mathbf{H})^T (\mathbf{X}_i\mathbf{H}) \Big) + s}.$$

For practical use, q and w_i should be set to maximize system performance, while s should be designed to satisfy Theorem IV.1 to ensure robustness in the presence of model uncertainty.

4.3.3.2 Classification II and III

For systems that include competing objectives, regardless of the specific domain, we consider two performance metrics, the error signal from the primary objective $\boldsymbol{\Psi}\mathbf{e}$,

and the error signals from the additional performance metrics $\boldsymbol{\varepsilon}_i$.

Given an update law (4.6) and learning filters (4.8)-(4.11) that satisfy Theorem IV.1, the norm of the converged primary error signal is,

$$\begin{aligned}\Big\|\boldsymbol{\Psi}\mathbf{e}_\infty\Big\|_2 &= \Big\|\boldsymbol{\Psi}\left(\mathbf{g}_d - \mathbf{H}\mathbf{u}_\infty\right)\Big\|_2 \\ &\leq \frac{\bar{\sigma}(\boldsymbol{\Psi}\mathbf{H})}{\underline{\sigma}(\boldsymbol{\Psi}\mathbf{H})}\frac{\bar{\sigma}\Big(\sum_i w_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\Big)+s}{\underline{\sigma}\Big(q(\boldsymbol{\Psi}\mathbf{H})^T(\boldsymbol{\Psi}\mathbf{H})+\sum_i w_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\Big)+s}\bar{\sigma}(\boldsymbol{\Psi}\mathbf{g}_d) \\ &\quad+\frac{\bar{\sigma}(\boldsymbol{\Psi}\mathbf{H})\bar{\sigma}\Big(\sum_i w_i(\mathbf{X}_i\mathbf{H})^T\Big)}{\underline{\sigma}\Big(q(\boldsymbol{\Psi}\mathbf{H})^T(\boldsymbol{\Psi}\mathbf{H})+\sum_i w_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\Big)+s}\bar{\sigma}(\mathbf{z}_{i,d}). \end{aligned} \tag{4.20}$$

Assuming small values for $\mathbf{z}_{i,d}$, (4.20) can be simplified to,

$$\Big\|\boldsymbol{\Psi}\mathbf{e}_\infty\Big\|_2 \leq C_{e,2}\frac{f(w_i)+s}{f(q,w_i)+s}\bar{\sigma}(\boldsymbol{\Psi}\mathbf{g}_d)+\boldsymbol{\epsilon}. \tag{4.21}$$

where $C_{e,2}=\frac{\bar{\sigma}(\boldsymbol{\Psi}\mathbf{H})}{\underline{\sigma}(\boldsymbol{\Psi}\mathbf{H})}$.

For comparison, we also consider the converged errors from the additional performance metrics,

$$\begin{aligned}\Big\|\boldsymbol{\varepsilon}_{i,\infty}\Big\|_2 &= \Big\|\left(\mathbf{z}_{i,d} - \mathbf{X}_i\mathbf{H}\mathbf{u}_\infty\right)\Big\|_2 \\ &\leq \frac{\bar{\sigma}(\mathbf{X}_i\mathbf{H})}{\underline{\sigma}(\mathbf{X}_i\mathbf{H})}\frac{q\bar{\sigma}(\boldsymbol{\Psi}\mathbf{H})^2+s}{\underline{\sigma}\Big(q(\boldsymbol{\Psi}\mathbf{H})^T(\boldsymbol{\Psi}\mathbf{H})+\sum_i w_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\Big)+s}\bar{\sigma}(\mathbf{z}_{i,d}) \\ &\quad+\frac{q\bar{\sigma}(\boldsymbol{\Psi}\mathbf{H})\bar{\sigma}(\mathbf{X}_i\mathbf{H})}{\underline{\sigma}\Big(q(\boldsymbol{\Psi}\mathbf{H})^T(\boldsymbol{\Psi}\mathbf{H})+\sum_i w_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\Big)+s}\bar{\sigma}(\boldsymbol{\Psi}\mathbf{g}_d) \\ &\quad+\frac{\bar{\sigma}(\mathbf{X}_i\mathbf{H})\bar{\sigma}\Big(\sum_k w_k(\mathbf{X}_k\mathbf{H})^T\Big)}{\underline{\sigma}\Big(q(\boldsymbol{\Psi}\mathbf{H})^T(\boldsymbol{\Psi}\mathbf{H})+\sum_i w_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\Big)+s}\bar{\sigma}(\mathbf{z}_{k,d}), \end{aligned} \tag{4.22}$$

where $i=1,2,\ldots P$, $k=1,2,\ldots,i-1,i+1,i+2,\ldots P$, and P is the number of the additional performance metrics. Assuming small values for all $\mathbf{z}_{i,d}$ and $\mathbf{z}_{k,d}$, (4.22)

can be simplified to,

$$\left\|\boldsymbol{\varepsilon}_{i,\infty}\right\|_2 \leq \frac{f(q)}{f(q,w_i)+s}\bar{\sigma}(\boldsymbol{\Psi}\mathbf{g}_d) + \boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_2. \tag{4.23}$$

With some small modifications as a result of replacing $\mathbf{X}$ with $\mathbf{T}$, similar relationships to those defined in (4.20) - (4.23) will also hold for classification III.

As can be seen from (4.21) and (4.23), converged performance results in a direct trade-off between the primary and additional metrics. Increasing the emphasis on the primary metric q results in a decrease in the converged error signal from (4.21) and an increase in (4.23). The opposite occurs when emphasizing the additional performance metrics w_i.

Similar results can be obtained for the domain specific performance analysis. Please see Appendix B for full derivation of the converged error signals.

4.3.4 Tuning guidelines

Based on the analysis in 4.3.1-4.3.3, and assuming diagonal real-valued matrices $[\,\mathbf{Q}, \mathbf{S}, \mathbf{R}, \mathbf{W}_i\,] = [\,q\mathbf{I}, s\mathbf{I}, r\mathbf{I}, w_i\mathbf{I}\,]$, we provide the following design tuning guidelines for norm-optimal multi-objective ILC (illustrated in Fig. 4.2):

- s1 $\mathbf{Q}$ *design*: Determine the q gain to meet the primary performance requirements. Generally, $q = 1$ for uniform weighting of the error.
- s2 $\mathbf{S}$ *design*: Design s to ensure the system is robust in the presence of model uncertainty. Start with $s \approx 0.01\,\|\mathbf{H}\|_2$. Ensure that the system is stable.
- s3 $\mathbf{W}_i$ *design*: Start with $w_i = 0$ and increase the w_i gain until the performance of the i_{th} objective meets the desired performance requirements. If there exists a trade-off between the primary and additional performance objectives, determine w_i to meet this trade-off.
- s4 $\mathbf{S}$ *design*: Subsequently reduce s until the system diverges. Select the gain as

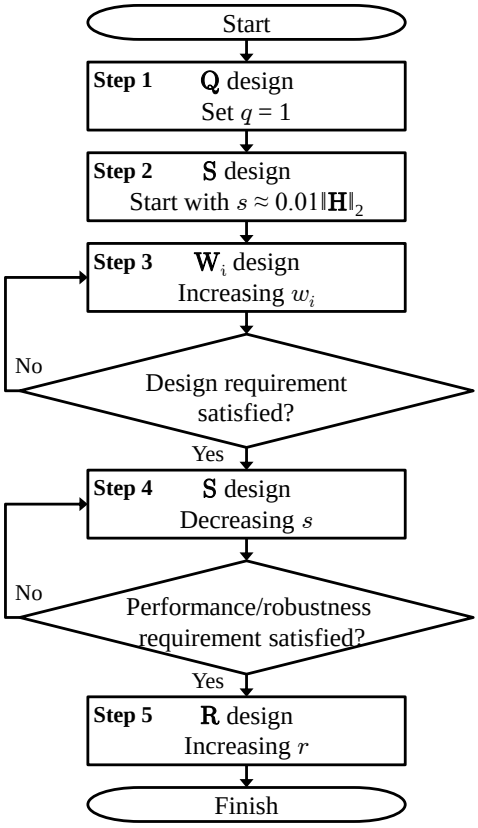


Figure 4.2: Flowchart for design guidelines

$s = 2 \cdot s_{min}$ to provide a safety factor.

s5 $\mathbf{R}$ *design*: Start with $r = 0$ and increase r until the steady-state error fluctuations are within the desired bounds.

4.4 Simulation Set-up and Demonstration

4.4.1 Additive Manufacturing System: Electrohydrodynamic Jet Printing

To validate the performance capabilities of the proposed framework, we consider a micro-additive manufacturing process known as electrohydrodynamic jet (e-jet) printing [70, 71, 72], Fig. 4.3. E-jet printing has demonstrated superior resolution (sub $0.1 - 10\mu m$) as compared to ink-jet printing ($> 30\mu m$), printing micron and

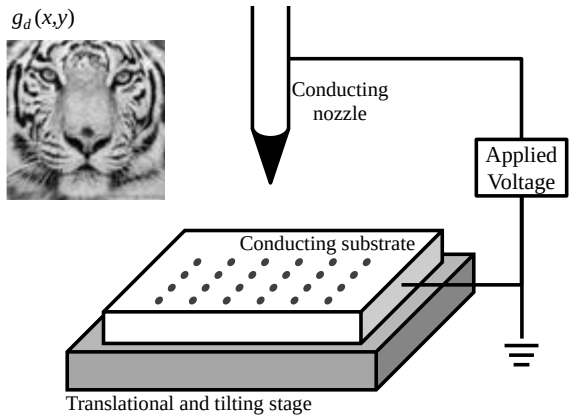


Figure 4.3: Schematic diagram of an e-jet system and reference image (64×64 grayscale image of a tiger where the eyes are selected as the primary points of interest).

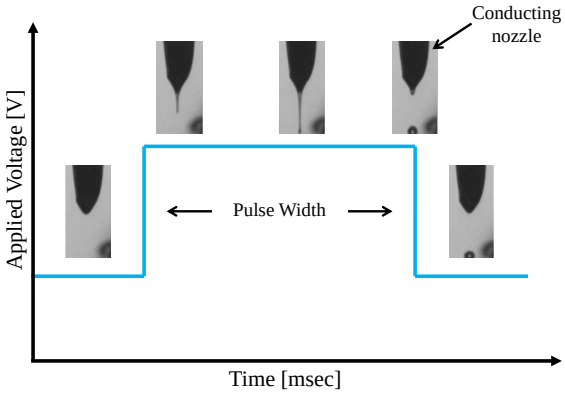


Figure 4.4: Temporal dynamics of e-jet printing: the material deposition is driven by the applied voltage and the width of a pulse width.

submicron scale droplets using a wide variety of inks such as organic and inorganic materials, and suspensions of solids (e.g. silver nanoparticles and DNA) [73, 74, 75].

Figure 4.3 also shows the reference image used in these demonstrative simulations. The image is a 64×64 gray-scale image of a tiger. Note that white area (a value of 1) denotes a maximum height of material, while the black area (a value of 0) represents locations with zero material deposition.

The dynamics of E-jet printing can be split into the temporal dynamics that drive the ejection process, and the spatial dynamics that govern the patterning of the

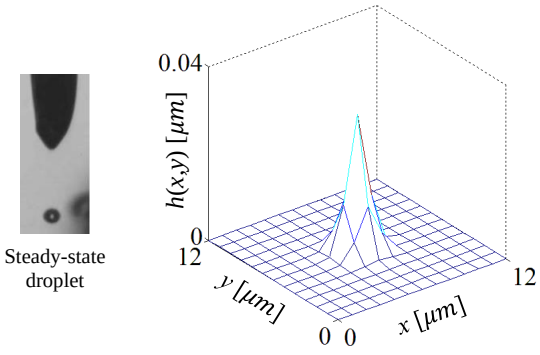


Figure 4.5: Spatial dynamics of e-jet printing: the spatial impulse response of the droplet is represented by a 2D Gaussian distribution.

material on the substrate surface. Material ejection is driven by an applied voltage, with the volume of material determined by the width of a pulsed input signal, Fig. 4.4.

Steady-state behavior of the ejected material is represented in the spatial pattern that is deposited on the substrate. The spatial pattern is defined through the spreading of a droplet at a given point in space. This behavior is represented as a truncated 2D Gaussian distribution, Fig. 4.5. The distribution of the material is mathematically represented by the spatial impulse function, $h(x, y)$, given as [38, 42, 49],

$$h(x, y) = \alpha e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}}$$

where $\alpha = 0.031\mu\text{m msec}^{-\frac{1}{2}}$, $\sigma = 1.415\mu\text{m}$, and $h(x, y)$ is sampled at $x = 0, 1, \ldots, 12\mu\text{m}$ and $y = 0, 1, \ldots, 12\mu\text{m}$ [49].

4.4.2 Performance objectives

To demonstrate the proposed control framework on a model of this high-resolution additive manufacturing process, two critical performance objectives associated with this application must be evaluated: image quality and material use. To evaluate

image quality mathematically, the image quality index is borrowed from the field of image processing [76] and applied to additive manufacturing.

Definition 7. *Image quality index is the product of the brightness of a select group of pixels within the image that are of great importance (also termed focus), the brightness of the entire map, and the contrast between the selected locations and the surrounding pixels ([76]).*

$$\Phi_j = \underbrace{\left(\frac{\|\mathbf{\Psi}\mathbf{e}_j\|_2}{\|\mathbf{\Psi}\mathbf{e}_0\|_2}\right)}_{focus} \underbrace{\left(\frac{\|\boldsymbol{\varepsilon}_{b,j}\|_2}{\|\boldsymbol{\varepsilon}_{b,0}\|_2}\right)}_{map\ brightness} \underbrace{\left(\frac{\|\boldsymbol{\varepsilon}_{c,j}\|_2}{\|\boldsymbol{\varepsilon}_{c,0}\|_2}\right)}_{contrast}.$$

Image quality ranges from $[0, 1]$*, with* 0 *denoting high image quality and* 1 *denoting poor image quality.*

Definition 8. *Focus defines a specific selection of locations in the image that are of great importance, and therefore should be emphasized within the performance output.*

Definition 9. *Brightness is a measure of the output map as compared to the desired reference image. Perfect image brightness is denoted with the value* 0*, with the image degrading to an undesirable map for a value of one.*

Definition 10. *Contrast defines the interface between the selected locations of focus (interest) and the surrounding pixels or locations in the spatial map.*

We now separate image quality and material use into primary, complementary, and competing objectives.

4.4.2.1 Primary performance objective

The primary performance objective is defined as a group of selected locations that are of primary interest. This objective defines the **focus** of the image, and combines with the additional complementary objectives to achieve an overall performance quality index (see (4.24)-(4.26)).

4.4.2.2 Complementary performance objectives

The complementary performance objectives include overall image quality termed brightness, and the interface points between the selected points of interest and their respective boundary points. These interface points are termed contrast points. The primary and complementary performance metrics are grouped into a single objective labeled image quality. Borrowing from classical image processing [76] and the definition provided above, image quality can be mathematically represented as,

$$\textbf{focus: } \boldsymbol{\Psi}\mathbf{e}_j = \boldsymbol{\Psi}(\mathbf{g}_d - \mathbf{H}\mathbf{u}_j), \tag{4.24}$$

$$\textbf{contrast: } \boldsymbol{\varepsilon}_{c,j} = \mathbf{X}_c\mathbf{e}_j, \tag{4.25}$$

$$\text{with } \mathbf{X}_c = \mathbf{C}_s^{-1}(\mathbf{C}_s - \mathbf{C}_b) \text{ defined in (4.28)-(4.29)},$$

$$\textbf{brightness: } \boldsymbol{\varepsilon}_{b,j} = \mathbf{X}_b\mathbf{e}_j, \quad \text{with } \mathbf{X}_b = \mathbf{I}. \tag{4.26}$$

To illustrate the concept of contrast, Fig. 4.6 shows an example output map where the gray locations in the map represent the selected points of interest and the white blocks represent the boundary indices. As an example, consider the interface at $g_j(2,2)$. There exist 5 adjacent boundary points; $[g_j(1,1), g_j(1,2), g_j(1,3), g_j(2,1),$

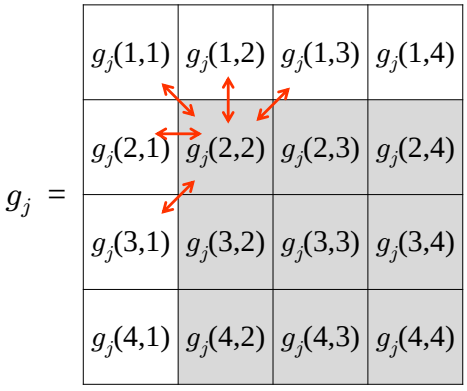


Figure 4.6: Output example to illustrate the **contrast** performance metric. Gray locations indicate focus points; white blocks define boundary indices.

$g_j(3,1)]$. Given these boundary points, the interface at $g_j(2,2)$ can be defined as,

$$Z_j(2,2) = \frac{1}{5}\Bigg\{ \sum_{i=1}^{3} \Big(g_j(2,2) - g_j(1,i)\Big) + \sum_{k=2}^{3} \Big(g_j(2,2) - g_j(k,1)\Big) \Bigg\}. \tag{4.27}$$

Generalizing (4.27) and vectorizing the resultant formula, the generalized contrast performance metric can be defined,

$$\mathbf{Z}_{c,j} = \mathbf{X}_c \mathbf{g}_j = \mathbf{C}_s^{-1} \left(\mathbf{C}_s - \mathbf{C}_b\right) \mathbf{g}_j. \tag{4.28}$$

In (4.28), the variable $\mathbf{C}_b$ defines the matrix of interface locations around a selected point and can be mathematically represented as,

$$\mathbf{C}_b = \begin{bmatrix} c_b(1,1) & \dots & c_b(1,MN) \\ \vdots & \ddots & \vdots \\ c_b(MN,1) & \dots & c_b(MN,MN) \end{bmatrix},$$

with $c_b(i,j)$ used to represent an interface location,

$$c_b(i,j) = \begin{cases} 1, & i = n_p, \\ & j = \textit{interface location around } n_p, \\ & p = 1, \dots, \bar{M} \\ 0, & \textit{otherwise}. \end{cases}$$

In (4.28), $\mathbf{C}_s$ is a diagonal matrix denoting the number of interface locations,

$$\mathbf{C}_s = \begin{bmatrix} c_s(1,1) & & 0 \\ & \ddots & \\ 0 & & c_s(MN,MN) \end{bmatrix}, \tag{4.29}$$

where c_s sums the number of boundary points around a point of interest n_p,

$$c_s(i,i) = \begin{cases} \sum_{j=1}^{MN} c_b(i,j), & i = n_p, \quad p = 1,\ldots,\bar{M} \\ 0, & otherwise. \end{cases}$$

Please see Appendix C for examples of $\mathbf{C}_b$ and $\mathbf{C}_s$ matrices based on Fig. 4.6.

4.4.2.3 Competing performance objectives

Although 3D printing provides material savings over many other fabrication techniques (e.g. lithography, milling), there are applications in bio-printing in which the materials are extremely expensive and therefore usage should be minimized. In such instances, there is a design trade-off between device performance (e.g. signal resolution in a bio-sensor) and material use.

Given the phase changes (e.g. evaporation) and other means of material loss that occur post printing; the true amount of material utilized during the fabrication process is defined with respect to the input signal, rather than the output signal.

$$\begin{aligned} \textbf{material use: } \boldsymbol{\varepsilon}_{m,j} &= \mathbf{m}_d - \mathbf{m}_j \\ &= \mathbf{m}_d - \mathbf{T}_m \mathbf{u}_j \\ &= \mathbf{m}_d - \begin{bmatrix} v & & 0 \\ & \ddots & \\ 0 & & v \end{bmatrix} \mathbf{u}_j \end{aligned} \tag{4.46}$$

In (4.46), $\boldsymbol{\varepsilon}_{m,j}$ defines the error signal for material use, where $\mathbf{m}_d$ and $\mathbf{m}_j$ denote

desired and measured material usage. The amount of material released per millisecond pulse width is defined as $v = 0.1\mu\text{m}^3/\text{msec}$ based on experimental testing [77].

4.4.2.4 Cost function

Using the performance objectives defined in (4.24)-(4.26) and (4.46), a multi-objective cost function can be determined.

$$J = \mathbf{u}_{j+1}^T \mathbf{S}\mathbf{u}_{j+1} \; + \; (\mathbf{u}_{j+1} - \mathbf{u}_j)^T \mathbf{R}(\mathbf{u}_{j+1} - \mathbf{u}_j) \; + \; \underbrace{\underbrace{\boldsymbol{\varepsilon}_{m,j+1}^T \mathbf{W}_m \boldsymbol{\varepsilon}_{m,j+1}}_{\text{Classification III: material use}}}_{\text{Competing Objective}}$$
$$+ \; \underbrace{\underbrace{(\boldsymbol{\Psi}\mathbf{e}_{j+1})^T \mathbf{Q}(\boldsymbol{\Psi}\mathbf{e}_{j+1})}_{\text{Primary: focus}} \; + \; \underbrace{\boldsymbol{\varepsilon}_{c,j+1}^T \mathbf{W}_c \boldsymbol{\varepsilon}_{c,j+1}}_{\text{Classification I: contrast}} \; + \; \underbrace{\boldsymbol{\varepsilon}_{b,j+1}^T \mathbf{W}_b \boldsymbol{\varepsilon}_{b,j+1}}_{\text{Classification I: brightness}}}_{\text{Primary + Complementary Objectives}} \tag{4.47}$$

In (4.47), $\mathbf{Q}$, $\mathbf{W}_c$, $\mathbf{W}_b$, and $\mathbf{W}_m$ are the weighting matrices on the error signals for focus, contrast, brightness and material usage.

4.4.3 Simulation Setup

To provide a more realistic simulation, we consider a system with model uncertainty and noise. In [49], the quantified deviation of the spatial impulse function, $\hat{g}(x, y, u)$ is provided. This is experimentally generated based on AFM data. $\hat{g}(x, y, u)$ depends on the input signal and defines nonlinear, unmodeled dynamics that lead to model uncertainty in the system. External disturbances are modeled as white noise, $\eta_j(x, y) = \mathcal{N}(0, 0.0125)$ μm for all j. Given these definitions, the plant dynamics of the simulated e-jet printing system are defined as,

$$\mathbf{g}_j = \mathbf{H}\mathbf{u}_j + \hat{\mathbf{g}}(\mathbf{u}_j) + \boldsymbol{\eta}_j,$$

where $\hat{\mathbf{g}}(\mathbf{u}_j)$ and $\boldsymbol{\eta}_j$ are vectorized forms of $\hat{g}(x, y, u)$ and $\eta_j(x, y)$, respectively.

Norm optimal ILC controllers were designed to satisfy the condition for monotonic convergence. Weighting gains and convergence rates for 4 cases are provided in Table 4.1. To illustrate the effects of design on convergence rate, different controllers were designed using the methodology presented in subsection 4.3.4 to ensure stable systems with consistent convergence rates.

4.4.4 Simulation results

To validate the effect of different weighting gains on image quality and material usage, simulation tests were conducted for the four cases provided in Table 4.1. Figures 4.7 and 4.8 illustrate the converged values of the image quality and material use as a function of the different cases. Case 1 represents the baseline performance with an emphasis on the primary objective; a focus on the tiger's eyes, Fig. 4.9(a). Subsequent cases provide a comparison between the baseline case and the impact of emphasizing brightness Fig. 4.9(b), contrast Fig. 4.9(c), and material use Fig. 4.9(d).

Note from Table 4.1 that the criteria for convergence ($\gamma < 1$) was satisfied for all cases. In order to achieve consistent convergence results (i.e. the same γ value), the r gain was increased in cases 2 and 4. This follows the relationship denoted in Fig. 4.1. Case 3 did not require a substantial modification to the r gain, which may be attributed to the smaller changes observed in this case as compared to case 1 as illustrated in Figs. 4.7 and 4.8.

As can be seen from Fig. 4.7, increasing the w_i weightings for brightness (case 2) and contrast (case 3) result in an increase in the image quality, while an emphasis

Table 4.1: Weighting gains and convergence rates

Case	q	s	r	w_b	w_c	w_m	γ
1	1	$1e^{-4}$	$1.00e^{-3}$	0	0	0	0.909
2	1	$1e^{-4}$	$1.08e^{-3}$	1	0	0	0.909
3	1	$1e^{-4}$	$1.00e^{-3}$	0	1	0	0.909
4	1	$1e^{-4}$	$1.01e^{-1}$	0	0	1	0.909

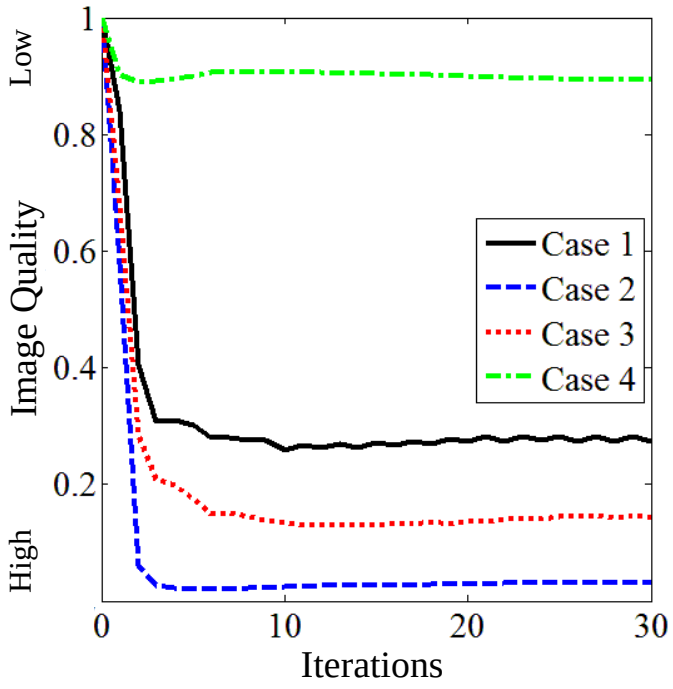


Figure 4.7: Converged results of the image quality for the four weighting matrix designs.

on material use (case 4) decreases the image quality. The reverse can been seen from the perspective of material use shown in Fig. 4.8; the largest *material savings* were observed when weighting material use (case 4), while the most *material use* stemmed from an emphasis on image brightness (case 2). These results follow the correlations described in equations (4.21) and (4.23), and are further verified in Fig. 4.9.

The output images after 30 iterations can be seen in Fig. 4.9. Figure 4.9(a) provides a baseline image where the additional performance metrics are not weighted: $[w_b, w_c, w_m] = [0, 0, 0]$. As shown in Fig. 4.9(b), increasing the w_b gain for brightness improves the image quality over the entire image at the expense of material use. Although increasing the w_c contrast gain improves the image quality around the selected locations (see Fig. 4.9(c)), the boundary change is subtle and does not require as significant an increase in material use as the brightness metric. Figure 4.9(d) (increase in w_m metric) clearly illustrates the trade-off between image quality

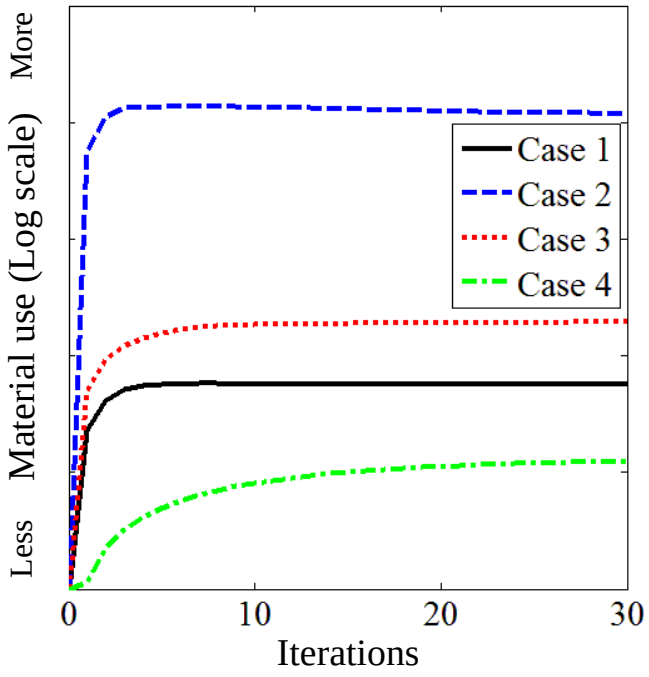
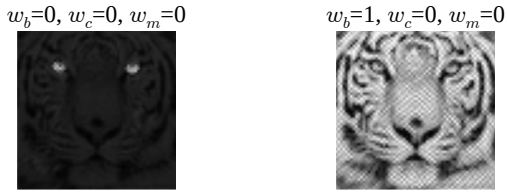
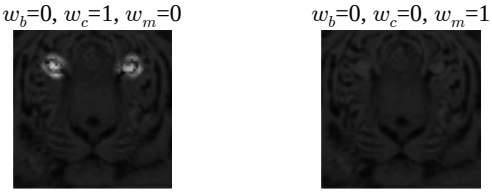


Figure 4.8: Converged results of the material use for the four weighting matrix designs.



(a) Baseline (b) Brightness



(c) Contrast (d) Material use

Figure 4.9: Simulation results as a function of objective function weightings.

and material use.

4.5 Concluding Remarks

This chapter presents multi-objective spatial ILC as an enabling tool for evaluating the performance advantages of multiple performance objectives in applications that require spatial closeness. Independent from time, these performance objectives are defined along a 2D spatial map with respect to (x, y) and are used for design and performance optimization. The multi-objective spatial ILC framework is analogous to the multi-objective temporal ILC framework, but there exists key modifications which stem from the 2D formulation of the performance objectives in the spatial domain. The specific contributions of this work include: (1) the extension of the multi-objective learning framework from the time domain to the spatial domain, (2) the development of a more formal and generalized multi-objective ILC framework, (3) new stability and convergence analyses based on the generalized framework, as well as an updated design methodology, and (4) a simulation validation of the new control approach on a model of a high-resolution additive manufacturing system.

To demonstrate the proposed framework, we considered a micro-additive manufacturing process, electrohydrodynamic jet (e-jet) printing, and the fabrication of an image which focused on some critical locations. Image quality and material usage were used to compare performance trade-offs for design and performance optimization. Simulation results with varying weighting gains illustrated the advantages of the proposed spatial framework for optimizing multiple performance metrics within a single framework. By addressing multiple performance objectives, a more customized fabrication process is possible. Through this customization, the control designer can more readily observe the trade-off benefits associated with varying performance metrics to enable an informed control design for enhanced fabrication.

The ability to personalize the control design to specific cases is particularly important for additive manufacturing; a manufacturing paradigm enabling new construct designs and less material waste than traditional manufacturing methods. To fully

explore the benefits of this manufacturing paradigm, the control architecture must provide the necessary design flexibility and customization to enable the fabrication process. The multi-objective learning controller presented in this chapter provides a promising option to address this need.

CHAPTER V

Design and Analysis for a Centralized Weighting Matrix

In this chapter, we provide a single weighting matrix design for multiple performance objectives as an extension to the multi-objective iterative learning control framework presented in Chapters III and IV. While the initial format provided in Chapter IV presents a distributed design approach (see Eqs. (4.8)-(4.11)), weighting of the primary and complementary performance metrics can be captured within a single weighting matrix. To illustrate this centralized approach, we present the relevant analysis and a numerical example.

5.1 Single Weighting Matrix Design and Analysis

Recall that the generalized cost function for the multi-objective iterative learning controller from (3.6) can be given as,

$$J=(\mathbf{\Psi e}_{j+1})^T\mathbf{Q}(\mathbf{\Psi e}_{j+1})+\mathbf{u}_{j+1}^T\mathbf{S}\mathbf{u}_{j+1}+(\mathbf{\Delta u}_{j+1})^T\mathbf{R}(\mathbf{\Delta u}_{j+1})+\sum_i \boldsymbol{\varepsilon}_{i,j+1}^T\mathbf{W}_i\boldsymbol{\varepsilon}_{i,j+1},$$

where $\mathbf{Q}=q\mathbf{I}$, $\mathbf{S}=s\mathbf{I}$, $\mathbf{R}=r\mathbf{I}$ and $\mathbf{W}_i=w_i\mathbf{I}$ are positive-valued diagonal matrices. Given the structure of the lifted matrix formulation, we can combine the primary and classification I performance metrics into a single $\mathbf{Q}$ weighting matrix design. Classifi-

cations II and III can be combined with the weighting on the control input as a single $\mathbf{S}$ matrix. For brevity we will only present the full derivation and simple example for the single $\mathbf{Q}$ weighting matrix design. Similar approaches and assumptions can be applied to the single $\mathbf{S}$ weighting matrix design, with a few important notes provide in subsection 5.2.

5.1.1 Single $\bar{\mathbf{Q}}$ Design

To design for a single weighting gain applied to the error signal, the following assumptions must be satisfied:

A1 The additional performance metrics must be classified as complementary metrics (classification I), in which the error signal can be defined as a function of the original error signal, $\boldsymbol{\varepsilon}_{i,j+1} = \mathbf{X}_i \mathbf{e}_{j+1}$.

A2 The weighting gains q and w_i must be directly proportional to the centralized gain $\bar{q}$.

Assumption 1 ensures that the performance metrics can be written as an equivalent centralized matrix that emphasized the error signal. Assumption 2 is not a necessary requirement for combining classification I and the primary performance metric into a single weighting matrix. If this assumption is not satisfied, then the combined matrix directly follows the discretized design and there are not potential advantages to this design architecture. If assumption 2 is satisfied, then the proportional gains represent the relative importance of each metric and are used to derive a common centralized gain $\bar{q}$.

Applying these assumptions, the cost function (3.6) can be rewritten as,

$$\begin{aligned} J &= (\boldsymbol{\Psi}\mathbf{e}_{j+1})^T \mathbf{Q} (\boldsymbol{\Psi}\mathbf{e}_{j+1}) + \mathbf{u}_{j+1}^T \mathbf{S}\mathbf{u}_{j+1} + (\boldsymbol{\Delta}\mathbf{u}_{j+1})^T \mathbf{R} (\boldsymbol{\Delta}\mathbf{u}_{j+1}) + \sum_i \boldsymbol{\varepsilon}_{i,j+1}^T \mathbf{W}_i \boldsymbol{\varepsilon}_{i,j+1} \\ &= \mathbf{e}_{j+1}^T \bar{\mathbf{Q}} \mathbf{e}_{j+1} + \mathbf{u}_{j+1}^T \mathbf{S}\mathbf{u}_{j+1} + (\boldsymbol{\Delta}\mathbf{u}_{j+1})^T \mathbf{R} (\boldsymbol{\Delta}\mathbf{u}_{j+1}), \end{aligned} \tag{5.1}$$

with

$$\begin{aligned}\bar{\mathbf{Q}} &= \mathbf{\Psi}^T\mathbf{Q}\mathbf{\Psi} + \sum_i \mathbf{X}_i^T\mathbf{W}_i\mathbf{X}_i \\ &= q\mathbf{\Psi}^T\mathbf{\Psi} + \sum_i w_i\mathbf{X}_i^T\mathbf{X}_i \\ &= \bar{q}(\frac{q}{\bar{q}}\mathbf{\Psi}^T\mathbf{\Psi}) + \bar{q}(\sum_i \frac{w_i}{\bar{q}}\mathbf{X}_i^T\mathbf{X}_i) \\ &= \bar{q}(\alpha\mathbf{\Psi}^T\mathbf{\Psi} + \sum_i \beta_i\mathbf{X}_i^T\mathbf{X}_i).\end{aligned} \tag{5.2}$$

In (5.2), $q = \alpha\bar{q}$ and $w_i = \beta_i\bar{q}$, where $\{\alpha, \beta_i\}$ are gains that represent the relative importance of the metrics, and satisfy the condition $\alpha + \sum_i \beta_i = 1$. If $\{\alpha, \beta_i\}$ are constant, than only one gain $\bar{q}$ may be tuned to achieve the desired performance results. If the performance does not meet the design requirements, $\{\alpha, \beta_i\}$ must be revised and the tuning process repeated. This may lead to several design iterations while the *optimal* gains are found. This two-step tuning process is more time consuming than in the discrete case, and may provide a less direct method for understanding the impact of weighting the additional performance metrics on the overall system performance.

Taking the derivative of (5.1) with respect to $\mathbf{u}_{j+1}$ results in the following control update law,

$$\mathbf{u}_{j+1} = \mathbf{L}_u\mathbf{u}_j + \mathbf{L}_e\mathbf{e}_j \tag{5.3}$$

with

$$\mathbf{L}_u = \left\{\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H} + \mathbf{S} + \mathbf{R}\right\}^{-1}\left\{\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H} + \mathbf{R}\right\} \tag{5.4}$$

$$\mathbf{L}_e = \left\{\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H} + \mathbf{S} + \mathbf{R}\right\}^{-1}\mathbf{H}^T\bar{\mathbf{Q}}. \tag{5.5}$$

Utilizing the same process presented in Chapter IV, monotonic convergence of the system error,

$$\Big\|\mathbf{e}_\infty - \mathbf{e}_{j+1}\Big\|_2 \leq \gamma \Big\|\mathbf{e}_\infty - \mathbf{e}_j\Big\|_2,$$

is guaranteed if and only if the convergence rate satisfies $\gamma < 1$, where γ is now defined in terms of $\bar{q}$, α and β_i.

$$\begin{aligned}
\gamma &\triangleq \left\|\mathbf{H}\Big\{\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H} + \mathbf{S} + \mathbf{R}\Big\}^{-1}\mathbf{R}\mathbf{H}^{-1}\right\|_2 \\
&\leq \frac{\bar{\sigma}(\mathbf{H})}{\underline{\sigma}(\mathbf{H})}\frac{r}{\underline{\sigma}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H}) + s + r} \\
&\triangleq \frac{\bar{\sigma}(\mathbf{H})}{\underline{\sigma}(\mathbf{H})}\frac{r}{\underline{\sigma}\Big(\bar{q}\big\{\alpha(\mathbf{\Psi}\mathbf{H})^T(\mathbf{\Psi}\mathbf{H}) + \sum_i \beta_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\big\}\Big) + s + r}.
\end{aligned}$$

To investigate converged performance with a single weighting gain design, the norm of the bounded converged error is given as,

$$\begin{aligned}
\lim_{j\to\infty}\Big\|\mathbf{e}_j\Big\|_2 = \Big\|\mathbf{e}_\infty\Big\|_2 &= \Big\|\mathbf{g}_d - \mathbf{H}\mathbf{u}_\infty\Big\|_2 \\
&\leq \frac{\bar{\sigma}(\mathbf{H})}{\underline{\sigma}(\mathbf{H})}\frac{s\bar{\sigma}(\mathbf{g}_d)}{\underline{\sigma}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H}) + s} \\
&\triangleq \frac{\bar{\sigma}(\mathbf{H})}{\underline{\sigma}(\mathbf{H})}\frac{s\bar{\sigma}(\mathbf{g}_d)}{\underline{\sigma}\Big(\bar{q}\big\{\alpha(\mathbf{\Psi}\mathbf{H})^T(\mathbf{\Psi}\mathbf{H}) + \sum_i \beta_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\big\}\Big) + s}.
\end{aligned}$$

For practical use, $\bar{q}$ should be set to maximize system performance, while s must be designed to ensure robustness in the presence of model uncertainty. Similar to conventional ILC, the designs of $\bar{q}$ and s follow the standard performance/robustness trade-off.

5.1.2 Simple example

To provide a simple demonstrative example, we use a modified cost function for the additive manufacturing system (4.47) presented in Chapter. IV. In this example,

we disregard material usage and focus on the image quality applied to the output map Fig. 5.1.

$$
\begin{aligned}
J =& \mathbf{u}_{j+1}^T \mathbf{S}\mathbf{u}_{j+1} + (\boldsymbol{\Delta}\mathbf{u}_{j+1})^T \mathbf{R}(\boldsymbol{\Delta}\mathbf{u}_{j+1}) \\
&+ \underbrace{(\boldsymbol{\Psi}\mathbf{e}_{j+1})^T \mathbf{Q}(\boldsymbol{\Psi}\mathbf{e}_{j+1})}_{\text{Primary: focus}} + \underbrace{\boldsymbol{\varepsilon}_{c,j+1}^T \mathbf{W}_c \boldsymbol{\varepsilon}_{c,j+1}}_{\text{Classification I: contrast}} + \underbrace{\boldsymbol{\varepsilon}_{b,j+1}^T \mathbf{W}_b \boldsymbol{\varepsilon}_{b,j+1}}_{\text{Classification I: brightness}}
\end{aligned} \tag{5.6}
$$

The cost function (5.6) can be rearranged to,

$$
J = \mathbf{e}_{j+1}^T \bar{\mathbf{Q}} \mathbf{e}_{j+1} + \mathbf{u}_{j+1}^T \mathbf{S}\mathbf{u}_{j+1} + (\boldsymbol{\Delta}\mathbf{u}_{j+1})^T \mathbf{R}(\boldsymbol{\Delta}\mathbf{u}_{j+1}),
$$

where $\bar{\mathbf{Q}} = \boldsymbol{\Psi}^T \mathbf{Q} \boldsymbol{\Psi} + \mathbf{X}_c^T \mathbf{W}_c \mathbf{X}_c + \mathbf{X}_b^T \mathbf{W}_b \mathbf{X}_b = q\boldsymbol{\Psi}^T \boldsymbol{\Psi} + w_c \mathbf{X}_c^T \mathbf{X}_c + w_b \mathbf{X}_b^T \mathbf{X}_b$.

Applying the definitions provided in subsection 4.4.2.2 to the map illustrated in Fig. 5.1, $\boldsymbol{\Psi}$, $\mathbf{X}_c$, and $\mathbf{X}_b$ can be written as,

$$
\boldsymbol{\Psi} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}, \quad
\mathbf{X}_c = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.2 & -0.2 & -0.2 & -0.2 & 1 & 0 & -0.2 & 0 & 0 \\
0 & -0.5 & -0.5 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.5 & 0 & 0 & -0.5 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}, \quad
\mathbf{X}_b = \mathbf{I}
$$

$$
g_j = \begin{array}{|c|c|c|}
\hline
g_j(1,1) & g_j(1,2) & g_j(1,3) \\
\hline
g_j(2,1) & g_j(2,2) & g_j(2,3) \\
\hline
g_j(3,1) & g_j(3,2) & g_j(3,3) \\
\hline
\end{array}
$$

Figure 5.1: Output example where gray locations denote the selected points and the white blocks indicate contrast locations.

From these definitions, $\bar{\mathbf{Q}} = q\mathbf{\Psi}^T\mathbf{\Psi} + w_c\mathbf{X}_c^T\mathbf{X}_c + w_b\mathbf{X}_b^T\mathbf{X}_b$ is obtained as,

$$\bar{\mathbf{Q}} = \begin{bmatrix} 0.04w_c + w_b & 0.04w_c & 0.04w_c & 0.04w_c & -0.2w_c & 0 & 0.04w_c & 0 & 0 \\ 0.04w_c & 0.29w_c + w_b & 0.29w_c & 0.04w_c & -0.2w_c & -0.5w_c & 0.04w_c & 0 & 0 \\ 0.04w_c & 0.29w_c & 0.29w_c + w_b & 0.04w_c & -0.2w_c & -0.5w_c & 0.04w_c & 0 & 0 \\ 0.04w_c & 0.04w_c & 0.04w_c & 0.29w_c + w_b & -0.2w_c & 0 & 0.29w_c & -0.5w_c & 0 \\ -0.2w_c & -0.2w_c & -0.2w_c & -0.2w_c & q + w_c + w_b & 0 & -0.2w_c & 0 & 0 \\ 0 & -0.5w_c & -0.5w_c & 0 & 0 & q + w_c + w_b & 0 & 0 & 0 \\ 0.04w_c & 0.04w_c & 0.04w_c & 0.29w_c & -0.2w_c & 0 & 0.29w_c + w_b & -0.5w_c & 0 \\ 0 & 0 & 0 & -0.5w_c & 0 & 0 & -0.5w_c & q + w_c + w_b & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q + w_b \end{bmatrix} \tag{5.7}$$

To design $\bar{\mathbf{Q}}$ with a single weighting gain $\bar{q}$, we divide $\bar{q}$ out of the matrix in (5.7).

$$\bar{\mathbf{Q}} = \bar{q}\begin{bmatrix} 0.04\frac{w_c}{\bar{q}} + \frac{w_b}{\bar{q}} & 0.04\frac{w_c}{\bar{q}} & 0.04\frac{w_c}{\bar{q}} & 0.04\frac{w_c}{\bar{q}} & -0.2\frac{w_c}{\bar{q}} & 0 & 0.04\frac{w_c}{\bar{q}} & 0 & 0 \\ 0.04\frac{w_c}{\bar{q}} & 0.29\frac{w_c}{\bar{q}} + \frac{w_b}{\bar{q}} & 0.29\frac{w_c}{\bar{q}} & 0.04\frac{w_c}{\bar{q}} & -0.2\frac{w_c}{\bar{q}} & -0.5\frac{w_c}{\bar{q}} & 0.04\frac{w_c}{\bar{q}} & 0 & 0 \\ 0.04\frac{w_c}{\bar{q}} & 0.29\frac{w_c}{\bar{q}} & 0.29\frac{w_c}{\bar{q}} + \frac{w_b}{\bar{q}} & 0.04\frac{w_c}{\bar{q}} & -0.2\frac{w_c}{\bar{q}} & -0.5\frac{w_c}{\bar{q}} & 0.04\frac{w_c}{\bar{q}} & 0 & 0 \\ 0.04\frac{w_c}{\bar{q}} & 0.04\frac{w_c}{\bar{q}} & 0.04\frac{w_c}{\bar{q}} & 0.29\frac{w_c}{\bar{q}} + \frac{w_b}{\bar{q}} & -0.2\frac{w_c}{\bar{q}} & 0 & 0.29\frac{w_c}{\bar{q}} & -0.5\frac{w_c}{\bar{q}} & 0 \\ -0.2\frac{w_c}{\bar{q}} & -0.2\frac{w_c}{\bar{q}} & -0.2\frac{w_c}{\bar{q}} & -0.2\frac{w_c}{\bar{q}} & \frac{q}{\bar{q}} + \frac{w_c}{\bar{q}} + \frac{w_b}{\bar{q}} & 0 & -0.2\frac{w_c}{\bar{q}} & 0 & 0 \\ 0 & -0.5\frac{w_c}{\bar{q}} & -0.5\frac{w_c}{\bar{q}} & 0 & 0 & \frac{q}{\bar{q}} + \frac{w_c}{\bar{q}} + \frac{w_b}{\bar{q}} & 0 & 0 & 0 \\ 0.04\frac{w_c}{\bar{q}} & 0.04\frac{w_c}{\bar{q}} & 0.04\frac{w_c}{\bar{q}} & 0.29\frac{w_c}{\bar{q}} & -0.2\frac{w_c}{\bar{q}} & 0 & 0.29\frac{w_c}{\bar{q}} + \frac{w_b}{\bar{q}} & -0.5\frac{w_c}{\bar{q}} & 0 \\ 0 & 0 & 0 & -0.5\frac{w_c}{\bar{q}} & 0 & 0 & -0.5\frac{w_c}{\bar{q}} & \frac{q}{\bar{q}} + \frac{w_c}{\bar{q}} + \frac{w_b}{\bar{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{q}{\bar{q}} + \frac{w_b}{\bar{q}} \end{bmatrix}$$

$$= \bar{q}\begin{bmatrix} 0.04\beta_c + \beta_b & 0.04\beta_c & 0.04\beta_c & 0.04\beta_c & -0.2\beta_c & 0 & 0.04\beta_c & 0 & 0 \\ 0.04\beta_c & 0.29\beta_c + \beta_b & 0.29\beta_c & 0.04\beta_c & -0.2\beta_c & -0.5\beta_c & 0.04\beta_c & 0 & 0 \\ 0.04\beta_c & 0.29\beta_c & 0.29\beta_c + \beta_b & 0.04\beta_c & -0.2\beta_c & -0.5\beta_c & 0.04\beta_c & 0 & 0 \\ 0.04\beta_c & 0.04\beta_c & 0.04\beta_c & 0.29\beta_c + \beta_b & -0.2\beta_c & 0 & 0.29\beta_c & -0.5\beta_c & 0 \\ -0.2\beta_c & -0.2\beta_c & -0.2\beta_c & -0.2\beta_c & \alpha + \beta_c + \beta_b & 0 & -0.2\beta_c & 0 & 0 \\ 0 & -0.5\beta_c & -0.5\beta_c & 0 & 0 & \alpha + \beta_c + \beta_b & 0 & 0 & 0 \\ 0.04\beta_c & 0.04\beta_c & 0.04\beta_c & 0.29\beta_c & -0.2\beta_c & 0 & 0.29\beta_c + \beta_b & -0.5\beta_c & 0 \\ 0 & 0 & 0 & -0.5\beta_c & 0 & 0 & -0.5\beta_c & \alpha + \beta_c + \beta_b & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha + \beta_b \end{bmatrix}$$

In this configuration, $\beta_c = \frac{w_c}{\bar{q}}$, $\beta_b = \frac{w_b}{\bar{q}}$, and $\alpha = \frac{q}{\bar{q}}$.

While the $\bar{\mathbf{Q}}$ matrix structure is image specific, weighting gain design can be simplified to the following three steps: (1) select the appropriate relative importance of the different performance metrics such that they satisfy the condition $\alpha + \sum_i \beta_i = 1$, (2) design $\bar{q}$ to maximize performance, and (3) design s and r to meet robustness and convergence requirements.

5.1.3 Simulation Demonstration

To determined the effect of $\bar{q}$ design on image quality, we conduct a series of demonstration tests utilizing the cost function (5.6), and tiger image from Fig. 4.3. Relative importance of the metrics is assigned as the following, $\{q, w_c, w_b\} = \{0.5, 0.45, 0.05\}$, where brightness is identified as the least important metric, respectively. Utilizing these gains, the weighting matrix $\bar{\mathbf{Q}}$ can be written as,

$$\bar{\mathbf{Q}} = \bar{q}(0.5\mathbf{\Psi}^T\mathbf{\Psi} + 0.45\mathbf{X}_c^T\mathbf{X}_c + 0.05\mathbf{X}_b^T\mathbf{X}_b)$$

The weighting gains s and r are fixed constants $\{s, r\} = \{1e^{-4}, 1e^{-3}\}$, while $\bar{q}$ is varied as $\{0.5, 1, 2\}$. The weighting gains and quality indices are provided in Table 5.1. As can be seen, larger $\bar{q}$ gains lead to smaller image quality values, where a small quality index indicates better image quality.

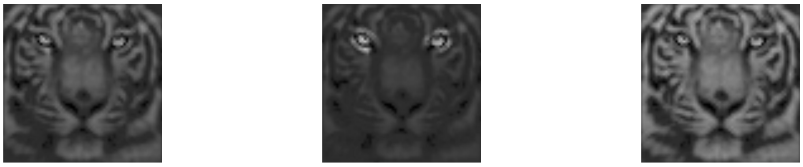
Table 5.1: Weighting gains and quality indices

	$\bar{q}$	q	w_c	w_b	s	r	Quality index
Case I	0.5	0.25	0.225	0.025	$1e^{-4}$	$1e^{-3}$	0.074
Case II	1	0.5	0.45	0.05	$1e^{-4}$	$1e^{-3}$	0.029
Case III	2	1	0.9	0.1	$1e^{-4}$	$1e^{-3}$	0.016

Table 5.2: Alternative weighting gains for same quality index

	q	w_c	w_b	s	r	Quality index
Case I	0.25	0.225	0.025	$1e^{-4}$	$1e^{-3}$	0.074
Alternative I	1	1	0.0174	$1e^{-4}$	$1e^{-3}$	0.074
Alternative II	1	0	0.038	$1e^{-4}$	$1e^{-3}$	0.074

Interestingly, matrix designs that utilize a centralized $\bar{\mathbf{Q}}$ and constant $\{\alpha, \beta_i\}$ values have a one-to-one mapping between image quality and the single gain $\bar{q}$. This mapping is defined through the fixed ratios between the weighting gain and the performance metrics. Recall that if the performance does not satisfy the design requirements, the $\{\alpha, \beta_i\}$ ratios can be revised and $\bar{q}$ re-tuned until a satisfactory performance



(a) Case I (b) Alternative I (c) Alternative II

Figure 5.2: Output images with the same quality index for different sets of weighting gains.

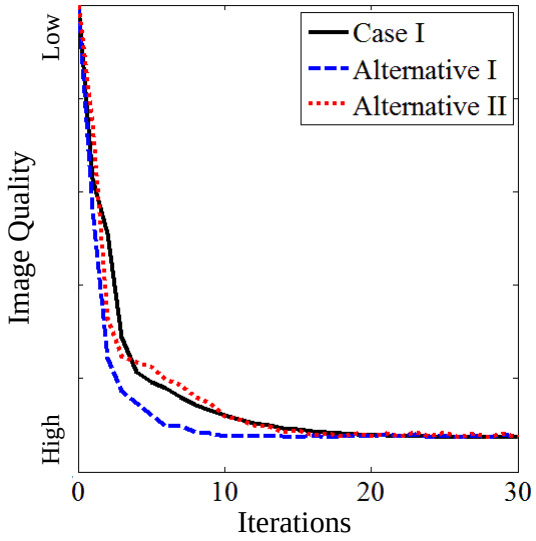


Figure 5.3: Image quality convergence for various gain designs.

is achieved.

As stated above, using the centralized design may result in a more indirect understanding of the relative importance of the different performance metrics. For example, in a distributed framework, alternative weighting gain designs may result in the same quality index value with noticeably varying output results, see Table 5.2 and Figs. 5.2 and 5.3. While this variation is possible with a centralized design approach, the direct design flexibility obtained by using a distributed design approach provides much faster tuning of the output results, with more obvious design intuition.

5.2 Single $\bar{\mathbf{S}}$ Design

Similar to the single $\bar{\mathbf{Q}}$ weighting matrix design, a set of assumptions must be met to enable the combination of the weighting matrices $\mathbf{S}$ and $\mathbf{W}_i$ into a single matrix.

A1 The desired values for the competing metrics are zero; i.e. $z_{i,j} = 0$ for all classification II or III performance metrics.

A2 Weighting gains s and w_i are directly proportional to the centralized gain $\bar{s}$.

Applying these assumptions, the cost function (3.6) can be rewritten as,

$$\begin{aligned} J &= (\boldsymbol{\Psi}\mathbf{e}_{j+1})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{e}_{j+1}) + \mathbf{u}_{j+1}^T\mathbf{S}\mathbf{u}_{j+1} + (\boldsymbol{\Delta}\mathbf{u}_{j+1})^T\mathbf{R}(\boldsymbol{\Delta}\mathbf{u}_{j+1}) + \sum_i \boldsymbol{\varepsilon}_{i,j+1}^T\mathbf{W}_i\boldsymbol{\varepsilon}_{i,j+1} \\ &= (\boldsymbol{\Psi}\mathbf{e}_{j+1})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{e}_{j+1}) + \mathbf{u}_{j+1}^T\bar{\mathbf{S}}\mathbf{u}_{j+1} + (\boldsymbol{\Delta}\mathbf{u}_{j+1})^T\mathbf{R}(\boldsymbol{\Delta}\mathbf{u}_{j+1}). \end{aligned}$$

For classification II performance metrics, the centralized $\bar{\mathbf{S}}$ matrix is defined as,

$$\begin{aligned} \bar{\mathbf{S}} &= \mathbf{S} + \sum_i (\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H}) \\ &= s\mathbf{I} + \sum_i w_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H}) \\ &= \bar{s}\Big(\frac{s}{\bar{s}}\Big)\mathbf{I} + \bar{s}\Big(\sum_i \frac{w_i}{\bar{s}}(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\Big) \\ &= \bar{s}\Big(\alpha\mathbf{I} + \sum_i \beta_i(\mathbf{X}_i\mathbf{H})^T(\mathbf{X}_i\mathbf{H})\Big). \end{aligned}$$

For classification III performance metrics, the centralized $\bar{\mathbf{S}}$ matrix is defined as,

$$\begin{aligned} \bar{\mathbf{S}} &= \mathbf{S} + \sum_i \mathbf{T}_i^T\mathbf{W}_i\mathbf{T}_i \\ &= s\mathbf{I} + \sum_i w_i\mathbf{T}_i^T\mathbf{T}_i \\ &= \bar{s}\Big(\frac{s}{\bar{s}}\Big)\mathbf{I} + \bar{s}\Big(\sum_i \frac{w_i}{\bar{s}}\mathbf{T}_i^T\mathbf{T}_i\Big) \end{aligned}$$

$$= \bar{s}\Big(\alpha \mathbf{I} + \sum_i \beta_i \mathbf{T}_i^T \mathbf{T}_i\Big),$$

where $s = \alpha\bar{s}$ and $w_i = \beta_i \bar{s}$, and $\{\alpha, \beta_i\}$ satisfy the condition $\alpha + \sum_i \beta_i = 1$.

As with the constrained single $\bar{\mathbf{Q}}$, if the ratios between s and w_i are fixed, we can simplify the controller design to a single $\bar{s}$ gain. It is important to note that $\bar{s}$ is related to both the control input and the competing performance metrics. As such, an increase in $\bar{s}$ results in an increase in the weighting on the competing performance metrics, as well as a more conservative control approach overall as a result of the increased weighting on the control input. In this architecture, the performance/robustness tradeoff for additional performance metrics is directly correlated to the gain $\bar{s}$. To truly adjust the tradeoff between robustness and performance, one must modify the $\{\alpha, \beta_i\}$ gains and then re-tune the $\bar{s}$ gain. Once gain, the distributed approach presented in the original multi-objective framework leads to more direct design flexibility with a more intuitive tuning structure.

5.3 Concluding Remarks

This chapter presents the design and analysis of a single centralized weighting matrix that addresses multiple complementary or competing performance metrics as an extension to the multi-objective iterative learning control framework presented in Chapters III and IV. The specific contributions of this chapter include: (1) the design and analysis of a single $\bar{\mathbf{Q}}$ and $\bar{\mathbf{S}}$ weighting matrix, and (2) a simple simulation demonstration for the single $\bar{\mathbf{Q}}$ weighting matrix that illustrates the design constraints in the centralized approach. From this chapter, we can clearly illustrate the enhanced design flexibility that is provided through the distributed approach.

CHAPTER VI

Multi-Objective Region-to-Region Iterative Learning Control

Conventionally, ILC has been applied to systems that perform the same task over numerous iterations. In Chapter III we considered the task of tracking a subset of reference points that are fixed for all iterations. In this chapter we explore the case where the system exhibits reference uncertainty. For example, consider an application where an unmanned air vehicle (UAV) is tasked with collecting information from an unattended ground sensor (UGS). If the exact locations of the UGSs are uncertain, we can assume a bounded region from which sensor data transmission can be guaranteed. To control such systems, we investigate region-to-region learning approaches for handling uncertain references. Controller analysis and a simulation demonstration are provided to validate the proposed learning controller.

6.1 Reformulation of the framework with reference uncertainty

Building from the point-to-point ILC framework defined in Chapter III, we introduce additive uncertainty into the reference signal $\mathbf{y}_d$ such that $\mathbf{y}_{\Delta_{d,j}} = \mathbf{y}_d + \mathbf{\Delta}_{d,j}$.

The error signal can now be written as,

$$\mathbf{e}_j = \mathbf{y}_{\Delta_{d,j}} - \mathbf{H}\mathbf{u}_j. \tag{6.1}$$

Given the update law $\mathbf{u}_{j+1} = \mathbf{L}_u\mathbf{u}_j + \mathbf{L}_e\mathbf{e}_j$, learning filters (3.3)-(3.4), and the error signal from (6.1), recursion of the error is derived as,

$$\begin{aligned}\boldsymbol{\Psi}\mathbf{e}_{j+1} =&(\boldsymbol{\Psi}\mathbf{H})\Big\{(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{H}) + \mathbf{S} + \mathbf{R}\Big\}^{-1}\mathbf{R}(\boldsymbol{\Psi}\mathbf{H})^{-1}\boldsymbol{\Psi}\mathbf{e}_j \\ &+ (\boldsymbol{\Psi}\mathbf{H})\Big\{(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{H}) + \mathbf{S} + \mathbf{R}\Big\}^{-1}\mathbf{S}(\boldsymbol{\Psi}\mathbf{H})^{-1}\boldsymbol{\Psi}(\mathbf{y}_d + \boldsymbol{\Delta}_{d,j})\end{aligned} \tag{6.2}$$

From (6.2) one can identify that the convergence rate $\gamma = (\boldsymbol{\Psi}\mathbf{H})\Big\{(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{H}) + \mathbf{S} + \mathbf{R}\Big\}^{-1}\mathbf{R}(\boldsymbol{\Psi}\mathbf{H})^{-1}$ is equivalent to the convergence rate derived in the point-to-point ILC algorithm presented in Chapter III. However, the presence of the reference uncertainty $\boldsymbol{\Delta}_{d,j}$ will directly impact the performance of the system.

Substituting (6.1) into update law $\mathbf{u}_{j+1} = \mathbf{L}_u\mathbf{u}_j + \mathbf{L}_e\mathbf{e}_j$,

$$\begin{aligned}\mathbf{u}_{j+1} =&\Big\{(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{H}) + \mathbf{S} + \mathbf{R}\Big\}^{-1}\mathbf{R}\mathbf{u}_j \\ &+ \left\{(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{H}) + \mathbf{S} + \mathbf{R}\right\}^{-1}(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}\boldsymbol{\Psi}\left(\mathbf{y}_d + \boldsymbol{\Delta}_{d,j}\right).\end{aligned} \tag{6.3}$$

Applying the w-transformation which was introduced in [8, 78, 79, 80], (6.3) is rewritten as,

$$\begin{aligned}&\Big[w\mathbf{I} - \Big\{(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{H}) + \mathbf{S} + \mathbf{R}\Big\}^{-1}\mathbf{R}\Big]\mathbf{u}_j \\ &\quad= \left\{(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{H}) + \mathbf{S} + \mathbf{R}\right\}^{-1}(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}\boldsymbol{\Psi}\left(\mathbf{y}_d + \boldsymbol{\Delta}_{d,j}\right)\end{aligned} \tag{6.4}$$

For the worst case scenario, it is assumed that $w = 1$ for the nominal reference values and $w = -1$ for the additive uncertainty [8]. With these assumptions, (6.4) is

rewritten as,

$$\begin{aligned}\mathbf{u}_j =& \left\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\mathbf{S}\right\}^{-1}(\mathbf{\Psi H})^T\mathbf{Q}\mathbf{\Psi}\mathbf{y}_d \\ &- \left\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\mathbf{S}+2\mathbf{R}\right\}^{-1}(\mathbf{\Psi H})^T\mathbf{Q}\mathbf{\Psi}\mathbf{\Delta}_{d,j}.\end{aligned} \tag{6.5}$$

Substituting (6.5) into $\mathbf{\Psi e}_j = \mathbf{\Psi y}_{\Delta_{d,j}} - \mathbf{\Psi H u}_j$ yields,

$$\begin{aligned}\mathbf{\Psi e}_j =& \mathbf{\Psi y}_{\Delta_{d,j}} - \mathbf{\Psi H u}_j \\ =& \left[\mathbf{I}-(\mathbf{\Psi H})\left\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\mathbf{S}\right\}^{-1}(\mathbf{\Psi H})^T\mathbf{Q}\right]\mathbf{\Psi}\mathbf{y}_d \\ &+\left[\mathbf{I}+(\mathbf{\Psi H})\left\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\mathbf{S}+2\mathbf{R}\right\}^{-1}(\mathbf{\Psi H})^T\mathbf{Q}\mathbf{\Psi}\right]\mathbf{\Delta}_{d,j}\end{aligned} \tag{6.6}$$

Lemma VI.1. *For bounded additive uncertainty $\mathbf{\Delta}_{d,j}$, the 2-norm of the steady-state error $\|\mathbf{\Psi e}_\infty\|_2$ at the selected points* (6.6) *is bounded.*

Proof of Lemma VI.1 : *If the trial varying disturbances are bounded, $\|\mathbf{\Delta}_{d,j}\|_2 \leq \boldsymbol{\beta} < \infty$, the steady state error is bounded,*

$$\begin{aligned}\lim_{j\to\infty} sup\left\|\mathbf{\Psi e}_j\right\|_2 \leq& \lim_{j\to\infty} sup\Big\|\left[\mathbf{I}-(\mathbf{\Psi H})\left\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\mathbf{S}\right\}^{-1}(\mathbf{\Psi H})^T\mathbf{Q}\right]\mathbf{\Psi}\mathbf{y}_d \\ &+\left[\mathbf{I}+(\mathbf{\Psi H})\left\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\mathbf{S}+2\mathbf{R}\right\}^{-1}(\mathbf{\Psi H})^T\mathbf{Q}\mathbf{\Psi}\right]\mathbf{\Delta}_{d,j}\Big\|_2 \\ \leq& \left\|\mathbf{I}-(\mathbf{\Psi H})\left\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\mathbf{S}\right\}^{-1}(\mathbf{\Psi H})^T\mathbf{Q}\right\|_2\left\|\mathbf{\Psi}\mathbf{y}_d\right\|_2 \\ &+\left\|\mathbf{I}+(\mathbf{\Psi H})\left\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\mathbf{S}+2\mathbf{R}\right\}^{-1}(\mathbf{\Psi H})^T\mathbf{Q}\mathbf{\Psi}\right\|_2\boldsymbol{\beta}\end{aligned}$$

Because of the uncertainty, the error will continue to fluctuate after convergence. □

Figure 6.1 and 6.2 show the sample of the uncertain locations and the error boundedness of the point-to-point ILC with uncertain locations.

An interesting application example can be found in the UAV/UGS surveillance tracking problem presented in [18]. If the bounded region lies within the signal transmission range, the system is guaranteed to transfer data; e.g. the *error signal* will be zero at any location where the UGS successfully transfers data to the UAV. To

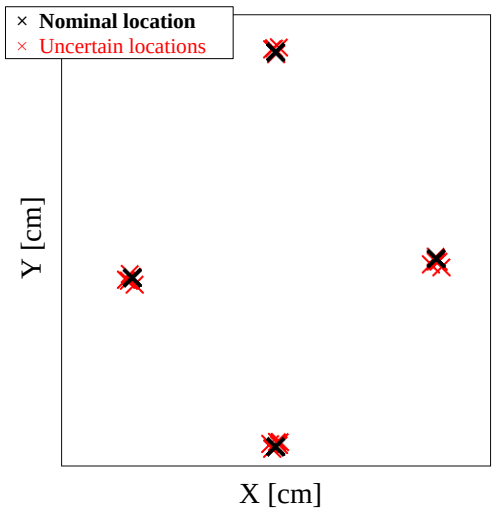


Figure 6.1: Sample of uncertain locations.

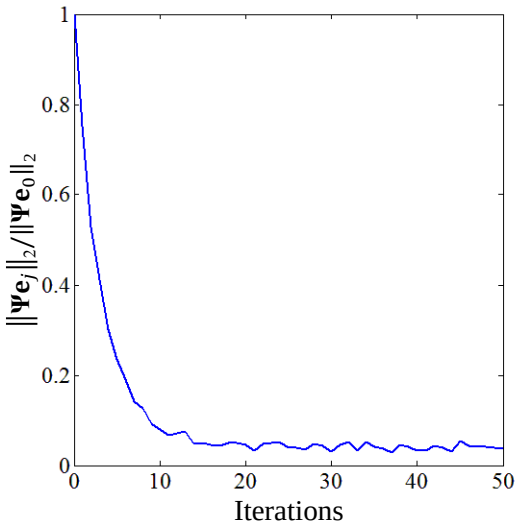


Figure 6.2: Error boundedness of point-to-point ILC with uncertain locations.

address this we introduce a region-to-region learning framework in which the regions are modified to account for reference uncertainty.

6.2 A region-to-region framework

Recent surveillance strategies from the Air Force combine UAV and multiple UGSs into a single cohesive unit to leverage attributes from each system [81, 82]. Combining these two units into a single system results in a cooperative surveillance scenario in which the UAVs rely on information from the UGSs in order to detect and respond to an intruder. Given the repetitive nature of surveillance, ILC can be utilized to realize performance improvements from pass to pass.

In this work we assume that a region of successful communication transfer around an UGS is known; however, the location of the UGS is uncertain. If the uncertain locations $\mathbf{\Psi y}_\Delta$ are bounded, the intersections of the uncertain regions can be used to identify a new region in which the UGSs can guarantee data transfer to the

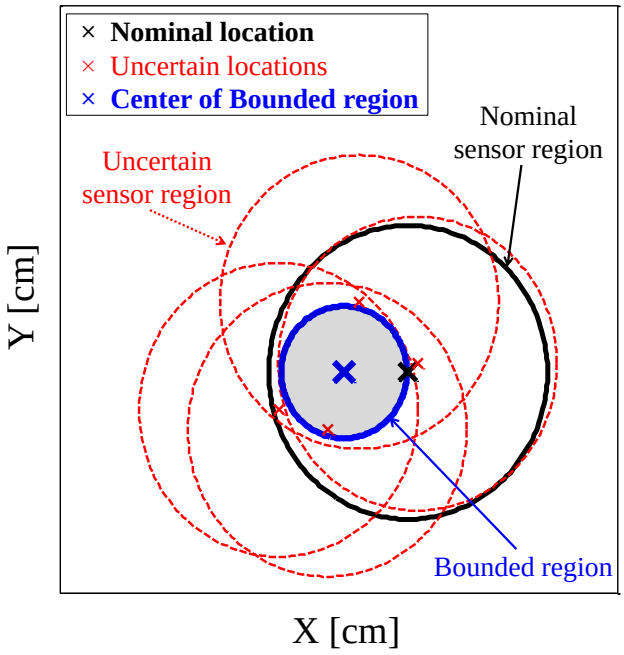


Figure 6.3: Nominal locations, uncertain locations and center point of estimated region with signal transfer ranges.

UAV. Thus, the control objective is to determine an optimized surveillance flight that ensures that the UAV flies within the bounded region around an uncertain UGS location.

To simplify the problem, we define the following assumptions:

A1 The region of data transfer for a UGS is a circular pattern.

A2 The bounded region for the UGS is a circular pattern and contained within the bounded uncertain regions, Fig. 6.3.

A3 The uncertainty in the reference locations can be quantified through repetitive passes of a UAV.

A4 Repetitive passes of a UAV are used to identify the center-point of the UGS for each location. A data transfer region emanates in a circular pattern from this center-point.

A5 The bounded region of the UGS is iteration invariant.

Assumptions 1 and 2 are used to simplify the region mapping. In practice, the pattern of the region may be distorted due to environmental conditions such as a mountain or other obstacles that block the signal, or orientation and alignment issues with the UGS. Numerous passes in various orientations and locations would be required to adequately identify the region pattern within which the data signal is guaranteed to transfer. Assumptions 3 and 4 ensure that a bounded region can be determined within the reference uncertainty, with the center-point identified from the signal data. In practice, the center-point of the uncertain reference locations would not be directly measurable with the UAV unless a camera is utilized. In the absence of vision, sensor strength and directionality may be used to estimate the center-point for updated UGS locations. Lastly, assumption 5 changes the problem from an uncertainty problem to a region-based tracking problem. Once the bounded region has been identified, this assumption means that the system does not consider further uncertainties in the UGS location. This is reasonable in practice since majority of the issues that would limit the sensor range or disrupt the pattern of the region are static conditions.

Once the regions are defined, the region tracking error is defined as,

$$\hat{\mathbf{e}}(n_i) = \begin{cases} 0, & \|\mathbf{y}_c(n_i) - \mathbf{y}(k)\|_2 \leq \mathbf{r}(n_i) \text{ for any } k \\ \mathbf{e}(n_i), & \text{otherwise} \end{cases}, \tag{6.7}$$

where $\mathbf{e}(n_i) = \mathbf{y}_c(n_i) - \mathbf{y}(n_i)$, $\mathbf{y}_c(n_i)$ is the center point of the new region and $\mathbf{r}(n_i)$ is the radius of the bounded region. n_i are the selected surveillance locations for all $i = 1, \ldots, M$, $M < N$, where N denotes the total number of possible surveillance locations in a given X-Y region.

From (6.7), perfect tracking ensures $\hat{\mathbf{e}} = 0$, while a missed region is defined as $\hat{\mathbf{e}} = \mathbf{\Psi e}$. Utilizing the modified tracking error, the cost function can now be written

as,

$$J = \hat{\mathbf{e}}_{j+1}^T \mathbf{Q} \hat{\mathbf{e}}_{j+1} + \mathbf{u}_{j+1}^T \mathbf{S} \mathbf{u}_{j+1} + (\mathbf{u}_{j+1} - \mathbf{u}_j)^T \mathbf{R} (\mathbf{u}_{j+1} - \mathbf{u}_j). \tag{6.8}$$

To derive the update law and learning filters, we consider the worst case tracking results,

$$\begin{aligned} \hat{\mathbf{e}}_{j+1} &= \mathbf{\Psi} \mathbf{e}_{j+1} \\ &= \hat{\mathbf{e}}_j - \mathbf{\Psi} \mathbf{H} (\mathbf{u}_{j+1} - \mathbf{u}_j) \end{aligned} \tag{6.9}$$

Applying (6.9) to (6.8) results in the standard point-to-point update law and learning filters from Chapter III, but the region trackin error is utilized instead of the point tracking error. Note that while controller design follows point-to-point ILC, the measured system error signal comes from the region tracking error $\hat{\mathbf{e}}_j$.

6.3 A multi-objective region-to-region framework

In this section, we introduce a multi-objective region-to-region learning framework. Adding the additional performance objective to the cost function in (6.8), the multi-objective region-to-region cost function becomes,

$$J = \hat{\mathbf{e}}_{j+1}^T \hat{\mathbf{Q}} \hat{\mathbf{e}}_{j+1} + \mathbf{u}_{j+1}^T \mathbf{S} \mathbf{u}_{j+1} \; + \; (\mathbf{u}_{j+1} - \mathbf{u}_j)^T \mathbf{R} (\mathbf{u}_{j+1} - \mathbf{u}_j) + \sum_i \boldsymbol{\varepsilon}_{i,j+1}^T \mathbf{W}_i \boldsymbol{\varepsilon}_{i,j+1},$$

where $\boldsymbol{\varepsilon}_{i,j+1}$ represents the error of an additional performance objective from one of the classifications presented in Chapter IV. If we once again consider a non-zero tracking error (the worst case, $\hat{\mathbf{e}}_{j+1} = \mathbf{\Psi} \mathbf{e}_{j+1}$), the system becomes identical to the multi-objective framework in Chapter IV.

Figure 6.4: Mobile wheeled robot, Lego Mindstorm

6.4 Simulation demonstration for UAV applications

6.4.1 Mobile wheeled robot

To demonstrate the multi-objective, region-based learning controller, a model of a mobile wheeled robot is used in simulation as a simplified version of the UAV system from [18], Fig. 6.4. The discrete time closed-loop transfer functions for each axis (assumed to be dynamically decoupled) are given as,

$$H_X(z) = \frac{0.002838z^{-1} - 9.966e^{-05}z^{-2} - 0.002576z^{-3}}{1 - 2.782z^{-1} + 2.576z^{-2} - 0.7935z^{-3}}$$
$$H_Y(z) = \frac{0.002374z^{-1} - 0.0001206z^{-2} - 0.002227z^{-3}}{1 - 2.829z^{-1} + 2.664z^{-2} - 0.8348z^{-3}}$$

with a sampling time of 0.05 seconds. H_X and H_Y are the closed-loop models of the x and y-axis of the wheeled mobile robot used in Chapter III.

For more realistic demonstrations, we consider a system subject to model uncertainty and noise. Each axis is subjected to additive uncertainty of the following

form:

$$\hat{H}_X(z) = \frac{-4.632e-05z^{-1}}{1-0.999z^{-1}}$$
$$\hat{H}_Y(z) = \frac{0.003542z^{-1}}{1-0.983z^{-1}}.$$

Additionally, external disturbances are approximated to be white noise, $\boldsymbol{\eta}_{x,j} = \mathcal{N}(0, 0.0125)$ and $\boldsymbol{\eta}_{y,j} = \mathcal{N}(0, 0.0125)$ for all j. Given these definitions, the plant dynamics of the wheeled mobile robot are defined as,

$$\begin{aligned}
\mathbf{x}_j =& \mathbf{H}_{x,t}\mathbf{u}_{x,j} + \boldsymbol{\eta}_{x,j} \\
=& (\mathbf{H}_x + \hat{\mathbf{H}}_x\boldsymbol{\Delta}_H)\mathbf{u}_{x,j} + \boldsymbol{\eta}_{x,j}, \\
\mathbf{y}_j =& \mathbf{H}_{y,t}\mathbf{u}_{y,j} + \boldsymbol{\eta}_{y,j} \\
=& (\mathbf{H}_y + \hat{\mathbf{H}}_y\boldsymbol{\Delta}_H)\mathbf{u}_{y,j} + \boldsymbol{\eta}_{y,j},
\end{aligned}$$

where $\hat{\mathbf{H}}_x$ and $\hat{\mathbf{H}}_y$ are the lifted form of the additive uncertainty, and $\|\boldsymbol{\Delta}_H\|_2 \leq 1$.

6.4.2 Desired task description

Unmanned autonomous vehicles (UAVs) have contributed to situational awareness through surveillance missions. UAVs can detect and gather data signals with a collection of on-board sensors; however, the sensor capabilities are limited because of the weight and energy requirements for the UAV. Unattended ground sensors (UGSs) can capture a wider range of information from a larger suite of sensors; however, the signal transfer capabilities are limited to relatively small distances. To leverage the attributes of both UAVs and UGSs, many surveillance missions are designed to combine UAVs and multiple UGSs into a single framework [15, 16, 17, 18].

For this simulation example, the estimated regions and their center points are illustrated in Fig. 6.5. The simulation includes 10 bounded regions with the ordering

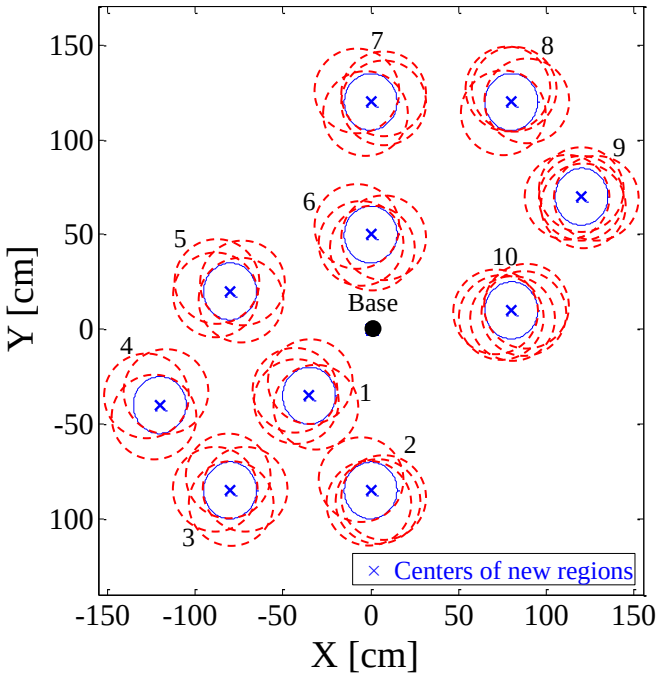


Figure 6.5: The estimated regions and their center points for simulation demonstrations

of the visits pre-defined (genrally solves as a traveling salesman problem).

6.4.3 Additional objective and cost function

To demonstrate the multi-objective aspect of this framework, we consider two performance metrics: 1) successful data transfer for all bounded regions, and 2) minimizing the overall distance travelled. The error signals for distance $\boldsymbol{\varepsilon}_x$ and $\boldsymbol{\varepsilon}_y$ are defined as,

$$\boldsymbol{\varepsilon}_x = \mathbf{l}_{x,d} - \mathbf{l}_x = \mathbf{l}_{x,d} - \mathbf{XHu}_x,$$

$$\boldsymbol{\varepsilon}_y = \mathbf{l}_{y,d} - \mathbf{l}_y = \mathbf{l}_{y,d} - \mathbf{XHu}_y,$$

where $\mathbf{l}_{x,d}$ and $\mathbf{l}_x$ denote desired and measured distances for the x-axis, and $\mathbf{l}_{y,d}$ and $\mathbf{l}_y$ denote desired and measured distances for the y-axis. The desired distance values $\mathbf{l}_{x,d}$ and $\mathbf{l}_{y,d}$ are assumed to have small values for minimization. Note that we do not set

these values equal to zero as a solution of no movement is a trivial and undesirable solution. $\mathbf{X}$ is a matrix that maps the output signal to the distance traveled for each axis,

$$\mathbf{X} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & -1 & 1 \end{pmatrix}.$$

The overall distance travelled is calculated as,

$$L = \sum_{k=1}^{N} \sqrt{\mathbf{l}_x^2(k) + \mathbf{l}_y^2(k)}.$$

Using these performance objectives, the cost function for a multi-objective region-to-region algorithm can be written as,

$$J = \mathbf{u}_{j+1}^T \mathbf{S} \mathbf{u}_{j+1} \; + \; (\mathbf{u}_{j+1} - \mathbf{u}_j)^T \mathbf{R} (\mathbf{u}_{j+1} - \mathbf{u}_j) + \underbrace{\hat{\mathbf{e}}_{j+1}^T \mathbf{Q} \hat{\mathbf{e}}_{j+1}}_{\text{Region Tracking}} \; + \; \underbrace{\boldsymbol{\varepsilon}_{j+1}^T \mathbf{W}_l \boldsymbol{\varepsilon}_{j+1}}_{\text{Minimizing Distance}}$$

6.5 Simulation results

Once the estimated regions and their center points are defined, we can compare point-to-point and region-to-region ILC in terms of meeting the performance objectives. Point-to-point ILC defines the tracking error with respect to the center-points of the new region, $\mathbf{e}_j(n_i) = \mathbf{y}_c(n_i) - \mathbf{y}_j(n_i)$, where $\mathbf{y}_c(n_i)$ is the center point at n_i, n_i are the selected instants for all $i = 1, \ldots, M$, $M < N$, and N denotes the number of possible points in a given X-Y region. Region-to-region ILC defines the tracking error in terms of the region $\hat{\mathbf{e}}_j$.

For the simulation demonstrations, the weighting gain q is fixed at 1 to emphasize

Table 6.1: Weighting gains for simulation demonstrations

Case	q	s	r	w
1 Point-to-point	1	$1e^{-5}$	$5e^{-2}$	0
2 Region-to-region	1	$1e^{-5}$	$5e^{-2}$	0
3 Multi-obj. region-to-region	1	$1e^{-5}$	$5e^{-2}$	$2.5e^{-1}$

the tracking performance. The weighting gains $[s, r]$ are fixed at $[1e^{-5}, 5e^{-2}]$ to satisfy the criteria for robustness and convergence. Table 6.1 provides the gain selections for all of the controllers.

As illustrated in Fig. 6.6, point-to-point ILC represents a conservative approach in which the UAV tracks the center-point of each identified region. While this approach guarantees strict signal transfer, it does not take advantage of the potential distance savings that could be achieved by leveraging the signal transfer range. Using the region-to-region ILC approach, the UAV is tasked with staying within the regions, resulting in a less constrained solution that can be optimized to minimize the distance traveled.

Simulation results for point-to-point, region-to-region, and multi-objective region-to-region control approaches are presented in Fig. 6.6. Figure 6.7 illustrates that all three learning approaches can guarantee signal transfer between the UAV and the UGSs, i.e. zero converged error. Figure 6.8 presents the overall distance travelled with the three learning algorithms. Utilizing the advantage of a guaranteed signal transfer within the regions results in a shorter path as compared to the point-to-point approach. Furthermore, directly weighting a distance-based performance metric results in the shortest overall distance traveled, while maintaining zero converged error (multi-objective region to region ILC).

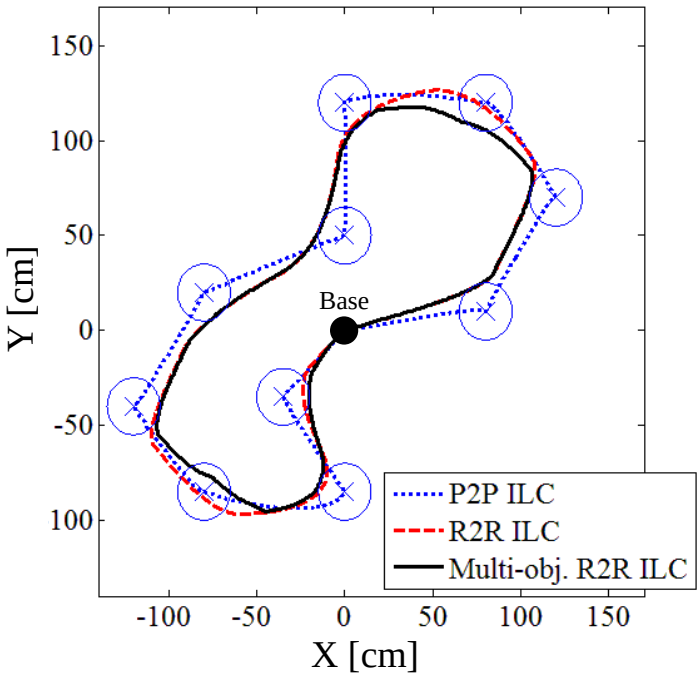


Figure 6.6: Output Paths for all cases: point-to-point(P2P), region-to-region(R2R), and multi-objective region-to-region(Multi-obj. R2R) ILC

6.6 Concluding Remarks

This chapter presents a multi-objective region-to-region iterative learning control framework. The specific contributions of this chapter include:, (1) stability and convergence analysis for learning controllers with reference uncertainty, (2) development and analysis of a region-to-region learning framework to address uncertainty in the reference, and (3) simulation validations of the region-based learning framework.

To validate the region-based learning framework we use a model of a wheeled mobile robot system to represent a UAV in a UAV/UGS surveillance scenario. Simulation results demonstrate the additional design and performance flexibility that can be leveraged when using the modified signal transfer region, as opposed to a center point location used in point-to-point ILC. This flexibility can then be utilized to address additional performance objectives in a multi-objective region-to-region learning framework.

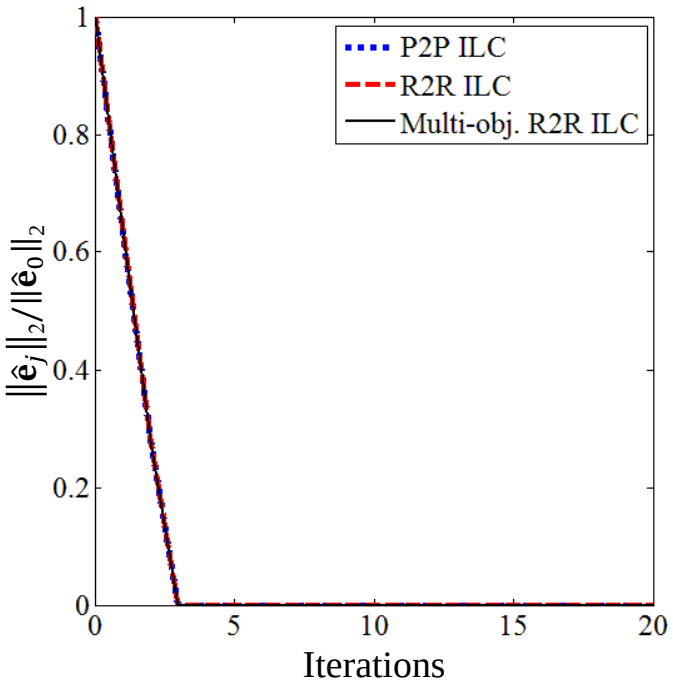


Figure 6.7: Normalized $\|\hat{\mathbf{e}}_j\|_2$ for point-to-point(P2P), region-to-region(R2R), and multi-objective region-to-region(Multi-obj. R2R) ILC

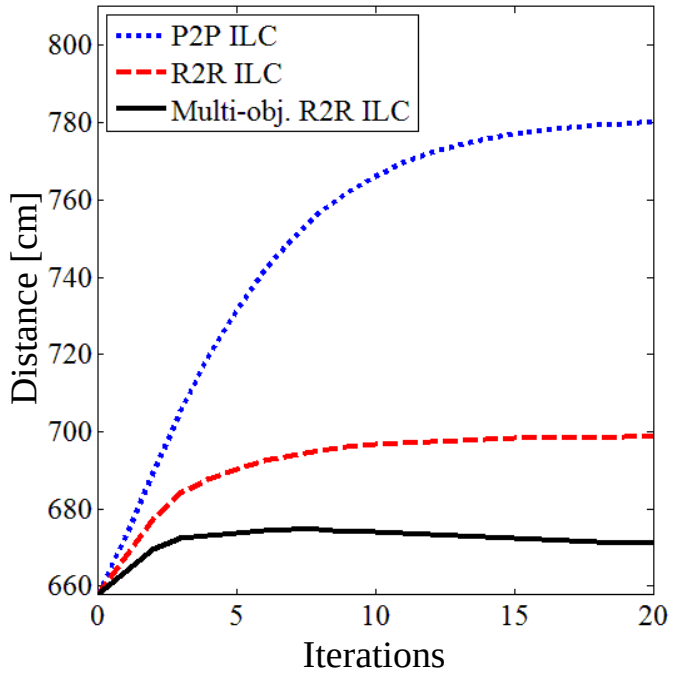


Figure 6.8: Overall distance travelled for point-to-point(P2P), region-to-region(R2R), and multi-objective region-to-region(Multi-obj. R2R) ILC

CHAPTER VII

Concluding Remarks and Future Directions

The goal of this dissertation was to develop various methods for relaxing the fundamental assumption of iteration invariant reference signals with full reference signal requirements. The methods were designed to leverage advantages that result from allowing reference variations from pass to pass, while focusing on a subset of key reference points. Modifications to the conventional norm optimal ILC framework were provided for both the temporal and spatial domains in order to apply ILC to new application spaces in both autonomous vehicles and additive manufacturing.

To achieve this goal, the following scientific questions were addressed:

(1) *Can the requirement for strict reference tracking constraint be relaxed in order to leverage the newly available control flexibility towards the improvement of multiple performance metrics?*

As we demonstrated in Chapter III, relaxing the strict reference tracking constraint led to additional control bandwidth that enabled the controller to address multiple performance metrics. This initial framework, presented in [13, 29], provided the foundation for the multi-objective ILC framework. Simulation and experimental results validated this initial framework.

(2) *What knowledge from temporal ILC can be applied towards the development of*

spatial ILC? What are the differences that drive new design methodologies or stability analysis?

Building from our previous work in multi-objective ILC and the initial development of a spatial ILC framework [49], we extended the spatial framework to a multi-objective learning framework in the spatial domain [38, 51]. This framework leverages 2D convolution to address spatial closeness for applications such as additive manufacturing. This work resulted in a more formal multi-objective formulation including the expansion of the framework to four categories of performance metrics with stability and convergence analysis. Simulation results on a model of a high-resolution additive manufacturing system validated the extended framework. A comparison analysis between a centralized and distributed weighting gain design demonstrated the additional performance flexibility gained through the use of the distributed weighting matrix design.

(3) *Can strict performance be guaranteed for iteration varying references with a certain level of uncertainty?*

To address this question, we first provided the stability and performance analysis for a learning controller with reference uncertainty. We then extended this analysis to incorporate a bounded region-based reference tracking problem in which the reference uncertainty is addressed through the development of a bounded region. This region can be used to guarantee the tracking performance of systems within a bounded neighborhood. A simulation example of a surveillance problem in which a UAV is tasked with collecting data from a series of UGSs validated the region-based approach. Comparisons with conventional point-to-point, region-to-region, and a multi-objective region-based ILC framework demonstrated the performance flexibility that can be achieved when leveraging the regions.

7.1 Future directions

This research provides the foundation for multi-objective iterative learning control. Despite the progress and results presented in this dissertation, there exist remaining theoretical and application gaps that should be addressed. Below we present an incomplete list of these potential extensions:

(1) Validation of the multi-objective spatial ILC framework on the experimental high-resolution additive manufacturing platform.

(2) Validation of the multi-objective region-to-region ILC framework on the experimental wheeled mobile robot testbed.

(3) Relaxation of the region description to include non-symmetric signal transmission patterns to better represent environmental disturbances and device malfunctions.

(4) Expanding the iteration invariance assumption to consider iteration varying reference signals.

(5) Extension of the multi-objective iterative learning control framework to other application domains such as supply chain management and manufacturing. In these domain spaces, historical data may provide additional insights into the decision making process. Additionally, alternative performance requirements such as cost, time, product quality, and throughput provide opportunities for a multi-objective learning framework.

APPENDICES

APPENDIX A

Wheeled Mobile Robot

For the simulation and experimental tests conducted in Chapters III and VI, we utilize the wheeled mobile robot shown in Fig. A.1. The robot moves on two wheels driven by independent electric motors. A gyroscope is mounted at the center of the wheels to measure heading angle of the robot. Figure A.2 shows a block diagram of the wheeled mobile robot.

To satisfy the requirement for a stable system, we stabilize the system using a simple proportional feedback controller that was tuned for stability rather than performance to highlight the performance improvements that can be obtained through the use of ILC. The proportional gains for the x and y-axis are $\{2, 2\}$, respectively.

Figure A.1: Mobile wheeled robot, LEGO NXT Mindstorm

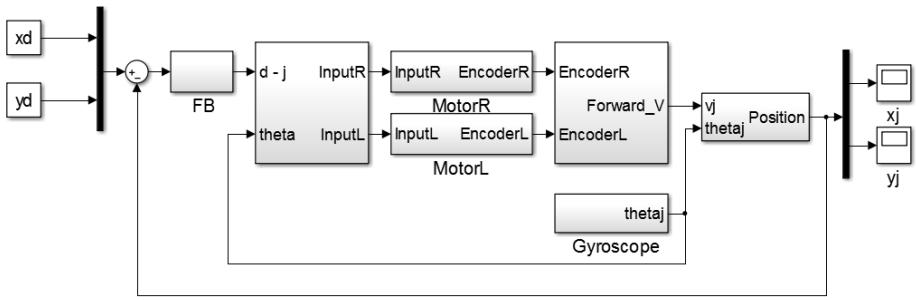


Figure A.2: Block diagram for the experimental system

Reference positions $\mathbf{x}_d$ and $\mathbf{y}_d$ are the desired locations of the center point of the mobile robot. $\mathbf{x}_j$ and $\mathbf{y}_j$ are the measured locations. Using the robot local frame (i.e., the moving coordinate system), the error coordinates can be defined as,

$$\begin{bmatrix} \mathbf{e}_{x,local} \\ \mathbf{e}_{y,local} \end{bmatrix} = \begin{bmatrix} \cos\boldsymbol{\theta}_j & \sin\boldsymbol{\theta}_j \\ -\sin\boldsymbol{\theta}_j & \cos\boldsymbol{\theta}_j \end{bmatrix} \begin{bmatrix} \mathbf{x}_d - \mathbf{x}_j \\ \mathbf{y}_d - \mathbf{y}_j \end{bmatrix}.$$

We measure θ_j using a gyroscope. Each motor contains an encoder that measures the rotation of the motor, with the radius of each wheel given as a constant value. Using this information we calculate the velocity of the vehicle as,

$$\mathbf{v}_j = R_w \frac{(\omega_{R,j} + \omega_{L,j})}{2},$$

where R_w is the wheel radius, $\omega_{R,j}$ is the angular velocity of the right wheel, $\omega_{L,j}$ is the angular velocity of the left wheel, and $\mathbf{v}_j$ is the vehicle speed. Using these measured and calculated values, the measured position is calculated as,

$$\begin{bmatrix} \mathbf{x}_j \\ \mathbf{y}_j \end{bmatrix} = \int \begin{bmatrix} \dot{\mathbf{x}}_j \\ \dot{\mathbf{y}}_j \end{bmatrix} dt = \int \begin{bmatrix} \mathbf{v}_j \cos\boldsymbol{\theta}_j \\ \mathbf{v}_j \sin\boldsymbol{\theta}_j \end{bmatrix} dt.$$

It should be noted that these coordinate transformations and variable calculations are embedded within the robot. In practice, once the system is stabilized with a

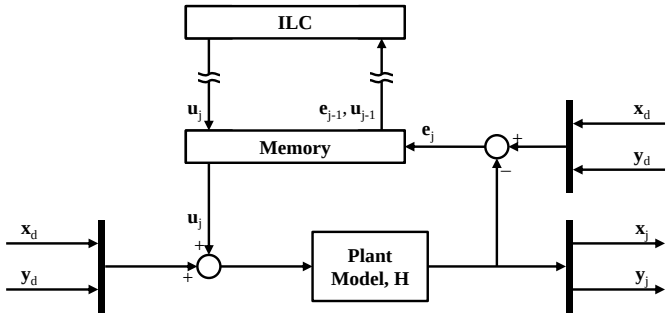


Figure A.3: Series architecture in which the reference signal to the system is updated. The plant model **H** is the full closed-loop system. [1]

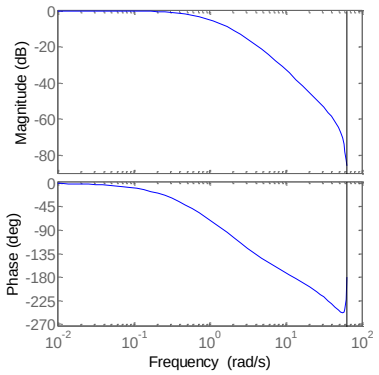
feedback controller, we input a desired path $\mathbf{x}_d, \mathbf{y}_d$ and obtained the measured output $\mathbf{x}_j, \mathbf{y}_j$ from the robot interface. To apply ILC to this robotic system, we utilize the series architecture shown in Fig. A.3. This architecture is particularly useful when applying ILC to systems that do not allow direct access to modifying the control signal to the plant [1].

Using the experimental testbed, we can estimate the plant model as a MIMO system with a sampling time of 0.05 seconds. As we can see Fig. A.3, the combined input signals to the plant are $\{\mathbf{x}_d + \mathbf{u}_{x,j}, \mathbf{y}_d + \mathbf{u}_{y,j}\}$, with output signals $\{\mathbf{x}_j, \mathbf{y}_j\}$. For system identification, we assume $\{\mathbf{u}_{x,j} = \mathbf{0}, \mathbf{u}_{y,j} = \mathbf{0}\}$. After sending a series of inputs to the system, the plant model is identified using the Matlab function 'ident'. The closed-loop transfer functions were identified as,

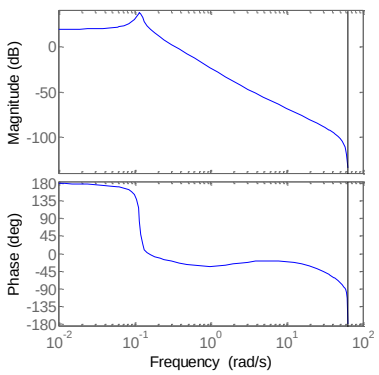
$$H(z) = \begin{bmatrix} H_{XX}(z) & H_{YX}(z) \\ H_{XY}(z) & H_{YY}(z) \end{bmatrix},$$

where

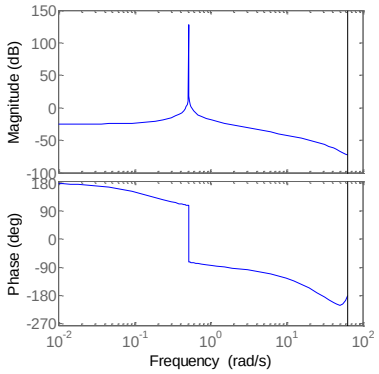
$$H_{XX}(z) = \frac{0.002838z^{-1} - 9.966e^{-05}z^{-2} - 0.002576z^{-3}}{1 - 2.782z^{-1} + 2.576z^{-2} - 0.7935z^{-3}}$$
$$H_{YX}(z) = \frac{-4.909e-05z^{-1} - 4.351e-06z^{-2} + 4.635e-05z^{-3}}{1 - 2.977z^{-1} + 2.953z^{-2} - 0.9766z^{-3}}$$
$$H_{XY}(z) = \frac{0.00178z^{-1} - 0.0006096z^{-2} - 0.001196z^{-3}}{1 - 2.294z^{-1} + 1.589z^{-2} - 0.2945z^{-3}}$$



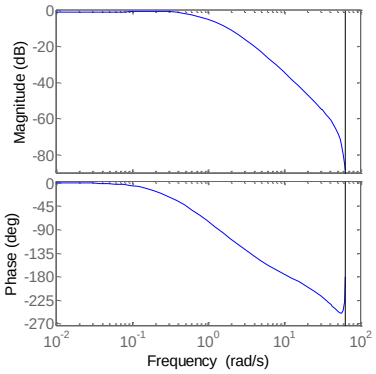
(a) Bode plot of H_{XX}



(b) Bode plot of H_{YX}



(c) Bode plot of H_{XY}



(d) Bode plot of H_{YY}

Figure A.4: Bode plots of plant model H.

$$H_{YY}(z) = \frac{0.002374z^{-1} - 0.0001206z^{-2} - 0.002227z^{-3}}{1 - 2.829z^{-1} + 2.664z^{-2} - 0.8348z^{-3}}.$$

Bode plots for the transfer functions are shown in Fig. A.4. Running the simulations with the full plant model results in relatively small differences as compared to a simplified SISO model that ignores the off-axis transfer functions. As such, for simplicity we utilize the dynamically decoupled, discrete-time closed-loop plant model for each dominant axis given as,

$$H_X(z) = H_{XX}(z)$$

$$H_Y(z) = H_{YY}(z).$$

APPENDIX B

Key manuscript derivations

In this appendix, we provide the derivations of the converged control and error signals from Subsection 4.3.3.

Error convergence: Classification II

By applying $\mathbf{e}_j = \mathbf{g}_d - \mathbf{H}\mathbf{u}_j$ and $\boldsymbol{\varepsilon}_{i,j} = \mathbf{z}_{i,d} - \mathbf{X}_i\mathbf{H}\mathbf{u}_j$ for classification II, the update law (4.6) is rearranged as,

$$\begin{aligned}
\mathbf{u}_{j+1} &= \mathbf{L}_u\mathbf{u}_j + \mathbf{L}_e\mathbf{e}_j + \sum_i \mathbf{L}_i\boldsymbol{\varepsilon}_{i,j} \\
&= \mathbf{L}_u\mathbf{u}_j + \mathbf{L}_e(\mathbf{g}_d - \mathbf{H}\mathbf{u}_j) + \sum_i \mathbf{L}_i(\mathbf{z}_{i,d} - \mathbf{X}_i\mathbf{H}\mathbf{u}_j) \\
&= (\mathbf{L}_u - \mathbf{L}_e\mathbf{H} - \sum_i \mathbf{L}_i\mathbf{X}_i\mathbf{H})\mathbf{u}_j + \mathbf{L}_e\mathbf{g}_d + \sum_i \mathbf{L}_i\mathbf{z}_{i,d}.
\end{aligned}$$

If $(\mathbf{I} - \mathbf{L}_u + \mathbf{L}_e\mathbf{H} + \sum_i \mathbf{L}_i\mathbf{X}_i\mathbf{H})$ is nonsingular, the steady state control signal can be found as,

$$\mathbf{u}_\infty = (\mathbf{I} - \mathbf{L}_u + \mathbf{L}_e\mathbf{H} + \sum_i \mathbf{L}_i\mathbf{X}_i\mathbf{H})^{-1}(\mathbf{L}_e\mathbf{g}_d + \sum_i \mathbf{L}_i\mathbf{z}_{i,d}).$$

By implementing $\mathbf{u}_\infty$ into $\mathbf{\Psi e}_\infty = \mathbf{\Psi}(\mathbf{g}_d - \mathbf{H u}_\infty)$ and applying the learning filters for classification II (4.9)-(4.11),

$$\begin{aligned}\mathbf{\Psi e}_\infty = &\Big[\mathbf{I} - \mathbf{\Psi H}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}(\mathbf{\Psi H})^T\mathbf{Q}\Big](\mathbf{\Psi g}_d) \\ &-\mathbf{\Psi H}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i\mathbf{z}_{i,d}.\end{aligned} \tag{B.1}$$

The 2-norm of the error from (B.1) is bounded as,

$$\begin{aligned}&\left\|\mathbf{\Psi e}_\infty\right\|_2 \\ &\leq \left\|\mathbf{\Psi H}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\{\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i(\mathbf{X}_i\mathbf{H})+\mathbf{S}\}(\mathbf{\Psi H})^{-1}(\mathbf{\Psi g}_d)\right\|_2 \\ &\quad + \left\|\mathbf{\Psi H}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\sum_i(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i\mathbf{z}_{i,d}\right\|_2 \\ &\triangleq \frac{\left(\sum_i w_i\bar{\sigma}(\mathbf{X}_i\mathbf{H})^2+s\right)\bar{\sigma}(\mathbf{\Psi g}_d)+\bar{\sigma}(\mathbf{\Psi H})\sum_i w_i\bar{\sigma}(\mathbf{X}_i\mathbf{H})\bar{\sigma}(\mathbf{z}_{i,d})}{\underline{\sigma}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H})+s} \\ &\triangleq \frac{f(w_i)+s}{f(q,w_i)+s}\bar{\sigma}(\mathbf{\Psi g}_d)+\boldsymbol{\epsilon}.\end{aligned}$$

By implementing $\mathbf{u}_\infty$ into $\boldsymbol{\varepsilon}_{i,\infty} = \mathbf{z}_{i,d} - \mathbf{X}_i\mathbf{H u}_\infty$ and applying the learning filters for classification II (4.9)-(4.11),

$$\begin{aligned}\boldsymbol{\varepsilon}_{i,\infty} = &\Big[\mathbf{I} - \mathbf{X}_i\mathbf{H}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}(\mathbf{X}_i\mathbf{H})^T\mathbf{W}_i\Big]\mathbf{z}_{i,d} \\ &-\mathbf{X}_i\mathbf{H}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}(\mathbf{\Psi H})^T\bar{\mathbf{Q}}(\mathbf{\Psi g}_d) \\ &-\mathbf{X}_i\mathbf{H}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\sum_k(\mathbf{X}_k\mathbf{H})^T\mathbf{W}_k\mathbf{z}_{k,d}.\end{aligned} \tag{B.2}$$

The 2-norm of the error for the additional performance metrics (B.2) is bounded as,

$$\begin{aligned}&\left\|\boldsymbol{\varepsilon}_{i,\infty}\right\|_2 \\ &\leq \left\|\mathbf{X}_i\mathbf{H}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\{(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi H})+\mathbf{S}\}(\mathbf{X}_i\mathbf{H})^{-1}\mathbf{z}_{i,d}\right\|_2 \\ &\quad + \left\|\mathbf{X}_i\mathbf{H}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}(\mathbf{\Psi H})^T\mathbf{Q}(\mathbf{\Psi g}_d)\right\|_2\end{aligned}$$

$$
\begin{aligned}
&+\left\|\mathbf{X}_i\mathbf{H}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\sum_k(\mathbf{X}_k\mathbf{H})^T\mathbf{W}_k\mathbf{z}_{k,d}\right\|_2\\
&\triangleq \frac{q\bar{\sigma}(\mathbf{\Psi H})\bar{\sigma}(\mathbf{X}_i\mathbf{H})\bar{\sigma}(\mathbf{\Psi g}_d)}{\underline{\sigma}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H})+s}\\
&\quad+\frac{\left(q\bar{\sigma}(\mathbf{\Psi H})^2+s\right)\bar{\sigma}(\mathbf{z}_{i,d})+\bar{\sigma}(\mathbf{X}_i\mathbf{H})\sum_k w_k\bar{\sigma}(\mathbf{X}_k\mathbf{H})\bar{\sigma}(\mathbf{z}_{k,d})}{\underline{\sigma}(\mathbf{H}^T\bar{\mathbf{Q}}\mathbf{H})+s}\\
&\triangleq \frac{f(q)}{f(q,w_i)+s}\bar{\sigma}(\mathbf{\Psi g}_d)+\boldsymbol{\epsilon_1}+\boldsymbol{\epsilon_2}.
\end{aligned}
$$

Classification III

By applying $\mathbf{e}_j=\mathbf{g}_d-\mathbf{H}\mathbf{u}_j$ and $\boldsymbol{\varepsilon}_{i,j}=\mathbf{z}_{i,d}-\mathbf{T}_i\mathbf{u}_j$ for classification III, the update law (4.6) is rearranged as,

$$
\begin{aligned}
\mathbf{u}_{j+1}&=\mathbf{L}_u\mathbf{u}_j+\mathbf{L}_e\mathbf{e}_j+\sum_i\mathbf{L}_i\boldsymbol{\varepsilon}_{i,j}\\
&=\mathbf{L}_u\mathbf{u}_j+\mathbf{L}_e(\mathbf{g}_d-\mathbf{H}\mathbf{u}_j)+\sum_i\mathbf{L}_i(\mathbf{z}_{i,d}-\mathbf{T}_i\mathbf{u}_j)\\
&=(\mathbf{L}_u-\mathbf{L}_e\mathbf{H}-\sum_i\mathbf{L}_i\mathbf{T}_i)\mathbf{u}_j+\mathbf{L}_e\mathbf{g}_d+\sum_i\mathbf{L}_i\mathbf{z}_{i,d}.
\end{aligned}
$$

If $(\mathbf{I}-\mathbf{L}_u+\mathbf{L}_e\mathbf{H}+\sum_i\mathbf{L}_i\mathbf{T}_i)$ is nonsingular, the steady state control signal can be found as,

$$
\mathbf{u}_\infty=(\mathbf{I}-\mathbf{L}_u+\mathbf{L}_e\mathbf{H}+\sum_i\mathbf{L}_i\mathbf{T}_i)^{-1}(\mathbf{L}_e\mathbf{g}_d+\sum_i\mathbf{L}_i\mathbf{z}_{i,d}).
$$

By implementing $\mathbf{u}_\infty$ into $\mathbf{\Psi e}_\infty=\mathbf{\Psi}(\mathbf{g}_d-\mathbf{H}\mathbf{u}_\infty)$ and applying the learning filters for classification II (4.15)-(4.17),

$$
\begin{aligned}
\mathbf{\Psi e}_\infty=&\left[\mathbf{I}-\mathbf{\Psi H}(\mathbf{H}^T\tilde{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}(\mathbf{\Psi H})^T\mathbf{Q}\right](\mathbf{\Psi g}_d)\\
&-\mathbf{\Psi H}(\mathbf{H}^T\tilde{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\sum_i\mathbf{T}_i^T\mathbf{W}_i\mathbf{z}_{i,d}.
\end{aligned}
\tag{B.3}
$$

The 2-norm of the error signal from (B.3) is bounded as,

$$
\begin{aligned}
&\left\|\boldsymbol{\Psi}\mathbf{e}_{\infty}\right\|_2 \\
&\quad\leq\left\|\boldsymbol{\Psi}\mathbf{H}(\mathbf{H}^T\tilde{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\{\sum_i \mathbf{T}_i^T\mathbf{W}_i\mathbf{T}_i+\mathbf{S}\}(\boldsymbol{\Psi}\mathbf{H})^{-1}(\boldsymbol{\Psi}\mathbf{g}_d)\right\|_2 \\
&\quad+\left\|\boldsymbol{\Psi}\mathbf{H}(\mathbf{H}^T\tilde{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\sum_i \mathbf{T}_i^T\mathbf{W}_i\mathbf{z}_{i,d}\right\|_2 \\
&\quad\triangleq\frac{\left(\sum_i w_i\bar{\sigma}(\mathbf{T}_i)^2+s\right)\bar{\sigma}(\boldsymbol{\Psi}\mathbf{g}_d)+\bar{\sigma}(\boldsymbol{\Psi}\mathbf{H})\sum_i w_i\bar{\sigma}(\mathbf{T}_i)\bar{\sigma}(\mathbf{z}_{i,d})}{\underline{\sigma}(\mathbf{H}^{\mathbf{T}}\tilde{\mathbf{Q}}\mathbf{H})+s} \\
&\quad\triangleq\frac{f(w_i)+s}{f(q,w_i)+s}\bar{\sigma}(\boldsymbol{\Psi}\mathbf{g}_d)+\boldsymbol{\epsilon}.
\end{aligned}
$$

By implementing $\mathbf{u}_\infty$ into $\boldsymbol{\varepsilon}_{i,\infty}=\mathbf{z}_{i,d}-\mathbf{T}_i\mathbf{u}_\infty$ and applying the learning filters for classification III (4.15)-(4.17),

$$
\begin{aligned}
\boldsymbol{\varepsilon}_{i,\infty}=&\left[\mathbf{I}-\mathbf{T}_i(\mathbf{H}^T\tilde{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\mathbf{T}_i^T\mathbf{W}_i\right]\mathbf{z}_{i,d} \\
&-\mathbf{T}_i(\mathbf{H}^T\tilde{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{g}_d) \\
&-\mathbf{T}_i(\mathbf{H}^T\tilde{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\sum_k \mathbf{T}_k^T\mathbf{W}_k\mathbf{z}_{k,d}.
\end{aligned}
\tag{B.4}
$$

The 2-norm of the error signal for the additional performance metrics (B.4) is bounded as,

$$
\begin{aligned}
&\left\|\boldsymbol{\varepsilon}_{i,\infty}\right\|_2 \\
&\quad\leq\left\|\mathbf{T}_i(\mathbf{H}^T\tilde{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\{(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{H})+\mathbf{S}\}\mathbf{T}_i^{-1}\mathbf{z}_{i,d}\right\|_2 \\
&\quad+\left\|\mathbf{T}_i(\mathbf{H}^T\tilde{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}(\boldsymbol{\Psi}\mathbf{H})^T\mathbf{Q}(\boldsymbol{\Psi}\mathbf{g}_d)\right\|_2 \\
&\quad+\left\|\mathbf{T}_i(\mathbf{H}^T\tilde{\mathbf{Q}}\mathbf{H}+\mathbf{S})^{-1}\sum_k \mathbf{T}_k^T\mathbf{W}_k\mathbf{z}_{k,d}\right\|_2 \\
&\quad\triangleq\frac{q\bar{\sigma}(\boldsymbol{\Psi}\mathbf{H})\bar{\sigma}(\mathbf{T}_i)\bar{\sigma}(\boldsymbol{\Psi}\mathbf{g}_d)}{\underline{\sigma}(\mathbf{H}^{\mathbf{T}}\tilde{\mathbf{Q}}\mathbf{H})+s} \\
&\qquad+\frac{\left(q\bar{\sigma}(\boldsymbol{\Psi}\mathbf{H})^2+s\right)\bar{\sigma}(\mathbf{z}_{i,d})+\bar{\sigma}(\mathbf{T}_i)\sum_k w_k\bar{\sigma}(\mathbf{T}_k)\bar{\sigma}(\mathbf{z}_{k,d})}{\underline{\sigma}(\mathbf{H}^{\mathbf{T}}\tilde{\mathbf{Q}}\mathbf{H})+s} \\
&\quad\triangleq\frac{f(q)}{f(q,w_i)+s}\bar{\sigma}(\boldsymbol{\Psi}\mathbf{g}_d)+\boldsymbol{\epsilon_1}+\boldsymbol{\epsilon_2},
\end{aligned}
$$

where $i = 1, 2, \ldots P$, and $k = 1, 2, \ldots, i-1, i+1, i+2, \ldots P$.

APPENDIX C

Examples of contrast matrices

In this appendix, we provide examples of the $\mathbf{C}_b$ and $\mathbf{C}_s$ matrices from subsection 4.4.2.2 for the example image structure provided in Fig. 4.6.

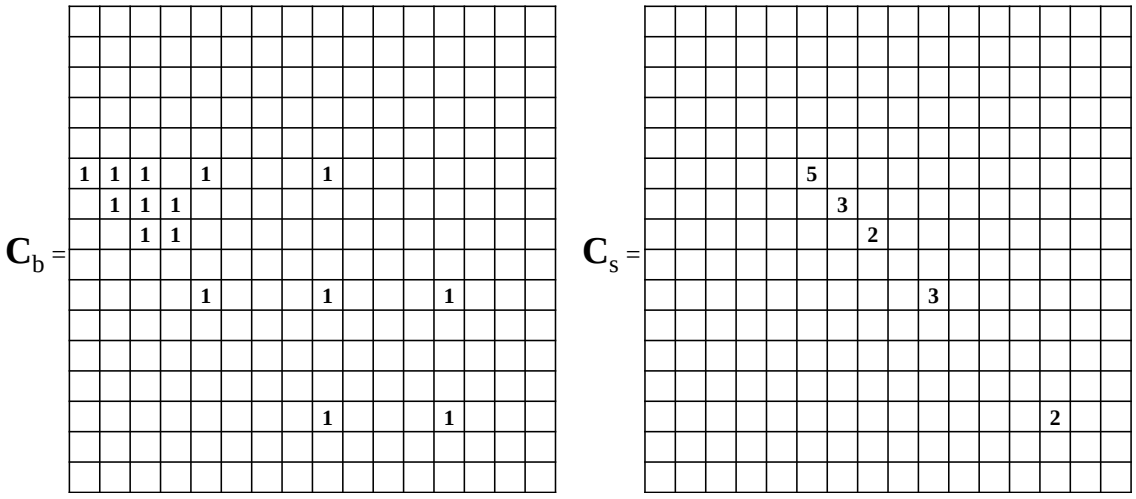


Figure C.1: Example $\mathbf{C}_b$ and $\mathbf{C}_s$ matrices.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *Control Systems Magazine, IEEE*, vol. 26, no. 3, pp. 96–114, 2006.

[2] K. L. Moore, *Iterative learning control for deterministic systems*. Springer, 1993.

[3] K. J. Hunt, D. Sbarbaro, R. Żbikowski, and P. J. Gawthrop, "Neural networks for control systemsa survey," *Automatica*, vol. 28, no. 6, pp. 1083–1112, 1992.

[4] G. Hillerström and K. Walgama, "Repetitive control theory and applications-a survey," in *Proceedings of the 13th IFAC World Congress*, vol. 500, pp. 1–6, 1996.

[5] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *International Journal of Control*, vol. 73, no. 10, pp. 930–954, 2000.

[6] D. Kim and S. Kim, "An iterative learning control method with application for cnc machine tools," in *Industry Applications Society Annual Meeting, 1993., Conference Record of the 1993 IEEE*, pp. 2106 –2111 vol.3, oct 1993.

[7] I. Rotariu, M. Steinbuch, and R. Ellenbroek, "Adaptive iterative learning control for high precision motion systems," *Control Systems Technology, IEEE Transactions on*, vol. 16, no. 5, pp. 1075–1082, 2008.

[8] K. Barton and A. Alleyne, "A norm optimal approach to time-varying ilc with application to a multi-axis robotic testbed," *Control Systems Technology, IEEE Transactions on*, vol. 19, no. 1, pp. 166–180, 2011.

[9] M. Mezghani, G. Roux, M. Cabassud, M. Le Lann, B. Dahhou, and G. Casamatta, "Application of iterative learning control to an exothermic semi-batch chemical reactor," *Control Systems Technology, IEEE Transactions on*, vol. 10, no. 6, pp. 822–834, 2002.

[10] J. H. Lee and K. S. Lee, "Iterative learning control applied to batch processes: An overview," *Control Engineering Practice*, vol. 15, no. 10, pp. 1306–1318, 2007.

[11] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic Systems*, vol. 1, pp. 123–140, 1984.

[12] A. Tayebi and S. Islam, "Adaptive Iterative Learning Control for robot manipulators: Experimental results," *Control Engineering Practice*, vol. 14, pp. 843–851, 2006.

[13] I. Lim and K. L. Barton, "Pareto iterative learning control: Optimized control for multiple performance objectives," *Control Engineering Practice*, vol. 26, pp. 125–135, 2014.

[14] B. Dijkstra, N. Rambaratsingh, C. Scherer, O. Bosgra, M. Steinbuch, and S. Kerssemakers, "Input design for optimal discrete time point-to-point motion of an industrial xy-positioning table," *Selected Topics in Signals, Systems and Control*, vol. 12, 2001.

[15] T. Samad, J. S. Bay, and D. Godbole, "Network-centric systems for military operations in urban terrain: the role of UAVs," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 92–107, 2007.

[16] W. Lee, H. Bang, and H. Leeghim, "Cooperative localization between small UAVs using a combination of heterogeneous sensors," *Aerospace Science and Technology*, vol. 27, no. 1, pp. 105–111, 2013.

[17] K. Krishnamoorthy, D. Casbeer, P. Chandler, M. Pachter, and S. Darbha, "UAV search & capture of a moving ground target under delayed information," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pp. 3092–3097, IEEE, 2012.

[18] K. Barton and D. Kingston, "Systematic surveillance for UAVs: A feedforward iterative learning control approach," in *American Control Conference (ACC), 2013*, pp. 5917–5922, IEEE, 2013.

[19] K. L. Moore, Y. Chen, and H.-S. Ahn, "Iterative learning control: A tutorial and big picture view," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 2352–2357, 2006.

[20] Aerotech, "Aerotech @ONLINE."

[21] L. Hladowski, Z. Cai, K. Galkowski, E. Rogers, C. T. Freeman, P. L. Lewin, and W. Paszke, "Repetitive process based iterative learning control designed by lmis and experimentally verified on a gantry robot," in *American Control Conference, 2009. ACC'09.*, pp. 949–954, IEEE, 2009.

[22] D. Bristow, K. Barton, and A. Alleyne, *The Control Handbook*, ch. Iterative Learning Control. CRC Press, Inc, 2010.

[23] N. Amann, D. H. Owens, and E. Rogers, "Iterative learning control for discrete-time systems with exponential rate of convergence," *IEE Proceedings-Control Theory and Applications*, vol. 143, no. 2, pp. 217–224, 1996.

[24] C. Freeman, A.-M. Hughes, J. Burridge, P. Chappell, P. Lewin, and E. Rogers, "Iterative learning control of FES applied to the upper extremity for rehabilitation," *Control Engineering Practice*, vol. 17, no. 3, pp. 368 – 381, 2009.

[25] S. Lim and H. Bang, "Waypoint planning algorithm using cost functions for surveillance," *International Journal of Aeronautical and Space Sciences*, vol. 11, pp. 136–144, 2010.

[26] C. T. Freeman, Z. Cai, E. Rogers, and P. L. Lewin, "Iterative Learning Control for Multiple Point-to-Point Tracking Application," *IEEE Transactions on Control Systems Technology*, vol. 19, pp. 590–600, 2011.

[27] GM, "Gm @ONLINE."

[28] K. L. Barton, D. J. Hoelzle, A. G. Alleyne, and A. J. W. Johnson, "Cross-coupled iterative learning control of systems with dissimilar dynamics: design and implementation," *International Journal of Control*, vol. 84, no. 7, pp. 1223–1233, 2011.

[29] I. Lim and K. L. Barton, "Pareto optimization-based iterative learning control," in *American Control Conference (ACC), 2013*, pp. 5171–5176, IEEE, 2013.

[30] K. Barton and D. Kingston, "Systematic surveillance for UAVs: A feedforward iterative learning control approach," in *American Control Conference (ACC), 2013*, pp. 5917–5922, IEEE, 2013.

[31] P. Yang and F. Catthoor, "Pareto-optimization-based run-time task scheduling for embedded systems," in *Hardware/Software Codesign and System Synthesis, 2003. First IEEE/ACM/IFIP International Conference on*, pp. 120 –125, oct. 2003.

[32] B. Boulet and Y. Duan, "The fundamental tradeoff between performance and robustness - a new perspective on loop shaping - classic control revisited part ii," *Control Systems, IEEE*, vol. 27, pp. 30 –44, june 2007.

[33] Y. Jin and B. Sendhoff, "Trade-off between performance and robustness: An evolutionary multiobjective approach," in *Proceedings of Second International Conference on Evolutionary Multi-criteria Optimization. LNCS 2632*, pp. 237–251, Springer, 2003.

[34] S. Mishra, U. Topcu, and M. Tomizuka, "Optimization-based constrained iterative learning control," *Control Systems Technology, IEEE Transactions on*, vol. 19, no. 6, 2011.

[35] C. T. Freeman, "Constrained point-to-point iterative learning control with experimental verification," *Control Engineering Practice*, vol. 20, no. 5, pp. 489–498, 2012.

[36] C. Freeman and Y. Tan, "Iterative learning control with mixed constraints for point-to-point tracking," *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 3, pp. 604–616, 2013.

[37] D. H. Owens, C. T. Freeman, and B. Chu, "Multivariable norm optimal iterative learning control with auxiliary optimisation," *International Journal of Control*, vol. 86, no. 6, pp. 1026–1045, 2013.

[38] I. Lim, K. L. Barton, and D. Hoelzle, "Spatial ilc for multi-objective systems," in *Dynamic System and Control Conference (DSCC), 2014*, ASME, 2014.

[39] J. Cesarano, J. G. Dellinger, M. P. Saavedra, D. D. Gill, R. D. Jamison, B. A. Grosser, J. M. Sinn-Hanlon, and M. S. Goldwasser, "Customization of load-bearing hydroxyapatite lattice scaffolds," *International Journal of Applied Ceramic Technology*, vol. 2, no. 3, pp. 212–220, 2005.

[40] R. Brooks, "GE aviation takes on additive manufacturing @ONLINE," 2012.

[41] M. Larsson, "Printed and flexible electronics @ONLINE."

[42] D. J. Hoelzle and K. L. Barton, "A new spatial iterative learning control approach for improved micro-additive manufacturing," in *American Control Conference (ACC), 2014*, pp. 1805–1810, June 2014.

[43] D. A. Bristow and A. G. Alleyne, "A high precision motion control system with application to microscale robotic deposition," *Control Systems Technology, IEEE Transactions on*, vol. 14, no. 6, pp. 1008–1020, 2006.

[44] S. Sahoo, S. Panda, and J. Xu, "Application of spatial iterative learning control for direct torque control of switched reluctance motor drive," in *Power Engineering Society General Meeting, 2007. IEEE*, pp. 1–7, IEEE, 2007.

[45] B. Cichy, K. Gałkowski, and E. Rogers, "Iterative learning control for spatio-temporal dynamics using crank-nicholson discretization," *Multidimensional Systems and Signal Processing*, vol. 23, no. 1-2, pp. 185–208, 2012.

[46] K. Barton and A. Alleyne, "A cross-coupled iterative learning control design for precision motion control," *Control Systems Technology, IEEE Transactions on*, vol. 16, no. 6, pp. 1218–1231, 2008.

[47] K. L. Moore, M. Ghosh, and Y. Q. Chen, "Spatial-based iterative learning control for motion control applications," *Meccanica*, vol. 42, no. 2, pp. 167–175, 2007.

[48] K. L. Moore and Y. Q. Chen, "Iterative learning control approach to a diffusion control problem in an irrigation application," in *Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on*, pp. 1329–1334, IEEE, 2006.

[49] D. J. Hoelzle and K. L. Barton, "On spatial iterative learning control via two dimensional convolution: Stability analysis and computational efficiency," *Control Systems Technology, IEEE Transactions on, accepted*, 2015.

[50] I. Lim, K. L. Barton, and D. Hoelzle, "Spatial ILC for multi-objective systems," in *Dynamic System and Control Conference (DSCC), 2014*, ASME, 2014.

[51] I. Lim, K. L. Barton, and D. Hoelzle, "A multi-objective iterative learning control approach for additive manufacturing applications," *Control Engineering Practice, submitted*, 2015.

[52] D. Bristow and B. Hencey, "A q, l factorization of norm-optimal Iterative Learning Control," in *Proc. of IEEE Conference on Decision and Control*, pp. 2380–2384, 2008.

[53] J. van de Wijdeven and O. Bosgra, "Residual vibration suppression using hankel iterative learning control," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 10, pp. 1034–1051, 2008.

[54] D. Bristow, K. Barton, and A. Alleyne, *The Control Handbook, Second Edition: Iterative Learning Control*. CRC Press, Inc., 2010.

[55] M. Phan and R. Longman, "A mathematical theory of learning control for linear discrete multivariable systems," in *Prod. of the AIAA/AAS Astrodynamics Specialist Conference*, pp. 740–746, 1988.

[56] G. Golub and C. Van Loan, *Matrix Computations*, ch. 2. Matrix Analysis. Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 2013.

[57] S. Lee, Y. Cho, M. Hwang-Bo, B. You, and S. Oh, "A stable target-tracking control for unicycle mobile robots," in *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 3, pp. 1822–1827, IEEE, 2000.

[58] T. Lee, K. Song, C. Lee, and C. Teng, "Tracking control of unicycle-modeled mobile robots using a saturation feedback controller," *Control Systems Technology, IEEE Transactions on*, vol. 9, no. 2, pp. 305–318, 2001.

[59] O. Castillo, L. Aguilar, and S. Cardenas, "Fuzzy logic tracking control for unicycle mobile robots," *Engineering Letters*, vol. 13, no. 2, 2006.

[60] G. Mester, "Motion control of wheeled mobile robots," in *4th Serbian-Hungarian Joint Symposium on Intelligent Systems, SISY*, pp. 119–130, 2006.

[61] G. Antonelli, S. Chiaverini, and G. Fusco, "A fuzzy-logic-based approach for mobile robot path tracking," *Fuzzy Systems, IEEE Transactions on*, vol. 15, no. 2, pp. 211–221, 2007.

[62] F. Martins, W. Celeste, R. Carelli, M. Sarcinelli-Filho, and T. Bastos-Filho, "An adaptive dynamic controller for autonomous mobile robot trajectory tracking," *Control Engineering Practice*, vol. 16, no. 11, pp. 1354–1363, 2008.

[63] P. Morin and C. Samson, "Motion control of wheeled mobile robots," *Springer Handbook of Robotics*, pp. 799–826, 2008.

[64] R. C. Gonzalez and P. Wintz, *Digital image processing, 1987*. Addison-Wesley, Reading, MA.

[65] G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3. JHU Press, 2012.

[66] R. M. Gray, *Toeplitz and circulant matrices: A review*. now publishers inc, 2006.

[67] G. E. Trapp, "Inverses of circulant matrices and block circulant matrices," *Kyungpook Mathematical Journal*, vol. 13, no. 1, pp. 11–20, 1973.

[68] M. Norrlöf and S. Gunnarsson, "Time and frequency domain convergence properties in iterative learning control," *International Journal of Control*, vol. 75, no. 14, pp. 1114–1126, 2002.

[69] T. Donkers, J. van de Wijdeven, and O. Bosgra, "Robustness against model uncertainties of norm optimal iterative learning control," in *American Control Conference, 2008*, pp. 4561–4566, IEEE, 2008.

[70] J.-U. Park, M. Hardy, S. J. Kang, K. Barton, K. Adair, D. kishore Mukhopadhyay, C. Y. Lee, M. S. Strano, A. G. Alleyne, J. G. Georgiadis, *et al.*, "High-resolution electrohydrodynamic jet printing," *Nature materials*, vol. 6, no. 10, pp. 782–789, 2007.

[71] K. Barton, S. Mishra, K. Alex Shorter, A. Alleyne, P. Ferreira, and J. Rogers, "A desktop electrohydrodynamic jet printing system," *Mechatronics*, vol. 20, no. 5, pp. 611–616, 2010.

[72] K. Barton, S. Mishra, A. Alleyne, P. Ferreira, and J. Rogers, "Control of high-resolution electrohydrodynamic jet printing," *Control Engineering Practice*, vol. 19, no. 11, pp. 1266–1273, 2011.

[73] D.-Y. Lee, J.-C. Lee, Y.-S. Shin, S.-E. Park, T.-U. Yu, Y.-J. Kim, and J. Hwang, "Structuring of conductive silver line by electrohydrodynamic jet printing and its electrical characterization," in *Journal of physics: conference series*, vol. 142, p. 012039, IOP Publishing, 2008.

[74] J.-U. Park, S. Lee, S. Unarunotai, Y. Sun, S. Dunham, T. Song, P. M. Ferreira, A. G. Alleyene, U. Paik, and J. A. Rogers, "Nanoscale, electrified liquid jets for high-resolution printing of charge," *Nano letters*, vol. 10, no. 2, pp. 584–591, 2010.

[75] J.-U. Park, J. H. Lee, U. Paik, Y. Lu, and J. A. Rogers, "Nanoscale patterns of oligonucleotides formed by electrohydrodynamic jet printing with applications in biosensing and nanomaterials assembly," *Nano letters*, vol. 8, no. 12, pp. 4210–4216, 2008.

[76] Z. Wang and A. C. Bovik, "A universal image quality index," *Signal Processing Letters, IEEE*, vol. 9, no. 3, pp. 81–84, 2002.

[77] C. Pannier, "Unpublished data," *University of Michigan, Ann Arbor, MI*, 2015.

[78] K. L. Moore, "Multi-loop control approach to designing iterative learning controllers," in *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, vol. 1, pp. 666–671, IEEE, 1998.

[79] K. L. Moore, "An iterative learning control algorithm for systems with measurement noise," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 1, pp. 270–275, IEEE, 1999.

[80] D. Owens, E. Rogers, and K. Moore, "Analysis of linear iterative learning control schemes using repetitive process theory," *Asian Journal of Control*, vol. 4, no. 1, pp. 68–89, 2002.

[81] D. Kingston and D. Casbeer, "Cooperative UAV/UGS scenario," 2011.

[82] K. Kalyanam, S. Dharba, M. Pachter, and P. Chandler, "Optimal search for a moving grouond target," 2012.