Iterative learning control with multiple pass points: Algorithms and applications

Tong Duy Son

School of Information and Mechatronics

Gwangju Institute of Science and Technology

2012
Iterative learning control with multiple pass points: Algorithms and applications

Advisor: Hyo-Sung Ahn

by

Tong Duy Son

School of Information and Mechatronics
Gwangju Institute of Science and Technology

A thesis submitted to the faculty of the Gwangju Institute of Science and Technology in partial fulfillment of the requirements for the degree of Master of Science in the School of Information and Mechatronics

Gwangju, Republic of Korea
December 5, 2011
Approved by
Professor Hyo-Sung Ahn
Thesis Advisor
Iterative learning control with multiple pass points: Algorithms and applications

Tong Duy Son

Accepted in partial fulfillment of the requirements for the degree of Master of Science

December 5, 2011

Thesis Advisor
Prof. Hyo-Sung Ahn

Committee Member
Prof. Jongho Lee

Committee Member
Prof. Hyuk-Sang Kwon
Dedicated to

my family.
Abstract

In this thesis, we present new iterative learning control (ILC) frameworks for multiple points tracking problems. Conventionally, one demand prior to designing ILC systems for such problems is to plan a reference trajectory that passes through all given points at given times. After that, an ILC controller is generated to follow this trajectory in the iteration domain. These works do not follow that direction. First, we present an algorithm in which not only control signal but also reference trajectory is updated at each trial. The scheme investigates the relationship between reference trajectory and ILC tracking control for improving system performance, particularly rate of convergence. Moreover, we show that it is possible to make benefits from cooperating between trajectory planning and tracking control stages. Second, a new ILC scheme is proposed to produce output curves that pass close to the desired points without considering the reference trajectory. Here, the control signals are generated by solving an optimal ILC problem with respect to the points. As such, the whole process becomes simpler; key advantages include significantly decreasing the computational cost and improving performance. Other contributions include multiple points ILC with input constraints and disturbances, uncertainties in data points which happen both repetitively and non-repetitively. Our works are then examined in numerical examples and applications.
# Contents

Abstract (English) i  
List of Contents ii  
List of Figures v  

1 Introduction 1  
  1.1 Introduction ............................................. 1  
  1.2 Multiple Points Tracking Control Problem ................. 3  
  1.3 Background ............................................. 4  
    1.3.1 Trajectory Planning ................................. 4  
    1.3.2 Trajectory Tracking Control using ILC ............... 5  
  1.4 Motivation and Contribution .............................. 7  
  1.5 Organization of the Thesis .............................. 9  

2 Terminal Iterative Learning Control 11  
  2.1 Introduction ............................................. 11  
  2.2 Background ............................................. 12  
  2.3 Initial Iterative Learning for Single Terminal Point ......... 13  
  2.4 ILC for Multiple Intermediate Pass Points .................. 17  
    2.4.1 Initial Learning .................................. 17  
    2.4.2 Iterative Learning using Continuous Control Input .. 19  
  2.5 Simulation ............................................. 22  
  2.6 Conclusion .............................................. 23  

3 Interpolation-Based Approach for Multiple Terminal Iterative Learning Control 29  
  3.1 Introduction ............................................. 29  
  3.2 Background ............................................. 30  
  3.3 Reference Trajectory Learning Update ....................... 32  
    3.3.1 Analysis ....................................... 33  
    3.3.2 Reference Trajectory Update Law .................... 33  
  3.4 Controller .............................................. 35
3.5 Simulation .................................................. 39
3.6 Conclusion .................................................. 40

4 Optimal Iterative Learning Control Without Tracking a Reference Trajectory 43
4.1 Introduction .................................................. 43
4.2 Optimal ILC For Continuous Systems ......................... 43
  4.2.1 ILC Controller ........................................ 46
  4.2.2 Convergence ........................................... 47
  4.2.3 Control Performance .................................. 49
4.3 Optimal ILC for Discrete-time Systems ............................. 50
4.4 Input Constraints ......................................... 54
4.5 Simulation Results ........................................ 56

5 Multiple Interval Points Tracking 60
5.1 Introduction .................................................. 60
5.2 Background ................................................. 61
5.3 The Optimization Problem .................................. 62
  5.3.1 The Maximization Problem ......................... 62
  5.3.2 The Equivalent Minimization Problem .............. 64
5.4 ILC Algorithm ............................................. 66
  5.4.1 Optimized Parameters .............................. 67
  5.4.2 Optimized ILC and Convergence Properties .......... 68
5.5 Simulation Results ........................................ 70

6 Terminal Iterative Learning Control for Thermoforming Machine 74
6.1 Introduction and Motivations ................................ 74
6.2 Modeling .................................................... 76
6.3 Simulation ................................................... 83

7 Multiple Points Satellite Antenna Iterative Learning Control 88
7.1 Introduction and Motivations ................................ 88
7.2 Desired Azimuth and Elevation Angles Computation ........... 92
7.3 Antenna Dynamics ......................................... 96
7.4 Current Trajectory Planning and Control Method ............. 97
7.5 Simulations .................................................. 98

8 Conclusion and Future Work .......................... 104

References ..................................................... 106

Acknowledgements .......................................... 111
List of Figures

2.1 Initial learning for single terminal point ........................................... 25
2.2 Initial learning for multiple pass points ............................................ 26
2.3 Convergence of the points’s errors in tracking the reference trajectory .... 27
2.4 Convergence of the points’s errors in our approach ............................... 28
3.1 Errors at the given points ................................................................. 41
3.2 Updated trajectories and the outputs by trials ..................................... 42
4.1 Performance of the ILC controller without reference trajectory with $q = 10$, $r = 2$, and $s = 0.01$. ......................................................... 57
4.2 Comparison of control energy between the proposed controller and the conventional norm-optimal ILC with $q = 10$, $r = 2$, and $s = 0.01$. ............... 58
4.3 Performance with $q = 20$, $r = 5$, and $s = 0.5$. ................................. 59
5.1 Errors and control energy of the interval tracking problem .................... 72
5.2 Updated parameters $\lambda_k$ and $q_1$ ................................................. 73
6.1 A thermoforming machine [30] .......................................................... 76
6.3 IR temperature sensors and the corresponding zone [34] ................. 78
6.4 Layers and nodes [34] ................................................................. 78
6.5 Temperature errors ................................................................. 84
6.6 Heater setpoints ................................................................. 85

– v –
6.7 Surface temperature ................................................................. 86
6.8 Surface temperature ................................................................. 87

7.1 X-band antenna azimuth and elevation motion [35] ....................... 91
7.2 Satellite antenna pointing .......................................................... 91
7.3 Satellite, ground station, and direction in ECF coordinate frame [15] 93
7.4 Off-pointing margin [15] ............................................................. 93
7.5 X-band antenna mounted on satellite [15] ................................... 94
7.6 Errors at points ................................................................. 100

7.7 Control signals and output of angles in the iteration domain with $q = 10^4$, $r = 0.02$, and $s = 10^{-4}$ .......................................................... 101

7.8 Control signals and output of angles in the iteration domain with $q = 5 \times 10^4$, $r = 0.01$, and $s = 10^{-3}$ .............................. 102
Chapter 1

Introduction

1.1 Introduction

Iterative learning control (ILC) is a control scheme that refines the input sequences from trials in order to improve the performance of repetitive operation systems. The prime strategy of ILC algorithms is to update the control signal using the information measured in previous iterations. As such, the ILC controller achieves high tracking performance in the presence of systems with model uncertainty and repeatable disturbances. Another contribution of ILC theory is to investigate the repetitive nature of a system that operates repetitively. Since the ILC algorithm was initially proposed by Arimoto [1], there have been numerous publications and studies performed; a number of surveys and books have effectively covered the novel ideas and development of ILC methodology [2]-[7].

Terminal iterative learning control (TILC) is a type of ILC control technique that focuses solely on controlling the terminal point of a system such that this point only tracks the given desired output point. TILC was first presented for rapid thermal processing chemical vapor deposition in wafer fabrication industry applications [8], where the ultimate control objective is to control the deposition thickness at the end of the thermal processing cycle. In [9]-[10], the plastic sheet surface temperature control in thermoforming machine is controlled by tuning the temperature setpoint of the heater. In addition, there has been some research focused
on the initial state learning for final state control in both theoretical [11], and practical applications such as ballistic control, train stopping control [12] - [14]. The main approach of TILC in these studies has been to update control signals using the terminal tracking error alone, rather than the whole output trajectory tracking error. Moreover, these investigations have shown that the approach could achieve convergence in the iteration domain.

However, previous terminal ILC research has typically only considered the final point; the existence of intermediate pass points has not been studied well in ILC theory although many systems require multiple target points to be tracked. For example, satellite antennas need to be maneuvered to point toward a desired ground location, which is determined from the desired azimuth and elevation angles at given sampling times [15], and point to point control in robotic-assisted upper limb stroke rehabilitation applications [16]. Here, the common approach for this problem is tracking a reference trajectory which is designed to go through or close to the data points. In addition, multiple point-to-point tracking control was considered by developing a frequency-domain framework in which the reference is updated between trials in [17]. However, since the control problem is focused on tracking multiple points during system operation—instead of the whole trajectory at all time instants—the ILC approach, which tracks a reference trajectory, is undesirable with regards to control effort and energy. In addition, it requires further computational analysis to generate a reference trajectory from given terminal points. These reasons are the primary motivation for our development of a new and advance TILC framework for tracking multiple intermediate pass points.
1.2 Multiple Points Tracking Control Problem

The control schemes to tracking multiple pass points problems are generally divided into two steps: trajectory planning and tracking control. In these schemes, the trajectory planner attempts to generate an optimal reference trajectory from the given set of points in the motion profile; the main focus of research in this area pertains to interpolation techniques. On the other hand, the controller—which is designed to track the reference trajectory—focuses on the system dynamics against system uncertainties and disturbances. Here, the improved accuracy in trajectory tracking results has led to the development of various control schemes, such as feedback control, robust control, and iterative learning control. Our research focus specifically on the iterative learning control techniques.

To formulate the problem mathematically, let us first consider a linear continuous-time system that operates on an interval \( t \in [0, T] \) as

\[
\begin{align*}
\dot{x}_k(t) &= Ax_k(t) + Bu_k(t) + \omega_k(t) \\
y_k(t) &= Cx_k(t) + \nu_k(t)
\end{align*}
\]

(1.1)

where \( k \) is the iteration index, and the time index is \( t = 0, 1, 2, ..., T \). The system is a multi-input multi-output (MIMO) system that has state \( x_k(t) \in \mathbb{R}^p \), control signal \( u_k(t) \in \mathbb{R}^m \), and output \( y_k(t) \in \mathbb{R}^n \). The matrices \( A, B, \) and \( C \) have appropriate dimensions. In the system model, \( \omega_k(t) \) and \( \nu_k(t) \) are repeating model uncertainty and measurement disturbance, respectively; moreover, repeated non zero initial conditions can be also captured in \( \nu_k(t) \). The system is assumed to be both controllable and observable.

In the tracking problem, there are specified data points in the motion profile. The given
time instants in the system operation are defined as \( t_1, t_2, \ldots, t_M \), where \( 0 \leq t_1 < t_2 < \ldots < t_M \leq T \); while the desired outputs at these points are given by

\[
y_d(t_1), \ y_d(t_2), \ \ldots, \ y_d(t_M).
\]

As a result, the control task is to construct a control law that drives the system output goes through data points as

\[
y_k(t_i) = y_d(t_i),
\]

where \( i = 1, 2, \ldots, M \).

### 1.3 Background

In conventional tracking schemes, the trajectory planner generates a reference trajectory \( r(t) \) such that it passes the desired points at \( t_1, t_2, \ldots, t_M \). Consequently, an ILC controller is designed to improve tracking of the reference trajectory. This section provides background in both stages: trajectory planning and ILC trajectory tracking control.

#### 1.3.1 Trajectory Planning

In trajectory planning stage, the common technique to find a reference trajectory from the given points is interpolating splines, which is a tool in numerical analysis. The generated function is a spline if it exactly interpolates given points. The framework of interpolating splines deals with the problem:

\[
\min \left\{ \int_0^{N-1} r''(t)^2 \, dt \right\}, \text{subject to } r(t_i) = y_d(t_i).
\]
Here, \( r''(t) \) denotes the second derivative, and the integral term is introduced in order to get smooth function of the reference trajectory [21].

However, the reference trajectory may not be feasible for the system since it varies too fast to allow the dynamics to follow the trajectory; or it may vary too slowly that gives a conservative control performance. As a result, control theoretic interpolating splines was introduced [22]. The suggested approach addresses these problems by considering auxiliary system dynamics, rather than spline functions.

1.3.2 Trajectory Tracking Control using ILC

In the tracking control stage, an ILC controller is applied to drive the system output goes as close as possible to the reference trajectory. The learning algorithm utilizes output errors and control inputs from previous iterations to compute an updated control signal. In ILC research, most of works consider ILC specially discrete-time systems due to the fact that the practical implementation of ILC requires the storage of past iterations data, which is typically sampled. Here, we can reformulate the equivalent discrete-time system in the lifted system framework as

\[ y_k = Pu_k + d_k, \quad (1.4) \]

where input signal, output, reference trajectory and disturbance are written in the super-vector form as
\[ u_k = \begin{bmatrix} u_k^T(0) & u_k^T(1) & \ldots & u_k^T(N-1) \end{bmatrix}^T \]
\[ y_k = \begin{bmatrix} y_k^T(1) & y_k^T(2) & \ldots & y_k^T(N) \end{bmatrix}^T \]
\[ r = \begin{bmatrix} r^T(1) & r^T(2) & \ldots & r^T(N) \end{bmatrix}^T \]
\[ d_k = \begin{bmatrix} d_k^T(1) & d_k^T(2) & \ldots & d_k^T(N) \end{bmatrix}^T ; \]

and the system matrix \( P \) is represented by
\[
P = \begin{bmatrix}
  p_1 & 0 & \cdots & 0 \\
  p_2 & p_1 & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
  p_N & p_{N-1} & \cdots & p_1
\end{bmatrix}
\]

with \( p_i = CA_i^{-1}B \).

The general ILC algorithm is formulated by
\[
u_{k+1} = T_u u_k + T_e (r - y_k) . \tag{1.5}
\]

Here, the algorithm is convergent if \( \rho(T_u - T_e P) < 1 \), where \( \rho(A) \) is the spectral radius of the matrix \( A \). Moreover, the algorithm guarantees monotonic convergence if the condition \( \bar{\sigma}(T_u - T_e P) < 1 \), where \( \bar{\sigma}(A) \) is the largest singular value of the matrix \( A \), is satisfied.

From a practical perspective, the goal of ILC is to generate an open-loop signal that approximately inverts the plants dynamics to track the reference and reject repeating disturbances and uncertainties. ILC controller is an open-loop control and has no feedback mechanism to respond to unanticipated, nonrepeating disturbances; thus, a feedback controller in combination with ILC are usually applied. Generally, the PD-type learning and
norm-optimal ILC are considered as the most two popular ILC design techniques. In the first algorithm, which was developed from the Arimoto’s original work [1], the learning function consists of a proportional and derivative gain on the error. However, monotonic convergence is not always guaranteed. The later algorithm is generated from minimizing the quadratic cost function which includes errors, control signal and the difference of control signals between two iterations. Hence, this approach is an optimal algorithm [18] - [20].

1.4 Motivation and Contribution

In this thesis, we investigate the relationship between trajectory planning and ILC controller in multiple points tracking problems. Our objectives are to utilize the repetitive nature of systems that operate repetitively through learning algorithms for making the tracking problem simpler, reducing computational cost while increasing system performance.

Firstly, we present a new learning technique that both reference trajectory and control signal are updated iteratively. As discussed, most of ILC algorithms come from research on tracking a reference trajectory. The objective is to produce control inputs such that errors between outputs and this trajectory are decreased trial by trial. Accordingly, the given reference trajectory plays an important role in the performance of the ILC controller. However, there is no connection between trajectory planning and the control algorithm; thus, it is not certain to guarantee that the specified trajectory is the optimal trajectory for the ILC implementations. Here, we develop this idea to construct an update law for reference trajectory such that convergence rate of the controller is improved. Also, it is natural to consider an ILC scheme such that the reference trajectory could be updated instead of being fixed in the
iteration domain. The advantage of fast convergence is the decrease of cost and complexity on practical implementation of real systems.

Secondly, we propose a direct approach where we do not need to divide the tracking problem into two steps separately. The strength of the proposed formulation is the methodology to obtain a control signal through learning laws that take into account the waypoints and dynamic system. Dividing the tracking problem into trajectory planning and ILC trajectory tracking shows drawbacks under certain circumstances. First, most trajectory planning algorithms face difficulties in generating an optimal reference trajectory. In particular, the existence of a large number of data points can lead to a significant increase in the computational analysis and memory requirements. Second, ILC theory has shown that the system performance and rate of convergence depend on both the system dynamics and the reference trajectory [4]. Consequently, even if an optimal trajectory is chosen, the ILC controller could be unsatisfactory. Third, the existence of errors in both stages can result in performance degradation as an effect of the indirect method. And the last reason is that when there is a change in the motion profile, the system has to be conducted again from beginning at both stages. Therefore, these reasons motivate our study to combine two stages into one ILC controller such that it improves the performance and optimizes the computational cost.

Lastly, we consider to allow the desired data points are not to be fixed, but instead to vary or fluctuate inside an interval iteratively. Since our approach for multiple tracking is with data points directly rather than the reference trajectory, this problem happens naturally in many applications. For example, the data points could not be defined exactly because of noise contaminated such as measurement noise, sensor noises, disturbances,... Moreover, the
given data could be uncertain between different iterations. Thus, specifically this allows us to apply learning algorithms to broader class of applications.

1.5 Organization of the Thesis

Chapter 2 starts with a discussion of some general introduction and properties of terminal ILC. As the first approach, we consider a single terminal point with initial input learning. Multiple intermediate pass points control problems, which based on the initial learning method, are subsequently examined by applying both an initial learning control input and a continuous control signal. Simulation results are given in the end of the chapter to verify our approaches.

In chapter 3, we develop a new approach which corporates both trajectory planning and trajectory tracking ILC. The objective of this work is to show that it is possible to make benefits from these two separate stages. At first, initial reference trajectory generation is given, then the learning update law for the reference trajectory iteratively is considered. Next, the ILC algorithms of both input and trajectory with the convergence and performance analyses are demonstrated. Finally, simulation results and conclusions are given.

The results on a new multiple points tracking ILC without reference trajectory is analyzed in the chapter 4. We develop the algorithms in both continuous and discrete time systems to show that it is not necessary to consider the whole reference trajectory for the tracking problem. On the other hand, our approach considers only given terminal points. To clearly show the effective of our approach, we also analyze the algorithm with input constraints, as well as simulations.
The last theory works in chapter 5 are followed by considering uncertainties in data points, and interval multiple points tracking. Specifically, the uncertainties are varied iteratively, and bounded. Hence, the problem now becomes a two objectives optimization problem; consequently, our approach is to deal with a min-max formulation. Comparing to the approach without uncertainties, the optimal control signal algorithm is now distinct in the learning gains which are modified by the uncertainty bound value.

To clearly verify our contribution, examples are shown in chapters 6 and chapter 7 through the re-heating of thermoforming machine and satellite antenna pointing applications, respectively. Finally, in chapter 8, conclusions are drawn from the work presented in the preceding chapters and several directions for further research are discussed.
Chapter 2

Terminal Iterative Learning Control

2.1 Introduction

The objective of this chapter, as our first try, is to explore terminal iterative learning control (TILC) in both single and multiple intermediate pass points tracking controls. Specifically, we apply an initial input learning technique to track one terminal point, because of the potential to increase the smooth motion of systems and reducing the effects of actuator limitations. Additionally, it is not necessary to continuously apply control signals that lead to more control efforts if the desired terminal output can be reached with an appropriate initial control input. After that, continuous control signals are generated to produce output curves that track multiple pass points based on the initial learning approach. The technique investigates only essential information at terminal points. Note that our approach is based on the norm optimal ILC, in which the ILC update law is obtained by minimizing the cost function trial-by-trial. For these problems, we show that stability and convergence in the iteration domain can be achieved and that the algorithms can produce superior performance by selecting suitable parameters.
2.2 Background

This section provides the problem setup of TILC control for a single terminal point and multiple pass points. Consider a discrete time linear system

\[
\begin{align*}
    x_k(t+1) &= Ax_k(t) + Bu_k(t) \\
y_k(t) &= Cx_k(t)
\end{align*}
\]

(2.1)

where \( k \) is the iteration index, and \( t = 0, 1, 2, \ldots, N - 1 \) is the sampling time index. Matrices \( A, B, \) and \( C \) are time invariant with appropriate dimensions; this system is a multi-input multi-output (MIMO) system that has \( x_k(t) \in \mathbb{R}^p, u_k(t) \in \mathbb{R}^m, \) and \( y_k(t) \in \mathbb{R}^n. \) Moreover, we assume that the system is both controllable and observable.

From linear control theory, the output of the system at the \( N \)-th sample time in the \( k \)-th iteration is given as

\[
y_k(t_N) = CA^tN x_k(0) + C \sum_{j=0}^{t_N-1} A^{t_N-j-1} Bu_k(j).
\]

(2.2)

Also, the initial state condition is assumed to be constant for all iterations; moreover, it is possible to assume that \( x_k(0) = 0 \) without loss of generality. The goal of TILC is to track terminal points during system operation by generating an optimal control signal through trials. After each trial, the outputs at the given terminal points are measured; consequently, the input is updated from an ILC learning law.

During the tracking of a single terminal point \( t_N \) that has its desired output is \( y_d(t_N) \), the input signal is constant at all sampling times in the same iteration, i.e., \( u_k(t_i) = u_k \) for \( t_i = \{0, 1, 2, \ldots, N\} \). As a result, the output of the terminal point can be described by

\[
y_k(t_N) = P_N u_k,
\]

(2.3)
where $P_N$ is the system matrix at the $N$-th terminal point, such that

$$P_N = C \sum_{j=0}^{N-1} A^{N-j-1}B. \quad (2.4)$$

Note that the system matrix $P_N$ is full rank, because the system is both controllable and observable.

On the other hand, the terminal control task that has intermediate pass points between the initial point and the terminal point is relatively more challenging. In this case, a control signal is generated such that the produced output curves go through the given pass points iteratively.

We define these points at each time instant as $t_1, t_2, \ldots, t_M$, where $0 \leq t_1 < t_2 < \ldots < t_M \leq t_N$, and the desired outputs at these points are

$$y_{d}(t_1), y_{d}(t_2), \ldots, y_{d}(t_M).$$

And in the $k$-th iteration, the output at the $i$-th intermediate point is calculated as

$$y_k(t_i) = C \sum_{j=0}^{i-1} A^{i-j-1}Bu_k(j). \quad (2.5)$$

### 2.3 Initial Iterative Learning for Single Terminal Point

In this section, we consider the initial learning control strategy for tracking the $N$-th terminal point. To investigate the norm optimal ILC approach, we suggest a performance index with respect to the terminal point, such that

$$J(u_{k+1}) = e_{k+1}^T(t_N)Qe_{k+1}(t_N) + (u_{k+1} - u_k)^T R(u_{k+1} - u_k) + u_{k+1}^T Su_{k+1}, \quad (2.6)$$

where $Q, R,$ and $S$ are real symmetric positive definite matrices with appropriate dimensions, and the error at the terminal is

$$e_k(t_N) = y_{d}(t_N) - y_k(t_N). \quad (2.7)$$
Here, the control signal at the \((k+1)\)-th trial can be attained by differentiating vector \(J(u_{k+1})\) with respect to \(u_{k+1}\); setting this derivative equal to zero yields

\[-P_N^T Q e_{k+1}(t_N) + R(u_{k+1} - u_k) + S u_{k+1} = 0.\] (2.8)

Then, the control input is iteratively computed as

\[(P_N^T Q P_N + R + S) u_{k+1} = R u_k + P_N^T Q y_d(t_N).\] (2.9)

Since \(P_N^T Q P_N + R + S\) is nonsingular, (2.9) can be rewritten by the following ILC rule

\[u_{k+1} = L_u u_k + L_d y_d(t_N),\] (2.10)

where

\[L_u = (P_N^T Q P_N + R + S)^{-1} R,\] (2.11)

\[L_d = (P_N^T Q P_N + R + S)^{-1} P_N^T Q.\] (2.12)

The linear iterative system (2.10) is asymptotically stable (AS) if it generates inputs through trials that are bounded for all iterations, and \(u_\infty = \lim_{k \to \infty} u_k\) exists. Furthermore, the system is AS if

\[\rho(L_u) < 1,\] (2.13)

where \(\rho(L_u)\) is the spectral radius of \(L_u\) [3]. The controller is AS is thus convergent. As a result, we obtain the following stability property of the ILC law (2.10).

**Theorem 1** For the linear system (2.1), the ILC learning algorithm

\[u_{k+1} = L_u u_k + L_d y_d(t_N)\] (2.14)

is asymptotically stable for all symmetric positive definite matrices \(Q, R,\) and \(S\).
Considering inverse of the nonsingular matrix $L_u$,

\[
L_u^{-1} = R^{-1} (P_N^T Q P_N + R + S)
\]

\[
= I + R^{-1} P_N^T Q P_N + R^{-1} S.
\]  \hfill (2.15)

It is then obvious that $R^{-1} P_N^T Q P_N + R^{-1} S$ is positive definite; consequently, the matrix $L_u^{-1}$ has its all eigenvalues greater than 1, as shown in the following.

Consider a positive definite matrix $M$ and an identity matrix $I$. If we define the eigenvalues of the matrices $M$ and $(I + M)$ as $\lambda(M)$ and $\lambda(I + M)$, respectively, then

\[
\lambda(I + M) = \lambda(M) + 1.
\]  \hfill (2.16)

Since $\lambda(M) > 0$, then $\lambda(I + M) > 1$. Therefore, $L_u^{-1}$ has all eigenvalues greater than 1. Accordingly, it leads to the conclusion from the inverse eigenvalue theorem in linear algebra [26],

\[
\rho(L_u) = \max |\lambda_i(L_u)| < 1.
\]  \hfill (2.17)

However, the concept of AS is not strongly stated in ILC applications due to the inherently large transient growth possibility, which is one of main obstacles in ILC design. Hence, another strong concept in ILC is the monotonic convergence condition. Briefly, let us provide the background ideas and conditions for monotonic convergence.

The ILC algorithm is referred to as monotonically convergent if $\|u_\infty - u_{k+1}\| < \gamma \|u_\infty - u_k\|$ such that $0 < \gamma < 1$ [3]. In the learning algorithm (2.10), $\gamma$ is defined as $\gamma = \|L_u\|$, which is the largest singular value of $L_u$. Accordingly, we consider another of the selected weighting matrices to guarantee the monotonic convergence of the law.
Lemma 2.3.1 For the linear system (2.1), the ILC learning algorithm

\[ u_{k+1} = L_u u_k + L_d y_d(t_N) \]  

(2.18)

guarantees monotonic convergence if the weighting matrices are chosen as \( Q = qI, R = rI, \) and \( S = sI, \) where \( q, r, \) and \( s \) are real positive parameters.

Applying the contraction mapping theorem with \( u_{k+1} = f(u_k), \)

\[ \| f(u_1) - f(u_2) \| = \| L_u (u_1 - u_2) \| \leq \bar{\sigma}(L_u) \| u_1 - u_2 \|. \]  

(2.19)

where the largest singular value of \( L_u \) is defined as

\[ \bar{\sigma}(L_u) = \sqrt{\rho(L_u^T L_u)}. \]  

(2.20)

Moreover, with \( Q = qI, R = rI, \) and \( S = sI, \) the learning matrix \( L_u \) is symmetric positive definite, \( L_u^T = L_u. \) Since the eigenvalues of \( L_u^2 \) are the squares of the eigenvalues of \( L_u, \) then

\[ \sqrt{\rho(L_u^T L_u)} = \sqrt{\rho(L_u^2)} = \rho(L_u). \]  

(2.21)

And the result of Theorem (1) leads to

\[ \sigma(L_u) < 1, \]  

(2.22)

and the final result is given.

As a consequence of the convergence property, the control signal at the steady state is calculated from (2.10) as

\[ u_\infty = (I - L_u)^{-1} L_d y_d(t_N) \]

\[ = (P_N^T Q P_N + S)^{-1} P_N^T Q y_d(t_N). \]  

(2.23)
Hence, the converged error \( e_\infty = \lim_{k \to \infty} e_k \) is

\[
e_\infty = y_d(t_N) - P_N u_\infty
= \left[ I - P_N (P_N^T Q P_N + S)^{-1} P_N^T Q \right] y_d(t_N).
\] (2.24)

From the steady state error, we can see that the steady state error depends on the relationship between matrices \( Q \) and \( S \). Specifically, if \( Q \) is large compared to \( S \), component-wise, then the error is small. Moreover, the smallest possible error, \( e_\infty = 0 \), requires \( S = 0 \).

### 2.4 ILC for Multiple Intermediate Pass Points

In the multiple intermediate pass points problem, there are given desired outputs \( y_d(t_1), y_d(t_2), \ldots, y_d(t_M) \) at time instants \( t_1, t_2, \ldots, t_M \) during system operation. The control task is to then construct a learning law that drives the outputs through, or at least close to, these points. In conventional control schemes, a reference trajectory \( y_{ref} \) is built such that \( y_{ref} \) passes the desired points at \( t_1, t_2, \ldots, t_M \). In this case, we can design a controller that incorporates the system model to thereby track the given trajectory. However, in this work, we attempt to design a new ILC formulation that focuses on only information obtained from the given pass points rather than use a reference trajectory.

#### 2.4.1 Initial Learning

As a initial attempt, we apply the same approach as the tracking single terminal point case. Hence, if the control signal is maintained constant as initial iterative learning, the output at the \( i \)-th pass points can be given as

\[
y_k(t_i) = P_i u_k,
\] (2.25)
where
\[ P_i = C \sum_{j=0}^{i-1} A^i \cdot A^{i-j-1} B. \quad (2.26) \]

The error at the \( i \)-th point is then computed as
\[ e_k(t_i) = y_d(t_i) - P_i u_k. \quad (2.27) \]

Next, we consider a norm optimal TILC performance index that incorporates multiple pass points \( t_1, t_2, \ldots, t_M \), such that
\[ J(u_{k+1}) = \sum_{i=1}^{M} e_{k+1}^T(t_i)Q_i e_{k+1}(t_i) + (u_{k+1} - u_k)^T R (u_{k+1} - u_k) + u_{k+1}^T S u_{k+1}, \quad (2.28) \]

where \( Q_i \) is the weighting matrix at the \( i \)-th terminal point.

Note that the primary objective here is to generate control signals that produce an output that minimizes errors at all terminal points iteratively. The ILC algorithm is generated from (2.28) by differentiating \( J(u_{k+1}) \); setting this differentiation to zero then produces
\[ \left( \sum_{i=1}^{M} P_i^T Q_i P_i + R + S \right)^{-1} R u_k + \sum_{i=1}^{M} P_i^T Q_i y_d(t_i). \quad (2.29) \]

Next, since \( \sum_{i=1}^{M} P_i^T Q_i P_i + R + S \) is nonsingular, (2.29) can be rewritten as
\[ u_{k+1} = \left( \sum_{i=1}^{M} P_i^T Q_i P_i + R + S \right)^{-1} R u_k + \sum_{i=1}^{M} P_i^T Q_i y_d(t_i). \quad (2.30) \]

Accordingly, a theorem can be formulated to analyze the stability of the update algorithm.

**Theorem 2** For the linear system (2.1), the ILC learning algorithm (2.30) for tracking multiple pass points is asymptotically stable for symmetric positive definite matrices \( Q, R, \) and \( S \).
Since \( P_i^T Q_i P_i \) is positive definite with all pass points \( t_1, t_2, \ldots, t_M \), the result can be shown in the same way as Theorem (1).

To show the effectiveness of investigating the initial learning for tracking multiple pass points, the performance at each point is subsequently analyzed. First, the steady state control signal achieved from (2.30) is

\[
u_{\infty} = \left( \sum_{i=1}^{M} P_i^T Q_i P_i + S \right)^{-1} \sum_{i=1}^{M} P_i^T Q_i y_d(t_i).
\] (2.31)

After that, the converged error of the terminal points at the \( i \)-th sampling time is as follows:

\[
e_{\infty}(t_i) = y_d(t_i) - P_i u_{\infty}.
\] (2.32)

Therefore, the steady state error of the terminal point depends on the other terminal pass points, in addition to the weighting matrices \( Q_i \) and \( S \). Moreover, the weighting matrix \( Q_i \) in the relationship with other terminal weighting matrices describes how important it is that the output curve \( y_{\infty}(t_i) \) goes close to the desired output \( y_d(t_i) \). Thus, this technique only could achieve desirable performance at all points under certain conditions of given pass points.

### 2.4.2 Iterative Learning using Continuous Control Input

When there are a large number of given intermediate pass points, the initial learning approach has inherent drawbacks in achieving performance at all points. For this reason, we consider a new ILC framework in which the control signal is time continuous.

First, we formulate the \( N \)-sample sequence of inputs in a super-vector framework as

\[
u_k = \begin{bmatrix} u_k^T(0) & u_k^T(1) & \ldots & u_k^T(N-1) \end{bmatrix}^T.
\] (2.33)
Then, we define $p_i(t)$ as

$$p_i(t) = \begin{cases} 
CA^{t-t_i-1}B & \text{if } t < t_i \\
0 & \text{if } t \geq t_i 
\end{cases}.$$ 

By these formulations, the output at the $i$-th time instant is expressed as

$$y_k(t_i) = C \sum_{j=0}^{t_i-1} A^{t_j-1} Bu_k(j)$$

$$= \sum_{t=0}^{N-1} p_i(t)u_k(t)$$

$$= p_i^T u_k$$

where $p_i$ is expressed as

$$p_i = \begin{bmatrix} p_i(0) & p_i(1) & \ldots & p_i(N-1) \end{bmatrix}^T.$$ 

Therefore, the cost function for the problem of tracking multiple intermediate pass points $t_1, t_2, \ldots, t_M$ can now be formulated as

$$J = \sum_{i=1}^{M} e_k^T(t_i)Q_i e_k(t_i) + u_{k+1}^T S u_{k+1} + (u_{k+1} - u_k)^T R (u_{k+1} - u_k)$$

where

$$e_k(t_i) = y_d(t_i) - p_i^T u_{k+1}$$

and $R, S, Q_i$ are symmetric positive definite matrices.

To work with multiple pass points, we define the super vector forms of system matrix $P$ and the desired output at pass points $y_d$ as

$$y_d = \begin{bmatrix} y_d^T(t_1) & y_d^T(t_2) & \ldots & y_d^T(t_M) \end{bmatrix}^T$$

$$P = \begin{bmatrix} p_1^T & p_2^T & \ldots & p_M^T \end{bmatrix}^T.$$
Note that different \( p_i(t) \) vanish at different times, thus the set of functions \( p_i(t) \) with \( i = 1, 2, \ldots, M \) are linearly independent. As such, the cost function (2.36) can be rewritten as

\[
J = \left[ y_d - Pu_{k+1} \right]^T Q \left[ y_d - Pu_{k+1} \right] + u_{k+1}^T Su_{k+1} + (u_{k+1} - u_k)^T R (u_{k+1} - u_k)
\]

where \( Q = \text{diag}(Q_1, Q_2, \ldots, Q_M) \). Consequently, the controller in the \((k+1)\)-th trial is attained from the differential condition, leading to

\[
-P^T Q (y_d - Pu_{k+1}) + R (u_{k+1} - u_k) + Su_{k+1} = 0. \tag{2.41}
\]

And the ILC algorithm is derived as

\[
(P^T Q P + R + S) u_{k+1} = Ru_k + P^T Q y_d. \tag{2.42}
\]

Also, since \( P^T Q P + R + S \) is positive definite,

\[
L_u = (P^T Q P + R + S)^{-1} R \tag{2.43}
\]

\[
L_d = (P^T Q P + R + S)^{-1} P^T Q y_d. \tag{2.44}
\]

The following theorem illustrates the results of this approach.

**Theorem 3** For the linear system (2.1), the ILC learning algorithm

\[
u_{k+1} = L_u u_k + L_d \tag{2.45}
\]

is asymptotically stable for all symmetric positive definite matrices \( Q, R \) and \( S \).

Moreover, if matrices \( R, S, \) and \( Q_i \), where \( i = 1, 2, \ldots, M \) are chosen as \( Q = qI \), \( R = rI \), \( S = sI \), then the algorithm achieves monotonic convergence.
The learning algorithm is presented in the lift domain notation of the discrete system. However, the result is still based on the result of Theorem (1).

The steady state input and errors are given from (2.42) as

\[ u_\infty = (P^T Q P_N + S)^{-1} P^T Q y_d. \]  

The steady state error \( e_\infty = y_d - P u_\infty \) is shown as

\[ e_\infty = \left[ I - P \left( P^T Q P + S \right)^{-1} P^T Q \right] y_d. \]  

Hence, \( Q \) and \( S \) decide the performance of the tracking technique. In practical applications, there is always the case that the importances of particular points are different. As a result, the entries of the matrix \( Q \) decide how different performance the points are achieved.

### 2.5 Simulation

To illustrate the ideas presented in this chapter, consider the discrete-time system

\[
x_k(t + 1) = \begin{pmatrix} 0.5 & 0.035 & 0.025 \\ 0.0255 & 0.6 & -0.99 \\ 0.75 & 0.03 & 0.025 \end{pmatrix} x_k(t) + \begin{pmatrix} 0.2 \\ 0.2 \\ 0.0 \end{pmatrix}^T u_k(t)
\]

\[
y_k(t) = \begin{pmatrix} 1.0 & 0.0 & 1.0 \end{pmatrix} x_k(t).
\]

where the system operates on an interval \( t \in [0, 20] \).

In the first simulation in Fig.2.1, the ILC law for tracking the single terminal point \( y_d = 2 \) at \( t = 20 \) is used with weighting parameters \( Q = 15, R = 0.7 \) and \( S = 0.3 \). The figure shows the superior performance and fast convergence of the errors in the iteration domain. As a
second simulation example Fig.2.2 shows the output curves which are produced from initial learning algorithm to control 3 pass points. Although the errors achieve convergence, the performances are dependence at given points.

In the next simulation, we demonstrate multiple pass points TILC using continuous control signals in Fig.2.3. A set of 10 points in the interval $[0, 20]$ was selected. The ILC law is conducted with $Q = qI$, $R = rI$ and $S = sI$ where the scalar gains are chosen as $q = 50$, $r = 5$ and $s = 0.1$. It is shown in Fig.2.3 that the convergence is obtained after 8 iterations with the converged error is approximate to zero.

By comparison, the final simulation tests an ILC algorithm which tracks a reference trajectory instead of points. First, a trajectory which goes through given points is generated from an interpolation splines technique. Then, an ILC law is made with this trajectory and the same scalar gains of diagonal weighting matrices. Fig.2.4 shows the performance almost the same as in Fig.2.3. Hence, our approach which does not require generating trajectory could obtain a similar result of tracking a trajectory. Moreover, we calculate the converged control energy in both cases. For our approach, the cost is calculated as

$$\frac{1}{2} \|u(t)\|^2 = 19.72,$$ (2.49)

while the trajectory tracking technique requires the larger cost of 21.05.

2.6 Conclusion

In this chapter, we have formulated and analyzed the TILC problem in single and multiple pass points for MIMO systems. It was found here that the optimal TILC scheme provided a suitable framework for obtaining asymptotic stability and monotonic convergence properties.
Moreover, the approach utilized only essential information at terminal points instead of the whole trajectory, which enabled us to improve the ILC controller, with respect to reducing the complexity and computational effort.
Figure 2.1: Initial learning for single terminal point
Figure 2.2: Initial learning for multiple pass points
Figure 2.3: Convergence of the points’s errors in tracking the reference trajectory
Figure 2.4: Convergence of the points’s errors in our approach
Chapter 3

Interpolation-Based Approach for Multiple Terminal Iterative Learning Control

3.1 Introduction

As discussed, there is a number of works in the topic of multiple points reaching in control theory. Most of them are based on tracking a fixed reference trajectory which is planned to achieve the desired terminal conditions. As a consequence, taking the reference trajectory as the objective, the ILC algorithm is applied to produce control inputs such that errors between outputs and the reference trajectory is decreased trial by trial as conventional ILC techniques. However, one of the important points in the ILC algorithm is the dependence of performance and rate of convergence on the given reference trajectory in each iteration and in the converged error. Therefore, it is natural to consider an ILC controller that reference trajectory could be updated by iterations for the performance improvement. Additionally, the chosen reference trajectory is usually considered before starting actual system operation, thus it is not guaranteed that this reference trajectory is the optimal trajectory regarding to performance and rate of convergence in the learning update law during trials. Another motivation of this idea is that there is not only one trajectory which satisfies the terminal conditions. In this chapter, we develop this idea by iteratively updating both input and ref-
ference trajectory to improve the rate of convergence in the ILC algorithm. The advantage of faster convergence is the decrease of cost and complexity on practical implementation of real systems.

In this chapter, we develop a new ILC algorithm to utilize the flexible selection of trajectory in terminal control. While one work has been considered on improving the trajectory in frequency domain for single-input single-output (SISO) systems [17], this work proposes an iterative learning update law for a class of iteration-updated trajectories based on the information availability of the given points. Note that this law can be used for not only SISO systems but also multi-input multi-output (MIMO) systems. Afterward, the iterative learning update law for control input is presented. Due to the flexibility of choosing learning factors in both controller and reference trajectories, our approach has significant advantages in convergence rate and performance compared to conventional ILC techniques.

3.2 Background

In a multiple terminal points problem, there are specified time instants in system operation $t_1, t_2, \ldots, t_M$, where $0 \leq t_1 < t_2 < \ldots < t_M \leq N - 1$. Let us define the desired outputs at these points as

$$y_d(t_1), y_d(t_2), \ldots, y_d(t_M).$$

The goal of multiple TILC is to design the ILC controller such that its system output passes through the given terminal points at the given time instants. To investigate the ILC theory, the first stage is to plan a reference trajectory $r(t)$ such that $r(t)$ goes through the desired points at $t_1, t_2, \ldots, t_M$. After that, an ILC controller is designed to track the reference trajectory in
the iteration domain. Specifically, the reference trajectory satisfies the terminal conditions

\[
\begin{align*}
    r(t_1) &= y_d(t_1),
    \\
r(t_2) &= y_d(t_2),
    \\
    &\vdots
    \\
r(t_M) &= y_d(t_M).
\end{align*}
\]

(3.1)

The conventional technique to find the reference trajectory from the given points is interpolating splines which is a tool in numerical analysis. The generated output function is an interpolating splines if it exactly interpolates given points. The framework of interpolating splines is dealing with the problem of solving

\[
\min \int_0^{N-1} r''(t)^2
\]

subject to \(r(t_i) = y_d(t_i)\).

Another interpolation technique which could be generated from numerical calculations is polynomial interpolation. For a given set of instant times and the corresponding outputs, the desired polynomial could be generated by the Lagrange polynomial approach as a linear combination of the Lagrange basis polynomials:

\[
r(t) = \sum_{i=1}^{M} y_d(t_i)L_{M,i}(t)
\]

where the Lagrange basis polynomials are defined by

\[
L_{M,i}(t) = \prod_{k \neq i} \frac{t-t_k}{t_i-t_k} = \frac{t-t_1}{t_i-t_1} \cdots \frac{t-t_{i-1}}{t_i-t_{i-1}} \frac{t-t_{i+1}}{t_i-t_{i+1}} \cdots \frac{t-t_M}{t_i-t_M}
\]
It is noted that the interpolating polynomial of the least degree is unique; however, there are unlimited polynomials with higher degree of polynomial.

Since the objective of this work is tracking control ILC-based, we can assume that the reference trajectory $r_0(t)$ which satisfies the condition (3.1) is obtained from the interpolation technique. Define the reference trajectory in the super-vector form:

$$r_0 = \begin{bmatrix} r_0^T(1) & r_0^T(2) & \ldots & r_0^T(N) \end{bmatrix}^T.$$

The ILC learning algorithm utilizes errors and control signals from previous iterations to compute updated control inputs with the learning matrix $L$ as

$$u_{k+1} = u_k + Le_k$$

where

$$e_k = r_0 - y_k.$$

As a result, the ILC algorithm is convergence if $\rho(I - LP) < 1$, where $\rho(A)$ is the spectral radius of the matrix $A$. Furthermore, the algorithm will achieve monotonic convergence, if the following requirement is satisfied: $\|I - LP\| < 1$.

### 3.3 Reference Trajectory Learning Update

The general problem of updating trajectory that we try to address can be formulated as finding a new reference trajectory in each iteration such that it brings a better performance than traditional ILC approaches where the reference trajectory is fixed throughout all iterations. Moreover, the new trajectories also satisfy the terminal condition (3.1). In this section, we present an approach to identify a class of trajectories that go through the terminal points
with specified outputs based on the interpolation technique. Next, the ILC update algorithm for reference trajectory is investigated.

3.3.1 Analysis

Employing the interpolation technique, a trajectory that passing all terminal points at specified instant times is assumed to be obtained. The initial reference trajectory $r_0(t)$ satisfies the terminal condition as

$$r_0(t_i) = y_d(t_i).$$

where $i = 1, 2, ..., M$.

Here, we propose an approach to find a class of trajectories which also pass the same points $t_1, t_2, ..., t_M$ as $r_0(t)$. Particularly, a new trajectory $r_i(t)$ is generated from the relationship as

$$r_i(t) = r_0(t) + h(t)g(t)$$

where

$$h(t) = (t - t_1)(t - t_2)\ldots(t - t_M)$$

with $t \in [1, N]$, and $g(t)$ is an arbitrary continuous function. Obviously, since $h(t_i) = 0$ at $i = 1, 2, ..., M$; the requirement of the new trajectory is satisfaction.

3.3.2 Reference Trajectory Update Law

Next, the proposed approach is applied to the update law of the reference trajectory. Let us denote the function $g(t)$ in the $k$-th iteration in the super vector form as

$$g_k = [g_k^T(1) \ g_k^T(2) \ \ldots \ g_k^T(N)]^T$$
and the reference trajectory at the iteration \(k\)-th is defined by

\[
r_k = \begin{bmatrix} r_k^T(1) & r_k^T(2) & \cdots & r_k^T(N) \end{bmatrix}^T
\]

Consequently, it leads to the following result on the update law of reference trajectory in the ILC design stage.

**Lemma 3.3.1** Given the initial reference trajectory \(r_0\), the trajectory which is updated from the following law

\[
r_{k+1} = r_k + Hg_{k+1},
\]

(3.3)

where

\[
H = \begin{bmatrix}
h(1) & 0 & \cdots & 0 \\
0 & h(2) & \cdots & 0 \\
0 & 0 & \ddots & \vdots \\
0 & 0 & \cdots & h(N)
\end{bmatrix}
\]

and

\[
h(t) = (t-t_1)(t-t_2)\ldots(t-t_M)
\]

guarantees the terminal condition (3.1) as the initial reference trajectory \(r_0\).

The proof is followed from equation (3.2), since \(h(t_i) = 0\) then

\[
r_{k+1}(t_i) = r_k(t_i)
\]

As a result, the updated trajectories at given terminal time instants remain constant. In addition, the new reference trajectory at other instant times is dependent on \(g_{k+1}\) and the diagonal matrix \(H\). In the learning update algorithm (3.3), \(g_{k+1}\) can be considered as a learning factor.
3.4 Controller

In this section, we analyze the ILC technique that both reference trajectory and input are iteratively updated in the ILC algorithm. First, we define the condition for the selection of a new trajectory in the iteration domain to satisfy the system improvement. Second, the monotonic convergence analysis of the new update law will be presented. Finally, we show that the new ILC scheme has faster rate of convergence than conventional ILC technique.

In the \((k+1)\)-th iteration, there are two updating rules,

\[
 u_{k+1} = u_k + L(r_{k+1} - y_k), \quad (3.4)
\]

and

\[
 r_{k+1} = r_k + Hg_{k+1}. \quad (3.5)
\]

Our scope of the trajectory update law is to increase the convergence rate of the ILC algorithm in compared to the same ILC algorithm with the given preplanned reference trajectory. Therefore, to select a new trajectory for better convergence at the \((k+1)\)-th trial, \(g_{k+1}\) should be chosen such that

\[
 \|r_{k+1} - y_k\| < \|r_k - y_k\|. \quad (3.6)
\]

As a consequence, the new trajectory is closer in the sense of norm with the current output compared to the reference trajectory in the previous iteration. The next lemma gives the solution for that condition.

**Lemma 3.4.1** Consider the learning update algorithm

\[
g_{k+1} = F(r_k - y_k), \quad (3.7)
\]
if the matrix $F$ satisfies

$$\|I + HF\| < 1$$

then

$$\|r_{k+1} - y_k\| < \|r_k - y_k\|.$$  \hfill (3.9)

From (3.5) and (3.7),

$$r_{k+1} - y_k = r_k + HF(r_k - y_k) - y_k = (I + HF)(r_k - y_k).$$

Hence, we obtain

$$\|r_{k+1} - y_k\| \leq \|I + HF\| \|r_k - y_k\|.$$  

The result follows since $\|I + HF\| < 1$.

Note that the matrix $H$ is given and specified based on the terminal points. And $H$ is a diagonal matrix, thus it is possible to find the suitable matrix $F$ analytically such that the condition (3.8) is achieved. Here, we propose an approach to define $F$ as a diagonal matrix,

$$F = \begin{bmatrix}
\gamma(1) & 0 & \ldots & 0 \\
0 & \gamma(2) & \ldots & 0 \\
0 & 0 & \ddots & \vdots \\
0 & 0 & \ldots & \gamma(N)
\end{bmatrix}$$  \hfill (3.10)

where

$$\gamma(t) = -\alpha \frac{\text{sgn}(h(t))}{M}; 0 < \alpha < 1$$  \hfill (3.11)
with $M = \max |h(t)|, t = 1, 2, \ldots, N$. From this settings, the matrix $(\mathbf{I} + \mathbf{H}\mathbf{F})$ is a diagonal matrix that has entries on diagonal area defined from

$$
\mathbf{I} + \mathbf{H}\mathbf{F} = \text{diag}(1 + h(t)\gamma(t))
$$

at $t = 1, 2, \ldots, N$. The definition leads to the inequality

$$
\|\mathbf{I} + \mathbf{H}\mathbf{F}\| < \max_{t=1,\ldots,N} |1 + h(t)\gamma(t)|.
$$

In the time interval $t \in [1, N]$, $h(t)$ is a continuous function, hence there exist the maximum value of $h(t)$,

$$
M = \max |h(t)|, 1 \leq t \leq N.
$$

Thus from the setting of $\gamma(t)$ in (3.11),

$$
\max_{t=1,\ldots,N} |1 + h(t)\gamma(t)| = 1,
$$

the condition (3.8) is achieved.

By the chosen of the new update law for the reference trajectory, we now show the monotonic convergence property of the new ILC algorithm. We first provide the definition of monotonic convergence.

**Definition 3.4.1** An ILC system is monotonic convergence if the errors in the iteration domain satisfy

$$
\|e_{k+1}\| < \|e_k\|
$$

where the error at the iteration $k$-th is defined as

$$
e_k = r_k - y_k
$$
Consequently, the monotonic convergence property of our approach is given.

**Theorem 4** Consider the linear system (1.4) controlled with the ILC updating equations

\[
\begin{align*}
    u_{k+1} &= u_k + L(r_{k+1} - y_k) \\
    r_{k+1} &= r_k + HF(r_k - y_k)
\end{align*}
\]  

(3.12)

where \( F \) is chosen as in the Lemma (3.4.1), then the ILC system is monotonic convergence if

\[
\| I - PL \| < 1
\]

The error at the \((k+1)\)-th iteration is derived as

\[
\begin{align*}
    r_{k+1} - y_{k+1} &= r_{k+1} - P[u_k + L(r_{k+1} - y_k)] \\
    &= r_{k+1} - y_k - PL(r_{k+1} - y_k) \\
    &= (I - PL)(r_{k+1} - y_k)
\end{align*}
\]

Hence, it leads to the inequality

\[
\| e_{k+1} \| \leq \| I - PL \| \| e_k \| \\
\leq \| I - PL \| \| I + HF \| \| e_k \|
\]  

(3.13)

Finally, from Theorem (4), the monotonic convergence is guaranteed if \( \| I - PL \| < 1 \).

To clearly show the effect of our approach, the following corollary demonstrates the faster rate of convergence of our approach than conventional ILC approach.

**Corollary 3.4.1** The established algorithm with updating trajectory shows a faster rate of convergence compared to the ILC algorithm with a fixed trajectory.
Consider the ILC control algorithm which uses the fixed trajectory \( r \) as an objective in all iteration,

\[
\bar{u}_{k+1} = \bar{u}_k + L_e (r - \bar{y}_k)
\]

As a result, the error at the \((k+1)\)-th is

\[
\bar{e}_{k+1} = r - \bar{y}_{k+1} = r - P \left[ \bar{u}_k + L(r - \bar{y}_k) \right] = (I - PL)\bar{e}_k,
\]

then

\[
\|\bar{e}_{k+1}\| \leq \|I - PL\| \|\bar{e}_k\|
\]

From (3.13), the ILC algorithm with updating trajectory has a faster rate of convergence since \(\|I + HF\| < 1\).

### 3.5 Simulation

Performances of the proposed techniques are presented through an example of the tracking problem with a linear discrete-time system model. We consider reference trajectory updating algorithm, and then compare with the conventional approach.

The system is chosen as

\[
\begin{align*}
x(t+1) &= \begin{pmatrix} 0.5 & 0.035 & 0.025 \\ 0.025 & 0.6 & -0.99 \\ 0.75 & 0.03 & 0.025 \end{pmatrix} x(t) + \begin{pmatrix} 0.2 \\ 0.2 \\ 0 \end{pmatrix} u(t) \\
y(t) &= \begin{pmatrix} 1.0 & 0.1 & 1.0 \end{pmatrix} x(t)
\end{align*}
\]
which operates on interval $t \in [0, 49]$. We select 5 points in the interval as desired points in the motion profile.

The first figure shows faster rate of convergence with reference trajectory updating law included in the ILC controller ($\alpha = 0.6$). The initial reference trajectory is generated by interpolation splines technique; moreover, this trajectory is also used in the conventional ILC algorithm for comparison purpose. The outputs track these updated reference trajectories, and the errors between them are expressed in the Fig.3.1. Moreover, Fig.3.2 demonstrates the updated reference trajectories and the generated output curves in the iteration domain. We can see that all the reference trajectories pass the terminal points $y_d(t_i)$. The output goes through the desired points after 50 iterations. It should be noted, in practical applications, there are constraints on the velocity, acceleration, and jerk at the terminal points; thus, they should be considered in the optimal reference trajectory.

3.6 Conclusion

This chapter presented an iterative learning control algorithm to track given multiple terminal points. Specifically, we showed a design method to improve rate of convergence of the ILC controller by investigating the relationship of possible trajectories through the interpolation technique. The main theoretical contribution of this chapter is that we can relax the condition of the fixed trajectory in the traditional ILC theory by using update laws for both input sequences and target trajectories in the iteration domain. Since our approach is based on flexible objective trajectories, this method could be developed in the future to solve other problems in ILC theory such as tracking control with changing reference trajectory.
Figure 3.1: Errors at the given points
Figure 3.2: Updated trajectories and the outputs by trials
Chapter 4

Optimal Iterative Learning Control Without Tracking a Reference Trajectory

4.1 Introduction

This chapter presents our main results. Here, we propose and analyze a direct approach for the multiple points tracking ILC control problem where we do not need to divide the tracking problem into two steps separately. As discussed, dividing the tracking problem into trajectory planning and ILC trajectory tracking shows drawbacks under certain circumstances. The strength of the proposed formulation is the methodology to obtain a control signal through learning laws that only considering the given data points and dynamic system, instead of following the direction of tracking a prior identified trajectory. In effect, it is not necessary to plan a reference trajectory as conventional approaches, which is the main advantage of learning in repetitive systems.

4.2 Optimal ILC For Continuous Systems

Let us first consider an ILC algorithm capable of the tracking problem for continuous systems (1.1). From linear system theory, the measured output of the system at the $i$–th
sample time in the \( k \)-th iteration is given as

\[
y_k(t_i) = C e^{A t_i} x_k(0) + C \int_0^{t_i} e^{A(t_i-t)} B u_k(t) dt + \omega_k(t_i) + \nu_k(t_i),
\]

where \( \omega_k(t_i) = C \int_0^{t_i} e^{A(t_i-t)} \omega_k(t) dt \).

As a result, the error is computed as

\[
e_k(t_i) = y_d(t_i) - C e^{A t_i} x_k(0) - \omega_k(t_i) - \nu_k(t_i)
\]

\[
- C \int_0^{t_i} e^{A(t_i-t)} B u_k(t) dt.
\]

Since the initial state condition is assumed to be identical in all iterations, and the system uncertainty and disturbance are repeatable; thus, without loss of generality, it is possible to replace \( (y_d(t_i) - C e^{A t_i} x_k(0) - \omega_k(t_i) - \nu_k(t_i)) \) with \( y_d(t_i) \).

By defining

\[
p_i(t) = \begin{cases} 
C e^{A(t_i-t)} B & \text{if } t \leq t_i \\
0 & \text{if } t > t_i 
\end{cases},
\]

we can rewrite the error at the time instant \( t_i \) as

\[
e_k(t_i) = y_d(t_i) - \int_0^T p_i(t) u_k(t) dt.
\]

Then, the super vector frameworks with respect to the given time instants of desired outputs and errors are given as

\[
y_d = \begin{bmatrix} y_d^T(t_1) & y_d^T(t_2) & \cdots & y_d^T(t_M) \end{bmatrix}^T,
\]

\[
e_k = \begin{bmatrix} e_k^T(t_1) & e_k^T(t_2) & \cdots & e_k^T(t_M) \end{bmatrix}^T.
\]

Similarly, given

\[
P(t) = \begin{bmatrix} p_1^T(t) & p_2^T(t) & \cdots & p_M^T(t) \end{bmatrix}^T,
\]
the multiple errors in the super vector forms are:

\[ e_{k+1} = y_d - \int_0^T P(t)u_{k+1}(t)dt. \]  \hspace{1cm} (4.5)

Next, we consider the following performance index

\[
J = \sum_{i=1}^{M} e_{k+1}^T(t_i)q_i e_{k+1}(t_i) + \int_0^T u_{k+1}^T(t) Su_{k+1}(t) dt \\
+ \int_0^T (u_{k+1}(t) - u_k(t))^T R (u_{k+1}(t) - u_k(t)) dt, \tag{4.6}
\]

where \( R, S, \) and \( q_i \) are diagonal positive definite matrices with \( R, S = (rI, sI) \); and \( q_i \) is the weighting matrix for the error at the time instant \( t_i \). We can then rewrite (4.6) to incorporate the vector form of multiple errors as

\[
J = e_{k+1}^T Q e_{k+1} + \int_0^T u_{k+1}^T(t) Su_{k+1}(t) dt \\
+ \int_0^T (u_{k+1}(t) - u_k(t))^T R (u_{k+1}(t) - u_k(t)) dt, \tag{4.7}
\]

where \( Q \) is a symmetric positive definite weight matrix.

Additionally, the entries of the matrix \( Q \) determine weightings of importance of data points. The weighting matrices \( Q, R, \) and \( S \) furthermore control the system performance and the amount of smoothing used for generating the output curves. By minimizing \( J \), a sequence of optimal control signals in the iteration domain is produced. Moreover, by driving the outputs close to the desired prespecified points, it leads to a trade-off between the control energy and the system performance. It is notable that the cost function approach was previously investigated in a norm optimal ILC [20] for treatment with a desired trajectory rather than the specific data points.
4.2.1 ILC Controller

To obtain the optimal input at the \((k + 1)\)-th iteration, differentiating \(J\) with respect to \(u_{k+1}(t) \in L^2[0,T]\), then setting this derivative to vanish yields

\[
-P^T(t) Q \left( y_d - \int_0^T P(t) u_{k+1}(t) dt \right) + (R + S) u_{k+1}(t) = R u_k(t) .
\]  

(4.8)

Here, we introduce a new variable \(z_k\) such that

\[
u_k(t) = P^T(t) z_k
\]

(4.9)

with respect to the control signal at the \(k\)-th iteration; we can now rewrite (4.8) as

\[
-P^T(t) Q \left( y_d - \int_0^T P(t) P^T(t) z_{k+1} dt \right) + (R + S) P^T(t) z_{k+1} = R P^T(t) z_k.
\]

(4.10)

With the chosen \(R, S = (rI, sI)\), the following equation is derived:

\[
-Q y_d + \left( (r+s)I + Q \int_0^T P(t) P^T(t) dt \right) z_{k+1} = rz_k.
\]

(4.11)

The new algorithm is built on the basis of vector \(z_k\), and the control inputs are followed from \(z_k\) in the iteration domain. This derivation significantly decreases the computational cost of the ILC algorithm since the dimensions of system matrices are optimized into the number of the given data points.

In the next part, we will show the convergence property of the \(z_k\) updating equation. Accordingly, the convergence property of the controller is evaluated.
4.2.2 Convergence

The set of functions $p_i(t)$ with $i = 1, 2, \ldots, M$ are linearly independent since different $p_i(t)$ vanish at different times. Thus, given

$$W = \int_0^T P(t)P^T(t)dt,$$  \hspace{1cm} (4.12)

then $W$ is a symmetric positive definite matrix. Now, (4.11) can be rewritten as

$$((r+s) I + QW)z_{k+1} = (rI + QW)z_k + Qe_k,$$  \hspace{1cm} (4.13)

where $e_k = y_d - Wz_k$.

**Lemma 4.2.1** The iterative learning equation (4.13) is convergent if $Q, R,$ and $S$ are chosen such that $\rho \left( (r+s) I + QW \right)^{-1} r < 1$.

First, the non-singularity of $((r+s) I + QW)$ is proved. It can be examined easily that the following equalities always hold with appropriate dimensions of $K, L, X, Y$, where $K, L$ is invertible:

$$\begin{bmatrix} K & 0 \\ 0 & L + YK^{-1}X \end{bmatrix} = \begin{bmatrix} I & 0 \\ -YK^{-1} & I \end{bmatrix} \begin{bmatrix} K & -X \\ Y & L \end{bmatrix} \begin{bmatrix} I & K^{-1}X \\ 0 & I \end{bmatrix};$$
$$\begin{bmatrix} K + XL^{-1}Y & 0 \\ 0 & L \end{bmatrix} = \begin{bmatrix} I & XL^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} K & -X \\ Y & L \end{bmatrix} \begin{bmatrix} I & 0 \\ -L^{-1}Y & I \end{bmatrix}.$$

Then, using the product property for determining matrices in the above equalities, we obtain

$$\det \begin{bmatrix} K & -X \\ Y & L \end{bmatrix} = \det K \det (L + YK^{-1}X)$$
$$= \det L \det (K + XL^{-1}Y)$$
\hspace{1cm} (4.14)
Therefore, we see that the non-singularity of $L + YK^{-1}X$ is equivalent to the non-singularity of $K + XL^{-1}Y$. Noting that $(r + s)I + W^{\frac{1}{2}}QW^{\frac{1}{2}}$ is nonsingular since $W^{\frac{1}{2}}QW^{\frac{1}{2}} > 0$ and $(r + s)I > 0$. Now, substituting $K = (r + s)I, X = Y = W^{\frac{1}{2}}$, and $L = Q^{-1}$; we have $(r + s)I + W^{\frac{1}{2}}QW^{\frac{1}{2}}$ is nonsingular $\iff Q^{-1} + W(r + s)^{-1}I$ is nonsingular $\iff (r + s)I + QW$ is nonsingular.

Consequently, the sequence of $z_k$ is obtained from

$$z_{k+1} = T_zz_k + T_e e_k \quad (4.15)$$

where $T_z$ and $T_e$ are defined as

$$T_z = ((r+s)I + QW)^{-1}(rI + QW), \quad (4.16a)$$
$$T_e = ((r+s)I + QW)^{-1}Q. \quad (4.16b)$$

Therefore, this results in the condition for convergence as

$$\rho (T_z - T_e W) < 1, \quad (4.17)$$

where $T_z - T_e W = ((r+s)I + QW)^{-1}r$.

As such, from the result of Lemma (4.2.1), we obtain the following convergence property of the control input.

**Theorem 5** For the linear continuous system (1.1), the following ILC system

$$\begin{cases}
    u_k(t) = P^T(t)z_k \\
    z_{k+1} = T_zz_k + T_e e_k
\end{cases} \quad (4.18)$$

drives the system outputs close to the desired motion profile. Moreover, the control input converges to a fixed point $u_\infty(t)$ as

$$u_\infty(t) = P^T(t)(sI + QW)^{-1}Qy_d. \quad (4.19)$$
First, we define the $L_2$-norm of the control signal as

$$\|u(t)\|^2 = \int_0^T u^T(t)u(t)dt$$

$$= \int_0^T z_k^T P(t)P^T(t)z_k dt$$

$$= z_k^T W z_k.$$  \quad (4.20)

Since $W$ is a positive definite matrix, $W = V^T V$, in which $V$ has independent columns, leads to $z_k^T W z_k = \|V z_k\|^2$. Thus, $\|u_k(t)\| \leq \|V\| \|z_k\|$. Consequently, the convergence of the control signal is guaranteed from the convergence of the $z_k$ learning algorithm. In this case, the converged vector of $z_k$ is achieved from (4.13),

$$ ((r + s) I + QW) z_\infty = r z_\infty + Q y_d; \quad (4.21)$$

or equivalently,

$$z_\infty = (s I + QW)^{-1} Q y_d. \quad (4.22)$$

Hence, the converged input is

$$u_\infty(t) = P^T(t)(s I + QW)^{-1} Q y_d. \quad (4.23)$$

### 4.2.3 Control Performance

The performance of the controller depends on the steady state value of error $e_\infty$, such that

$$e_\infty = y_d - \int_0^T P(t)u_\infty(t)dt$$

$$= y_d - W (s I + QW)^{-1} Q y_d. \quad (4.24)$$
Hence, we can conclude that the steady state error does not depend on the parameter $R$; i.e., the performance of the controller and the rate of convergence are unrelated. And the smallest possible error at all terminal points $e_{\infty} = 0$ requires that $s = 0$.

### 4.3 Optimal ILC for Discrete-time Systems

In this section, we apply our tracking approach for discrete-time systems. Let us consider the equivalent linear discrete-time invariant system

$$
\begin{align*}
x_k(t+1) &= Ax_k(t) + Bu_k(t) + \omega_k(t) \\
y_k(t) &= Cx_k(t) + \nu_k(t)
\end{align*}
$$

(4.25)

where $t = 0, 1, 2, \ldots, N - 1$.

Similarly to the previous section, the errors of given points in the $k$-th iteration are computed as

$$
e_k(t_i) = y_d(t_i) - C \sum_{j=0}^{t_i-1} A^{t_i-j-1} Bu_k(j).
$$

(4.26)

Then, formulating the $N$-sample sequence of inputs in a super-vector framework:

$$
u_k = \begin{bmatrix} u_k^T(0) & u_k^T(1) & \ldots & u_k^T(N-1) \end{bmatrix}^T,
$$

and by introducing $g_i(t)$

$$
g_i(t) = \begin{cases} 
CA^{t_i-t-1}B & \text{if } t < t_i \\
0 & \text{if } t \geq t_i
\end{cases},
$$

(4.27)

the output at the $i$-th time instant is expressed as

$$
y_k(t_i) = \sum_{t=0}^{N-1} g_i(t) u_k(t)
$$

(4.28)
Then, by rewritten $g_i$ as
\[ g_i = \begin{bmatrix} g_i(0) & g_i(1) & \cdots & g_i(N-1) \end{bmatrix}^T, \]
we obtain $y_k(t_i) = g_i^T u_k$. As a result, the cost function for the problem of tracking multiple terminal points $t_1, t_2, \ldots, t_M$ in the discrete time model is given as
\[
J = \sum_{i=1}^{M} (y_d(t_i) - g_i^T u_k)^T q_i (y_d(t_i) - g_i^T u_k+1) + u_{k+1}^T S u_{k+1} + (u_{k+1} - u_k)^T R (u_{k+1} - u_k) \tag{4.29}
\]
where $R, S = (rI, sI)$, $q_i$ are positive definite matrices.

Next, define
\[
\begin{align*}
y_d &= \begin{bmatrix} y_d^T (t_1) & y_d^T (t_2) & \cdots & y_d^T (t_M) \end{bmatrix}^T \\
G &= \begin{bmatrix} g_1^T & g_2^T & \cdots & g_M^T \end{bmatrix}^T,
\end{align*}
\]
then the cost function (4.29) can be rewritten as
\[
J = (y_d - G u_{k+1})^T Q (y_d - G u_{k+1}) + u_{k+1}^T S u_{k+1} + (u_{k+1} - u_k)^T R (u_{k+1} - u_k). \tag{4.30}
\]
Note that the controller in the $(k+1)$-th trial is achieved from the condition $\delta J/\delta u_{k+1} = 0$, or
\[
-G^T Q (y_d - G u_{k+1}) + R (u_{k+1} - u_k) + S u_{k+1} = 0.
\]
If we, as before, let $u_k = G^T z_k$, then
\[
-Q (y_d - G G^T z_{k+1}) + r (z_{k+1} - z_k) + s z_{k+1} = 0. \tag{4.31}
\]
Hence, (4.31) is an iterative learning algorithm, i.e.,

\[(r + s)I + QW_d) z_{k+1} = (rI + QW_d) z_k + Qe_k,\]  

(4.32)

where \(W_d = GG^T\) is a symmetric positive definite matrix.

From Lemma (4.2.1), matrix \((r + s)I + QW_d\) is nonsingular. Therefore, by defining

\[L_z = ((r + s)I + QW_d)^{-1}(rI + QW_d),\]

and

\[L_e = ((r + s)I + QW_d)^{-1}Q,\]

we can provide the following theorem regarding the ILC control algorithm for discrete-time systems.

**Theorem 6** For the linear discrete-time system (4.25), the ILC system

\[
\begin{cases}
    u_k &= G^T z_k \\
    z_{k+1} &= L_z z_k + L_e e_k
\end{cases}
\]

(4.33)

drives the system outputs close to the desired motion profile. Moreover, the control input converges to a fixed point \(u_\infty\) as

\[u_\infty = G^T (sI + QW_d)^{-1} Qy_d,\]  

(4.34)

and the error \(e_\infty\) is defined as

\[e_\infty = y_d - W_d (sI + QW_d)^{-1} Qy_d.\]  

(4.35)

The results of Theorem (6) are obtained in the same manner as for Lemma (4.2.1) and Theorem (5).

Furthermore, if the importance of all the given waypoints are equal, \(Q = qI\), we can show the monotonic convergence of control signal in the iteration domain.
Lemma 4.3.1  If $Q$ is chosen as $Q = qI$, where $q$ is real positive, the ILC algorithms (4.33) obtain monotonic convergence of the control signal.

Given $u_{k+1} = f(u_k)$ and $z_{k+1} = g(z_k)$, then

$$f(u_1) - f(u_2) = G^T g(z_1) - G^T g(z_2)$$

$$= G^T ((r+s)I + QGG^T)^{-1} r(z_1 - z_2)$$

Since $GG^T$ is symmetric positive definite, setting $GG^T = UVU^T$, where $V$ is the positive definite diagonal matrix and $U$ is the unitary matrix. Then $U^T = V^{-1} U^T GG^T$, and

$$G^T ((r+s)I + QGG^T)^{-1} r(z_1 - z_2) = G^T U ((r+s)I + QV)^{-1} rU^T (z_1 - z_2)$$

$$= G^T UV_r U^T G(u_1 - u_2), \tag{4.36}$$

where $((r+s)I + QV)^{-1} rV^{-1} = V_1$.

Moreover, for any matrices $A, B$ and $X$;

$$\|AXB^T\| \leq \frac{1}{2} \|A^T AX + XB^T B\|. \tag{4.37}$$

It leads to

$$\|G^T UV_r U^T G\| \leq \frac{1}{2} \|U^T GG^T UV_r + V_r U^T GG^T U\|$$

Since $U^T GG^T U = V$, then

$$\|G^T UV_r U^T G\| \leq \left\|( (r+s)I + QV)^{-1} r \right\| \tag{4.38}$$

Moreover, $((r+s)I + QV)^{-1} r$ is a symmetric matrix which has the largest singular value equals its spectral radius as in the previous chapter; furthermore,

$$\bar{\sigma} \left( (r+s)I + QV \right)^{-1} r < 1 \tag{4.39}$$

- 53 –
In conclusion, the ILC algorithm is monotonic convergence since $\|f(u_1) - f(u_2)\| \leq \|u_1 - u_2\|$.

Here, we can clearly see a significant decrease of the computational analyses. In our learning algorithm, vector $z_k \in \mathbb{R}^M$, and $L_z$, $L_d$ are $mM \times mM$ matrices where $M$ is the number of terminal points. In comparison, the typical ILC algorithm updates the input with the system matrix $mN \times mN$. As the length of iteration increases ($N > 1000$), which is common in many applications such as robotics with a high sampling rate, the requirement of memory and time dramatically increases.

4.4 Input Constraints

Many control applications require constraints on the control signal $u_{k+1}$ and its change with time $\delta u_{k+1}$ due to physical limitations or performance requirements. Furthermore, the constraint on the change of input in the iteration domain $\Delta u_{k+1}$ should be incorporated. In this part, we will show that the proposed method can guarantee all the above constraints.

Consider the constraints as

- $u_l \leq u_{k+1} \leq u_h$
- $\Delta u_l \leq \Delta u_{k+1} \leq \Delta u_h$
- $\delta u_l \leq \delta u_{k+1} \leq \delta u_h$

where $\Delta u_{k+1} = u_{k+1} - u_k$, and $\delta u_{k+1}(t) = u_{k+1}(t - 1) - u_{k+1}(t - 2)$, $t = 1, 2, \ldots, N$. Note
that the constraints are chosen to be feasible. Here, $\delta u_{k+1}$ can be rewritten as

$$\delta u_{k+1} = Eu_{k+1}, \text{ where } E = \begin{pmatrix} I & 0 & \cdots & 0 & 0 \\ -I & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -I & I \end{pmatrix}.$$  \hspace{1cm} (4.40)

Thus, the given constraints can be combined into the following constraints equation:

$$\Xi \Delta u_{k+1} \leq \Theta$$ \hspace{1cm} (4.41)

where

$$\Xi = \begin{bmatrix} -I \\ I \\ -E \\ E \end{bmatrix}, \Theta = \begin{bmatrix} -\max\{u_l - u_k, \Delta u_l\} \\ \min\{u_h - u_k, \Delta u_h\} \\ - (\delta u_l + Eu_k) \\ \delta u_h - Eu_k \end{bmatrix};$$ \hspace{1cm} (4.42)

or equivalently,

$$\Xi G^T \Delta z_{k+1} \leq \Theta.$$ \hspace{1cm} (4.43)

Since $e_{k+1} = e_k - W_d \Delta z_{k+1}$ and $u_{k+1} = G^T (\Delta z_{k+1} + z_k)$, the problem of minimizing the cost function $J$ in (2.40) is derived as

$$\min_{\Delta z_{k+1}} \left\{ \Delta z_{k+1}^T \left( W_d^T Q W_d + (r + s) W_d \right) \Delta z_{k+1} + 2 (sz_k^T - e_k^T Q) W_d \Delta z_{k+1} \right\}$$ \hspace{1cm} (4.44)

As a result, the given minimization together with the constraints equation (4.43) represent a standard quadratic programming problem [27]. In addition, the solutions were proposed in [23] and [24].
4.5 Simulation Results

Performances of the proposed techniques are presented through an example of the tracking problem with a linear discrete-time system model to show advantages of the tracking problem without reference trajectory. Then the approach will be compared with the conventional approach.

The system is chosen as

\[
\begin{align*}
x(t+1) &= \begin{pmatrix} 0.5 & 0.035 & 0.025 \\ 0.025 & 0.6 & -0.99 \\ 0.75 & 0.03 & 0.025 \end{pmatrix} x(t) + \begin{pmatrix} 0.2 \\ 0.2 \\ 0 \end{pmatrix} u(t) \\
y(t) &= \begin{pmatrix} 1.0 & 0.1 & 1.0 \end{pmatrix} x(t)
\end{align*}
\]

which operates on interval \( t \in [0, 49] \). We select 5 points in the interval as desired points in the motion profile.

Fig.4.1 illustrates the convergences of error under our proposed ILC algorithm without any reference trajectory. Accordingly, based on suitable chosen weighting matrices, the controller produces superior performance; specifically, output signals go through, or very close to, desired given data points after some iterations. By comparison, the conventional norm optimal solution of tracking the given reference trajectory is also simulated in Fig.4.2. It can be seen that the proposed ILC law achieves significantly decrease of control energy, while increasing rate of convergence. Furthermore, it is worth stressing that here the learning parameters in the iteration domain are \( 5 \times 5 \) matrices, instead of \( 50 \times 50 \) as in the typical norm optimal ILC. The results are then also compared to different weighting matrices, Fig.4.3, to demonstrate the trade-off between the error and the energy of the control signal.
Figure 4.1: Performance of the ILC controller without reference trajectory with $q = 10$, $r = 2$, and $s = 0.01$. 
Figure 4.2: Comparison of control energy between the proposed controller and the conventional norm-optimal ILC with $q = 10$, $r = 2$, and $s = 0.01$. 
Figure 4.3: Performance with $q = 20$, $r = 5$, and $s = 0.5$. 
Chapter 5

Multiple Interval Points Tracking

5.1 Introduction

The main goal of this chapter is to study learning algorithms in the case where we allow the desired data points, in which the system is required to track, not to be fixed, but instead to vary or to fluctuate inside an interval iteratively. This problem happens naturally in many applications since the data points could not be defined exactly because of noise contaminated; for example, measurement noise, sensor noises, disturbances,... Moreover, the given data could be uncertain between different iterations. Thus, specifically this allows us to apply learning algorithms to broader class of applications.

The approach is presented through three main steps. Firstly, we formulate the new ILC problem including the uncertainties of the reference points, which is bounded by a given value. Next, the cost function in the optimization problem is derived as minimizing the worst-case function. Here, we apply optimization techniques; in effect, the control update laws are generated with adaptive learning gains. Finally, we show that we can achieve the convergence of the control signal under the uncertainty of the reference points, while the points could be fluctuated trial by trial inside a given interval.
5.2 Background

We now consider the optimal tracking interval of multiple points problem with the existence of uncertainties at reference points which are iteration varying. Here, the reference points at the $k$–th iteration is bounded in the interval $[y_d, y_d + \delta y_k]$, where the quantity of $\delta y_k$ determines the range of allowable uncertainties. More specifically, the reference points are formulated that are varied iteratively inside the given interval with $\|\delta y_d\| \leq \nu$.

First, let us define the error at the iteration $k$-th as

$$e_k = y_d - Gu_k$$  \hspace{1cm} (5.1)

Then, the error at the iteration $k+1$-th which includes the uncertainty $\delta y_{k+1}$ is given by

$$e_{k+1} = (y_d + \delta y_{k+1}) - Gu_{k+1}$$  \hspace{1cm} (5.2)

Next, the cost function accounts for uncertainties at the reference points as

$$J_2(u_{k+1}, \delta y_{k+1}) = \|y_d - Gu_{k+1} + \delta y_{k+1}\|^2_Q$$

$$\quad + \|u_{k+1}\|^2_S + \|u_{k+1} - u_k\|^2_R,$$  \hspace{1cm} (5.3)

Here, if $\delta y_k$ is a null matrix, $y_d$ is fixed and the optimization problem were given in the previous part. Otherwise, the design of control signal is equivalent to solving the min-max problem

$$\min_{u_{k+1}} \max_{\delta y_{k+1}} J_2(u_{k+1}, \delta y_{k+1}),$$  \hspace{1cm} (5.4)

subject to $\|\delta y_d\| \leq \nu$. 

- 61 -
Additionally, the constraints is known as second-order cone constraints [27]. In this paper, we deal this constraints with the equivalent quadratic constraint:

\[ \| \delta y_d \|^2 \leq \upsilon^2. \] (5.5)

From the cost function, it is notable that the maximum function of

\[ \| y_d - Gu_{k+1} + \delta y_{k+1} \|^2_Q \]

is convex in \( u_{k+1} \) since it is convex in \( u_{k+1} \) for any given \( \delta y_{k+1} \) and positive definite matrix \( Q \). Moreover, the functions

\[ \| u_{k+1} \|^2_S + \| u_{k+1} - u_k \|^2_R \]

is also strictly convex in \( u_{k+1} \) with positive definite \( R \) and \( S \). As a result, the problem has a unique and global minimum \( u_{k+1} \).

5.3 The Optimization Problem

We now solve maximization problem first. Obviously, we just need to consider maximizing the cost function

\[ \max_{\delta y_d} \left\{ J_3 (\delta y_{k+1}) = \| y_{k+1} - Gu_{k+1} + \delta y_{k+1} \|^2_Q \right\} \] (5.6)

subject to \( \| \delta y_{k+1} \| \leq \upsilon \).

5.3.1 The Maximization Problem

It can be seen that the cost \( J_3 (\delta y_{k+1}) \) is convex in \( \delta y_d \), so that the maximum over \( \delta y_{k+1} \) is achieved at the boundary, \( \| \delta y_{k+1} \| = \upsilon \). We can therefore now introduce a Lagrange
multiplier $\lambda$, the solution of (5.6) is then replaced by the unconstrained problem:

$$\max \left\{ \| y_d - Gu_k + \delta y_{k+1} \|^2_Q - \lambda \left( \| \delta y_{k+1} \|^2 - \upsilon^2 \right) \right\}$$

Note that since the original problem has an inequality constraint, the Lagrange multiplier must be nonnegative $\lambda \geq 0$. From the Karush-Kuhn-Tucker (KKT) conditions, the optimal solutions are given by differentiating the function with respect to $\delta \tilde{y}_{k+1}$ and $\lambda$, and denoting the optimal solutions by $\delta \bar{y}_{k+1}$ and $\bar{\lambda}$, we obtain the equations:

$$Q (y_d - Gu_k) = (\bar{\lambda} I - Q) \delta \bar{y}_{k+1} \quad (5.7)$$

and

$$\| \delta \tilde{y}_d \|^2 = \upsilon^2. \quad (5.8)$$

Moreover, to guarantee that they are optimal solutions, we need to check whether the cost function is concave. This condition results in the negative semi-definite of the function’s Hessian matrix, which means $(\bar{\lambda} I - Q) \geq 0$, or equivalently,

$$\bar{\lambda} \geq \| Q \|. \quad (5.9)$$

For the selection of $Q$ as above, $\delta \bar{y}_{k+1}$ is calculated by

$$\delta \bar{y}_{k+1} = (\bar{\lambda} I - Q) \dagger Q (y_d - Gu_k). \quad (5.10)$$

where $X \dagger$ denotes the pseudo-inverse of $X$. And $\bar{\lambda}$ is solution of the following equation:

$$(y_d - Gu_k)^T Q (\bar{\lambda} I - Q) \dagger (\bar{\lambda} I - Q) \dagger Q (y_d - Gu_k) = \upsilon^2 \quad (5.11)$$

Thus, the optimal solutions $\delta \bar{y}_{k+1}$ and $\bar{\lambda}$ are dependence on $u_{k+1}$. Finally, by replacing the optimal solutions, we get the maximum of $J_3 (\delta \bar{y}_{k+1})$ as

$$M (u_{k+1}) = (y_d - Gu_k)^T Q \left( I + (\bar{\lambda} I - Q) \dagger Q \right) (y_d - Gu_k) + \bar{\lambda} \upsilon^2 \quad (5.12)$$
where $\tilde{\lambda}$ satisfies the condition (5.10).

### 5.3.2 The Equivalent Minimization Problem

Now, we will show that the maximal function $M(u_{k+1})$ is the result of the equivalent two variables problem

\[
\min_{\lambda \geq \|Q\|} H(u_{k+1}, \lambda)
\]

(5.13)

where

\[
H(u_{k+1}, \lambda) = (y_d - Gu_{k+1})^T Q \left( I + (\lambda I - Q)^\dagger Q \right) (y_d - Gu_{k+1}) + \lambda u^2
\]

(5.14)

and $\lambda$ is an independent variable. The reason is that by differentiating $H(u_{k+1}, \lambda)$ to find the optimal $\lambda$, we will come to exactly the same equation of $\lambda$ as (5.11). Moreover, its second differential is positive definite.

As a consequence, the maximal function $M(u_{k+1})$ is the result of the equivalent problem (5.13). The original problem (5.3) is therefore now equivalent to

\[
\min_{u_{k+1}} \min_{\lambda \geq \|Q\|} \left\{ H(u_{k+1}, \lambda) + \|u_{k+1}\|^2_S + \|u_{k+1} - u_k\|^2_R \right\}
\]

In other way, we could rewritten as

\[
\min_{\lambda \geq \|Q\|} \min_{u_{k+1}} J_2(u_{k+1}, \lambda)
\]

where

\[
J_2(u_{k+1}, \lambda) = H(u_{k+1}, \lambda) + \|u_{k+1}\|^2_S + \|u_{k+1} - u_k\|^2_R
\]
For the new min-min problem, we first seek the optimal control signal $u_{k+1}$:

$$
\min_{u_{k+1}} \left\{ H(u_{k+1}, \lambda) + \|u_{k+1}\|^2_S + \|u_{k+1} - u_k\|^2_R \right\}
$$

(5.15)

The optimal control input is achieved as the last part by differentiating the cost function with respect to $u_{k+1}$ as

$$
-G^T Q_1 (y_d - Gu_{k+1}) + R(u_{k+1} - u_k) + Su_{k+1} = 0
$$

(5.16)

where $Q_1$ is defined by

$$
Q_1 = Q \left( I + (\lambda I - Q)^\dagger Q \right),
$$

(5.17)

In other way,

$$
(G^T Q_1 G + R + S) u_{k+1} = Ru_k + G^T Q_1 y_d
$$

(5.18)

**Lemma 5.3.1** The matrix $Q_1$ is positive definite with the chosen $\lambda \geq \|Q\|$. 

To prove, the well known matrix result, which is called the matrix inverse lemma, is investigated: If matrices $A$ and $B$ are invertible, then

$$
(A + CBD)^{-1} = A^{-1} - A^{-1} C (B^{-1} + DA^{-1} C)^{-1} DA^{-1}
$$

Here, we apply this result for the matrix $Q_1$ which is derived in the equation (5.17). After some derivations, we get:

$$
Q_1^{-1} = Q^{-1} - \lambda^{-1} I
$$

(5.19)

As a result, with the chosen of $\lambda$ in the lemma, $Q_1^{-1}$ is positive definite. As a consequence, $Q_1$ is positive definite.
Likewise, $G^T Q_1 G + R + S$ is also positive definite. Rewriting (5.18) in the iterative algorithm, we obtain the optimal control signal $u_{k+1}$, which depends on $\lambda$, as

$$
\begin{align*}
    u_{k+1} &= \left( G^T Q_1 G + R + S \right)^{-1} Ru_k \\
    &+ \left( G^T Q_1 G + R + S \right)^{-1} P^T Q_1 y_d 
\end{align*}
$$

(5.20)

Secondly, for the given optimal $u_{k+1}$, the next step is to find the optimal $\bar{\lambda}$ as solution of the problem

$$
\bar{\lambda} = \arg\min_{\lambda \geq \|Q\|} L(\lambda)
$$

where

$$
L(\lambda) = (y_d - Gu_{k+1})^T Q_1 (y_d - Gu_{k+1}) + \|u_{k+1}\|_S^2 + \|u_{k+1} - u_k\|_R^2 + \lambda u^2
$$

Finally, the optimization problem is achieved leading to our learning algorithm in the following section.

### 5.4 ILC Algorithm

In this part, we consider to apply the proposed analysis to the ILC tracking interval of multiple points algorithm. Settings similarly to the optimal nominal reference points tracking problem as

$$
u_{k+1} = G^T v_{k+1},$$

and then define $W = GG^T$. From (5.18), we achieve

$$
((r + s)I + Q_1 W) v_{k+1} = (rI + Q_1 W) v_k + Q_1 e_k
$$
where \( Q_1 = Q \left( I + (\lambda I - Q)^\dagger Q \right) \).

The solution of \( v_{k+1} \) is finally obtained in the ILC formulation by

\[
v_{k+1} = Ev_k + Fe_k
\]

where

\[
E = ((r + s)I + Q_1W)^{-1}(rI + Q_1W),
\]
\[
F = ((r + s)I + Q_1W)^{-1}Q_1
\]

As a result, the ILC problem can be divided into two steps. First, the optimal parameter \( \lambda_k \) is sought. After that, the ILC formulation is proposed with the updated learning gains, which are generated from \( \lambda_k \).

5.4.1 Optimized Parameters

In this stage, we define the optimal solution of \( \lambda_{k+1} \), and the corresponding weight matrix \( Q_{k+1} \), based on the measurements in the iteration \( k \). Specifically, \( \lambda_{k+1} \) is solution of the optimization problem:

\[
\lambda_{k+1} = \arg \min_{\lambda \geq \|Q\|} L(\lambda) \tag{5.21}
\]

where

\[
L(\lambda) = \|y_d - W(Ev_k + Fe_k)\|_{Q_1}^2 + \|G^T(Ev_k + Fe_k)\|_S^2 + \|G^T(Ev_k + Fe_k - v_k)\|_R^2 + \lambda v^2
\]
We thus see that the solution of the optimal nonnegative scalar parameter $\bar{\lambda}$ corresponds to the minimizing argument of the function over the semi-open interval $[\|Q\|, \infty)$. It is worth pointing that $u_k, y_d$ and all the weighting matrices are known from previous iterations. This indicates that the determination of can be sought via optimization routines such as steepest descent and Newton method.

### 5.4.2 Optimized ILC and Convergence Properties

As a result, we achieve the optimal ILC algorithm as

$$
\mathbf{v}_{k+1} = \mathbf{E}_{k+1}\mathbf{v}_k + \mathbf{F}_{k+1}\mathbf{e}_k
$$

where

$$
\mathbf{Q}_{k+1} = \mathbf{Q}\left(\mathbf{I} + (\bar{\lambda}_{k+1}\mathbf{I} - \mathbf{Q})^\dagger\mathbf{Q}\right),
$$

and $\mathbf{E}_{k+1} = \mathbf{E}(\mathbf{Q}_1 = \mathbf{Q}_{k+1}), \mathbf{F}_{k+1} = \mathbf{F}(\mathbf{Q}_1 = \mathbf{Q}_{k+1})$.

When the optimized learning gains are given, we finally come to the main results:

**Theorem 7** For the linear discrete-time system (1.1), and the uncertainties iteration varying in the reference data points $\delta y_d$, such that $\|\delta y_d\| \leq \nu$, then the ILC system

$$
\begin{cases}
\mathbf{u}_k &= \mathbf{G}^T\mathbf{v}_k \\
\mathbf{v}_{k+1} &= \mathbf{E}_{k+1}\mathbf{v}_k + \mathbf{F}_{k+1}\mathbf{e}_k
\end{cases}
$$

(5.23)

drives the system outputs close to the reference points.

For the proof of convergence, we first consider the case when $\bar{\lambda}_{k+1} = \|Q\| = q$, which leads to $\mathbf{Q}_{k+1} = \mathbf{Q}$ from (5.22). Consequently, this chosen denotes that the uncertainties are not
available, i.e \( \nu = 0 \). Thus, \( \hat{\lambda}_{k+1} = q \) is the non-optimal solution of the optimization problem. Hence, we have the inequality:

\[
J_2 \left( u_{k+1}, \hat{\lambda}_{k+1} \right) \leq J_2 \left( u_{k+1}, q \right)
\]  

(5.24)

where

\[
J_2 \left( u_{k+1}, q \right) = (y_d - Gu_{k+1})^T Q (y_d - Gu_{k+1}) + \|u_{k+1}\|_S^2 + \|u_{k+1} - u_k\|_R^2.
\]  

(5.25)

with \( q \) is constant.

Now consider minimizing \( J_2 \left( u_{k+1} \right) \) with respect to \( u_{k+1} \), we can see that the choice of \( u_{k+1} = u_k \) is a non-optimal solution. It leads to the following relationship:

\[
J_2 \left( u_{k+1}, \hat{\lambda}_{k+1} \right) \leq \|e_k\|_Q^2 + \|u_k\|_S^2.
\]  

(5.26)

Finally, the benefit of the analyses are utilized by the interlacing result

\[
\|e_{k+1}\|_Q^2 + \|u_{k+1}\|_S^2 \leq J_2 \left( u_{k+1}, \hat{\lambda}_{k+1} \right) \leq \|e_k\|_Q^2 + \|u_k\|_S^2.
\]  

(5.27)

The result states that the algorithm achieve stability condition in the iteration domain. Also, equality holds if, and only if all the following conditions are satisfied: \( u_{k+1} = u_k \), \( Q_{k+1} = Q \), and \( \nu = 0 \), i.e when there is no more uncertainties and the convergence of nominal reference points are achieved.

**Remark 5.4.1** Comparing to the previous norm optimal ILC approach without uncertainties, the optimal control signal algorithm is distinct in the weighting matrices. More specifically, the weighting matrices \( Q \) is replaced by corrected matrix \( Q_k \), and these corrections are...
defined by the optimal parameter \( \lambda_k \). Hence, this algorithm is a type of adaptive iterative learning control which is typical by the iteration varying learning gains [28].

**Remark 5.4.2** From the equation (5.27), we can see that the amount of the given bounded uncertainties \( \nu \) and the weight matrix \( Q \) decide the convergence and performance of the algorithm. The smaller \( \nu \), the better performance is achieved. Especially, if \( \nu = 0 \), we can achieve the same monotonic convergence result as in our previous part without uncertainties.

**Remark 5.4.3** One of the advantages of this approach is that we could consider the iteration varying amount of uncertainties boundedness \( \nu_k \), which could bring a better performance at a certain trial.

**Remark 5.4.4** The proposed approach could be deal in the same manner with the iteration varying disturbance and iteration varying initial conditions since the system could be modeled as \( y_k = G u_k + d_k \), where \( d_k \) is the varying factor.

### 5.5 Simulation Results

Performances of the proposed techniques to deal with interval point tracking problem which includes uncertainties at reference points are presented through an example of the tracking problem with a linear discrete-time system model. The system is chosen as

\[
\begin{align*}
    x(t+1) &= \begin{pmatrix} 0.2 & 0.3 & 0.1 \\ 0.025 & 0.6 & -1.0 \\ 0.75 & 0.3 & 0.025 \end{pmatrix} x(t) + \begin{pmatrix} 0.1 \\ 0.1 \\ 0.0 \end{pmatrix} u(t) \\
    y(t) &= \begin{pmatrix} 1.0 & 0.0 & 1.0 \end{pmatrix} x(t)
\end{align*}
\]
which operates on interval $t \in [0, 49]$. We select 5 points in the interval as desired points in the motion profile. The weight matrices are selected as $Q = 10I_{5 \times 5}$, $R = 2I_{50 \times 50}$, and $S = 0.01I_{50 \times 50}$.

Simulations present the performance of tracking multiple points where there is existence of non-repetitive uncertainties at the reference points. In practice, the non-repetitive uncertainties are often stochastic signals, e.g. noise. Here, we consider the noise signals at the reference points have a uniform distribution in the interval $[-0.01, 0.01]$; in effect, $\nu = 0.2236$. The validity of our proposed approach for non-repetitive uncertainties are simulated in the Figure 5.1 and Figure 5.2. Firstly, we show the convergence of errors in the iteration, which is bounded in the given interval. The maximum error is computed by the summation of $\nu$ and the converged error in the previous part. Furthermore, the updated parameters of $\lambda_k$ and $q_1$ are given in the Figure 5.2.
Figure 5.1: Errors and control energy of the interval tracking problem
Figure 5.2: Updated parameters $\lambda_k$ and $q_1$
Chapter 6

Terminal Iterative Learning Control for Thermoforming Machine

6.1 Introduction and Motivations

Today, thermoforming has become one of the fastest growing methods of processing plastics. Thermoforming is an industrial process in which plastic sheets are heated and then formed into specific parts. The process is generally divided into three phases: heating, forming and cooling. The sheet is heated in an oven until it becomes pliable; and then the softened sheet is formed over the mold and cooled until it hardens. With the development of thermoforming technique, numerous theoretical research on thermoforming control has been carried out. Various works have been done on the design of efficient in cycle control; currently, the temperature control is mainly based on traditional PID control of the heater temperatures. On the other hand, the adjustment of the temperature setpoint is made by trial and error. However, the research to investigate the repetitive nature of the system has not been focused. Since the repetitive production of sheets meeting the standards is very demanding in the industry, the implementation of iterative control on the machine has naturally become a potential research topic [29] - [32].

The development of heating controller will benefit the thermoforming industry in several
different ways. Firstly, the quality of the part will improve due to the better energy distribution within the sheet that leads to less rejected parts, and the material cost per part will then decrease. Similarly, the time needed to produce a sheet with the desired temperature distribution will be optimized, thus increasing the number of parts that can be produced per shift. Secondly, other benefits to industry derived from this controller include the minimization of the energy input to the thermoforming oven. By controlling the energy entering the sheet, an optimal oven temperature setting can be determined in order to lower the heating costs. The machine itself may also require less maintenance and repair work. Heating elements become less efficient as they age, and the controller will compensate for this loss of efficiency by directing the adjoining elements to heat more. The end result of all of these benefits is a decrease in production cost per part, an important savings for the thermoforming industry. These reasons motivate us to apply terminal ILC to this application.

Generally, the adjustment of the heater temperature setpoints is performed manually by trial and error. When using high-density polyethylene, this adjustment can be made relatively easily since there is a quite large margin around the required sheet temperature for molding. On the other hand, other plastics, for example nylon, have a narrower temperature margin, which makes heater temperature setpoint adjustment more difficult. Another challenge is the effect of ambient temperature variation inside the oven. Since a part of the energy exchange with the plastic sheet comes from convection, variation in the ambient air temperature can cause a drift in the plastic sheet temperature at the end of the heating cycle. Therefore, the adjustment of the heater temperature setpoint could be different at different times in a day. As a result, our research is to apply ILC for improving the control of thermoforming machine
temperature setpoint by learning from repeated operations. Obviously, to fulfill the TILC controller requirement, sensors were added to measure the temperature surface of plastic sheet. Finally, our settings and control technique will help to improve the temperature control of thermoforming machines performance by adjusting iteratively the heater temperature setpoint such that the surface temperature of the plastic sheet meets a desired temperature profile at the end of the heating cycle.

6.2 Modeling

The reheat phase of a thermoforming machine can be modeled using thermodynamic theory. The heating of the plastic sheet is done via radiant heaters located above and below the plastic sheet. The radiant energy coming from the heater hit the sheet surface and transmits energy from the heater to the sheet surface. The convection energy transfer occurs at the sheet surface and corresponds to an exchange of energy between the ambient air and
the plastic sheet surface. Finally, the energy exchange inside the plastic sheet is done by heat conduction and radiant energy penetrating inside the plastic sheet. Figure 6.2 shows the energy exchange. In the model of the heating phase of thermoforming, the plastic sheet is divided into zones, one for each sensor pair. To analyze the temperature behavior throughout the plastic sheet, each zone is divided into layers, and each layer has a node.

In the model, the plastic sheet is divided into $S$ zones, while the thickness is divided into $N$ layers each with a corresponding node. Nodes are evenly spaced by a distance $\Delta z$. Node 1, is located at the upper surface of the sheet while node $N$ is at the lower surface. Each internal node is located at the middle of the layer [32]. For a given zone, the model at the surface of the plastic sheet is expressed by

$$\frac{dT_1}{dt} = \frac{2}{\rho V C_p} \left( Q_{RT_1} + Q_{RB_1} + Q_{CT} + Q_{K1} \right)$$

where $T_1$ is the temperature of the top surface of the sheet, $\rho$ is the density of the plastic
Figure 6.3: IR temperature sensors and the corresponding zone [34]

Figure 6.4: Layers and nodes [34]
sheet, $V$ is the volume of layer, and $C_p$ is the specific heat of the plastic sheet. The radiative energy exchange is contained in $Q_{RT_1}$ and $Q_{RB_1}$. The energy coming from the $M$ top heaters is:

$$Q_{RT} = \sigma \varepsilon_{eff} A_h \sum_{i=1}^{M} F_i (\theta_i^4 - T_1^4),$$

and the energy coming from the $M$ bottom heaters:

$$Q_{RB} = \sigma \varepsilon_{eff} A_h \sum_{i=M+1}^{2M} F_{i+6} (\theta_{i+6}^4 - T_N^4)$$

where $\sigma$ is the Stefan Boltzmann constant, $\varepsilon_{eff}$ is the effective emissivity, $A_h$ is the area of the heater, $F_i$ is the view factor of the $i$-th heater, $\theta_i$ is the temperature of the $i$-th heater and $T_N$ is the temperature of the bottom surface of the plastic sheet.

The relation between $Q_{RT}$ and $Q_{RT_1}$ is defined from the Beer Lambert law. According to this law, the transmissivity of a material depend on the spectral absorption coefficient of the material and the material thickness. This law is stated as $T = e^{-Az}$, where $A$ is the average absorptivity of the material across its spectrum and $z$ the thickness of the material. The absorbed fraction of radiant energy in a layer is equal to the received energy minus the transmitted fraction, then

$$\beta(z) = 1 - T(z) = 1 - e^{-Az}$$

For an external layer we have:

$$\beta_1(z) = \beta(\Delta z/2) = 1 - e^{-A\Delta z/2}$$

and for an internal layer:

$$\beta_2(z) = \beta(\Delta z) = 1 - e^{-A\Delta z}$$

- 79 -
Thus, the heat absorption of radiant energy $Q_{RT1}$ is only a fraction of $Q_{RT}$ as $Q_{RT1} = \beta_1 Q_{RT}$, and similarly for the relation between $Q_{RB}$ and $Q_{RB1}$ as

$$Q_{RB1} = \beta_1 (1 - \beta_1) (1 - \beta_2)^{N-1} Q_{RB}$$

The convective energy exchange is expressed by $Q_{CT}$ as

$$Q_{CT} = h(T_\infty - T_1)$$

where $h$ is the convection factor and $T_\infty$ is the ambient air temperature near the surface of the plastic sheet. The conductive energy exchange is given by

$$Q_{K1} = -\frac{kA}{\Delta z} (T_1 - T_2)$$

where $k$ is the heat conduction parameter of the plastic sheet, $A$ is the surface of the plastic sheet zone, $\Delta z$ is the layer thickness and $T_2$ is the temperature of the layer 2. Similar equations apply to the bottom layer $N$.

Inside the plastic sheet, the model is given as

$$\frac{dT_j}{dt} = \frac{1}{\rho V C_p} \left( Q_{RTj} + Q_{RBj} + Q_{Kj} \right)$$

And the radiative terms are:

$$Q_{RTj} = \beta_2 (1 - \beta_1) (1 - \beta_2)^{j-2} Q_{RT}$$

$$Q_{RBj} = \beta_2 (1 - \beta_1) (1 - \beta_2)^{N-j-1} Q_{RB}$$

The conductive energy exchange, for internal layers becomes:

$$Q_{Kj} = -\frac{kA}{\Delta z} (T_{j-1} - 2T_j + T_{j+1})$$
Finally, at the top surface of the plastic sheet, node 1, in zone k, the temperature dynamic is expressed by

\[
\frac{dT_{k,1}}{dt} = \frac{2}{\rho VC_p} \left\{ \beta_1 Q_{RT_k} + \beta_1 (1 - \beta_1)(1 - \beta_2)^3 Q_{RB_k} + h(T_{oo} - T_{k,1}) + \frac{kA}{\Delta z} (T_{k,2} - T_{k,1}) \right\}
\]

Within the large brackets, the first two terms represents the conduction heat transfer and the convection heat transfer, respectively; the final term represents a combination of radiation heat transfer, energy absorption and energy transmission. Energy is absorbed both from the top and the bottom of the sheet, as the system has radiant energy sources acting on both sides of the sheet. Therefore, there are both upper and lower absorption terms.

Considering when the sheet is divided into 6 equal zones, each with 5 layers, making a total of 30 different locations at which the temperature will be calculated [31]. Each of these is defined as a state of the system. There are 12 sets of heating elements, each with its own PID controller. The outputs can be selected as desired, by choosing the entries of matrix C. Then, the temperature at any point within the sheet, or on the sheet surface, can be examined during the entire heating cycle. The dynamic equation of the bottom node surface of the plastic sheet:

\[
\frac{dT_{k,5}}{dt} = \frac{2}{\rho VC_p} \left\{ \beta_1 (1 - \beta_1)(1 - \beta_2)^3 Q_{RB_k} + h(T_{oo} - T_{k,5}) + \frac{kA}{\Delta z} (T_{k,4} - T_{k,5}) \right\}
\]

For the internal nodes i (of zone k), located inside the plastic sheet, we have the following dynamic:

\[
\frac{dT_{k,i}}{dt} = \frac{1}{\rho VC_p} \left\{ \beta_2 (1 - \beta_1) [(1 - \beta_2)^{i-2} Q_{RT_k} + (1 - \beta_2)^{i-4} Q_{RB_k}] + \frac{kA}{\Delta z} (T_{k,i-1} - 2T_{k,i} + T_{k,i+1}) \right\}
\]

The radiant terms are defined for the top heaters as

\[
Q_{RT_k} = \sigma \varepsilon_{eff} A_h \sum_{j=1}^{6} F_{kj} (\theta_j^4 - T_{k,1}^4)
\]

-81-
and for the bottom heaters as

$$Q_{RB_k} = \sigma \varepsilon_{eff} A_h \sum_{j=7}^{12} F_{kj} \left( \theta_j^4 - T_{k,5}^4 \right)$$

Now, we define new variables for simplicity as

$$a = \frac{1}{\rho \varepsilon f A_h \Delta x}$$

$$b = \frac{k}{\Delta x}$$

$$c_j = \sigma \varepsilon_{eff} \beta_2 (1 - \beta_2)^{i-2} (1 - \beta_1)$$

Finally, the state-space matrix equations for zone $i$ is defined as [31]:

$$\begin{bmatrix}
\dot{x}_1^i \\
\dot{x}_2^i \\
\dot{x}_3^i \\
\dot{x}_4^i \\
\dot{x}_5^i
\end{bmatrix} =
\begin{bmatrix}
-2a(h + b) & 2ab & 0 & 0 & 0 \\
ab & -2ab & ab & 0 & 0 \\
0 & ab & -2ab & ab & 0 \\
0 & 0 & ab & -2ab & ab \\
0 & 0 & 0 & 2ab & -2a(h + b)
\end{bmatrix}
\begin{bmatrix}
\dot{x}_1^i \\
\dot{x}_2^i \\
\dot{x}_3^i \\
\dot{x}_4^i \\
\dot{x}_5^i
\end{bmatrix}
+ \begin{bmatrix}
2ah & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 2ah
\end{bmatrix}
\begin{bmatrix}
T_{\text{top}} \\
T_{\text{bottom}}
\end{bmatrix}
+ \frac{a}{\sigma \varepsilon_{eff}}
\begin{bmatrix}
c_1 & c_5 \\
c_2 & c_4 \\
c_3 & c_3 \\
c_4 & c_2 \\
c_5 & c_1
\end{bmatrix}
\begin{bmatrix}
Q_{RT}^i \\
Q_{RB}^i
\end{bmatrix}$$

Since the state-space system representing the energy inside the sheet is non-linear in the last part, the controller options become more complicated. Nonlinear controllers often must be designed specifically for the system, and cannot be ported over to similar systems easily. A controller designed for the thermoforming machine might not achieve the same results on another thermoforming machine, if these non-linearities are not taken into account. Another
source of non-linearity is the heating elements themselves, but the inner feedback control loop can eliminate these potential problems. For this reason, it makes sense to investigate the error caused by linearizing this system about a given operating point to verify that the approximation does not differ too much from reality. From the given system, the linear approximation at the operating point in this case be \( T_{\text{set}} \), and the function \( f(T) = T^4 \). Then, the linear approximation becomes:

\[
T^4 \approx T_{\text{set}}^4 + 4T_{\text{set}}^3 (T - T_{\text{set}})
\]

\[
T^4 \approx 4T_{\text{set}}^3 T - 3T_{\text{set}}^4
\]

6.3 Simulation

The heating phase of the thermoforming process is considered here to heat the plastic sheet up to a desired surface temperature map, before the molding phase. Here, we consider the oven model has 4 independent heater zones on top that can be grouped together, for example, zones 1 and 2 are controlled as a single zone and similarly for zones 4, and 5. The same applies for the bottom heater zones. We can see a smooth behavior of the zone setpoints. The system has to compensate for variation in initial condition (here the initial surface temperature of the sheet). Linearizing the model of a thermoforming oven configured with four heater zones and four infrared temperature sensors around the operating point, we obtain [31]:

\[
P_N = \begin{bmatrix}
0.0336 & 0.3340 & 0.0121 & 0.1202 \\
0.3340 & 0.0336 & 0.1202 & 0.0121 \\
0.0121 & 0.1202 & 0.0336 & 0.3340 \\
0.1202 & 0.0121 & 0.3340 & 0.0336
\end{bmatrix}
\]
Moreover, for the simulation purpose, we suppose there are disturbances and measurement noises in each surface temperature and setpoint temperature during iterations. These disturbances are varying between \([-3, 3]\) Celsius degree as $d = \text{random('uni', -3, 3)}$ iteratively. The desired terminal points are 160, 160, 140, 140 respectively on the top and under the bottom.

In this simulation, we choose the weighting matrices as: $Q = 100I_4$, $R = 5I_4$, and $S = 0.5I_4$. The simulations show that even under the varying disturbances, we still could achieve a very good performance and speed of convergence after a few iterations. In the figure 6.8, we select different weighting matrices $Q = 100I_4$, $R = 10I_4$, and $S = I_4$, and we still achieve similar results.
Figure 6.6: Heater setpoints
Figure 6.7: Surface temperature
Figure 6.8: Surface temperature
Chapter 7

Multiple Points Satellite Antenna Iterative Learning Control

7.1 Introduction and Motivations

The X-band antennas in Korea multi-purpose satellite 3 (KOMPSAT-3) are used to communicate with ground station by transmitting the satellite images to the station. In the antenna control system, the antenna is maneuvered to direct towards the desired azimuth and elevation angles when it passes over the station. Specifically, the control system is a type of multiple points tracking control since we are given of satellite attitudes and positions that are varying as satellite rotates, and given the position of ground station that is fixed in reference coordinate frame, the directions of antennas are controlled continuously to point towards the ground station at certain sampling time. In order to achieve the pointing control requirement, the antenna must have a high rotational angular velocity and a large rotational angle.

The problem of tracking points antenna control has some main difficulties. The first obstacle is to model the antenna system such that it could be applied in real systems. As shown in Fig.7.1, the rotational inertia of the azimuth axis of an antenna is the function of its elevation angle. There are various strong interactions in the antenna system; for example, the elevation and azimuth of an antenna, the rotation of a satellite and the rotation of an
antenna, the vibration of an antenna supporting bar and the rotation of an antenna. Therefore, those factors are considered as the uncertainties in the dynamic model. Secondly, the large angular pointing control of an antenna has its direct effect on the stability of a satellite body. Variable parameters, unmodeled dynamics and strong interactions have their direct effect on the pointing accuracy and rotation speed of an antenna. For the establishment of an inter-satellite communication link, it is essential for an antenna pointing control to have high pointing accuracy and rotation speed. This is the reason why such problems as robust control, disturbance rejection control and decreasing impact to satellite body must be solved in the process of designing an antenna pointing control system.

Since the satellite rotates around the earth and thus tracking the same data points iteratively, the multiple points ILC system could be applied to deals with repetitive uncertainty problems. The other important reason that made us to consider to apply ILC technique is to deal with external disturbance, which is one of the main obstacle in antenna control pointing system.

In the satellite antenna system, the desired points are stored in a tracking profile file (TPF), which includes a sequence of desired azimuth and elevation angles of satellite antenna versus sampling time points. The set of desired azimuth and elevation angles that are generated by an optimization under some constraints is called corrected profile and the corrected profile after reconstruction is called tracking profile. The TPF is generated from a geometrical relationship among the satellite attitudes, orbit, and the position of ground station with respect to a given reference coordinate frame. The TPF is then finally uploaded to satellite as a mission file of satellite. It is notable that the changes and change rates of point-
ing directions of the antennas are restricted by mechanical or electrical constraints; thus, the sequence of desired azimuth and elevation angles should be determined in an optimization sense. In real system, it is necessary to validate whether the satellite antenna can direct towards the desired points of TPF. Though the TPF is uploaded to satellite, if the antenna cannot direct towards the desired azimuth and elevation angles accordingly, then the satellite images cannot be down-linked to the ground station at targeted time instants. Thus, it is essential to validate whether the desired azimuth and elevation angles of TPF can be achievable on the spaceborne satellite.

The other issues in the control task is to check whether the control mechanism of antennas has enough control ability to follow the desired azimuth and elevation angles. To this objective, it is important to have a wide control bandwidth; otherwise, if the control bandwidth is smaller than the frequency of TPF samples, the desired target cannot be achieved. Here, the central technical issue is associated to both dynamics of antenna mechanism, constraints ILC and external disturbance.

This chapter is dedicated to the satellite antenna pointing application. In particular, we introduces the technique which is used to compute desired azimuth and elevation angles. Next, the current method of trajectory planning and tracking control are reviewed. Finally, we apply our multiple points ILC techniques to an antenna system and then verifying by simulations.
Figure 7.1: X-band antenna azimuth and elevation motion [35]

Figure 7.2: Satellite antenna pointing
7.2 Desired Azimuth and Elevation Angles Computation

This section explains how the desired azimuth and elevation angles are computed from the given orbit and attitude of satellite w.r.t. the earth-centered fixed (ECF) frame. The set of desired angles is called preliminary tracking profile. Here, we suppose that the following information is given: satellite attitude w.r.t. the ECF frame, $C_{Body}^{ECF}$; the orientation of antenna in the body frame, $C_{Ant}^{Body}$; satellite position w.r.t. ECF, $p_{ECF}$; the position of targeted ground station w.r.t. ECF, $p_{ECF}^g$; and the position of projected nadir point, $p_{ECF}^{nadir}$ w.r.t. ECF.

Specifically, the task is to find the desired orientation angle of X-band antenna w.r.t. the body coordinate frame, $C_{Ant}^{Body}$, such that the antenna is maneuvered to point toward the ground station, within an off-pointing margin which is inside the interval bounded by $r$. Fig.7.3 depicts the pointing of satellite antenna toward the ground station, where $\Phi$ is the longitude of the satellite, $\lambda$ is the latitude of the satellite, and $G$ represents the point of ground station. In the picture, $\Phi_a$ and $\lambda_a$ are longitude and latitude of projected nadir point of satellite, respectively, and $\Phi_g$ and $\lambda_g$ are longitude and latitude of the ground station, respectively. The problem is to maneuver the satellite antenna such that it points toward the ground station, but taking account of satellite orbit and attitude. However, the satellite orbit and attitude are controlled by attitude and orbit control system in a higher operation level, they cannot be operated for a ground tracking purpose only. As a result, we are operate the antenna by only maneuvering the azimuth and elevation angles of it on the satellite body.

Fig.7.4 shows the off-pointing margin, which can be computed by effective beamwidth of the antenna. The off-pointing margin indicates the area in which the satellite can communicate with the ground station. Thus, if the line-of-sight direction of the antenna is within the circle,
Figure 7.3: Satellite, ground station, and direction in ECF coordinate frame [15]

Figure 7.4: Off-pointing margin [15]
the satellite can then be connected to the ground station. The tracking problem now becomes
an interval pointing control problem.

The body of antenna and its orthogonal axes are shown in Fig. 7.5. It was supposed that
$C_{Body}^{Ant}$ can be computed using the azimuth and elevation information of antenna, such as

$C_{Body}^{Ant} = C_{y,\alpha}C_{z,\beta}$, where $\beta$ is azimuth of antenna, $\alpha$ is elevation of antenna, and

$$
C_{y,\alpha} = \begin{bmatrix}
\cos \alpha & 0 & -\sin \alpha \\
0 & 1 & 0 \\
\sin \alpha & 0 & \cos \alpha
\end{bmatrix};
C_{z,\beta} = \begin{bmatrix}
\cos \beta & \sin \beta & 0 \\
-\sin \beta & \cos \beta & 0 \\
0 & 0 & 1
\end{bmatrix}
$$

Then, the unit line-of-sight direction of antenna w.r.t. ECF coordinate frame is computed as

$$
l_{aECF} = C_{Body}^{ECF}C_{Body}^{Ant} [1, 0, 0]^T
$$

The unit vector represents the actual pointing direction of the antenna in the ECF frame.

Next, assuming that the offset between the antenna and the origin of the satellite body frame
is negligible compared to the distance between the satellite and the ground station, we compute the desired pointing direction of the antenna in the ECF frame using the positions of satellite and ground station such as

\[ \mathbf{l}_{ECF}^d = \frac{\mathbf{p}_{ECF}^g - \mathbf{p}_{ECF}^s}{\left\| \mathbf{p}_{ECF}^g - \mathbf{p}_{ECF}^s \right\|} \]

The task is to maneuver the antenna such that it directs the ground station (i.e. \( \mathbf{l}_{ECF}^a \rightarrow \mathbf{l}_{ECF}^d \)).

To this aim, by equalizing \( \mathbf{l}_{ECF}^a \) and \( \mathbf{l}_{ECF}^d \), we have

\[ \mathbf{C}_{Body}^{ECF} \mathbf{C}_{Ant}^{Body} \begin{bmatrix} 1, 0, 0 \end{bmatrix}^T = \mathbf{C}_{Body}^{ECF} \mathbf{C}_{Ant}^{Body} \begin{bmatrix} 1, 0, 0 \end{bmatrix}^T = \frac{\mathbf{p}_{ECF}^g - \mathbf{p}_{ECF}^s}{\left\| \mathbf{p}_{ECF}^g - \mathbf{p}_{ECF}^s \right\|} \]

which yields

\[
\begin{bmatrix}
\cos \alpha & 0 & -\sin \alpha \\
0 & 1 & 0 \\
\sin \alpha & 0 & \cos \alpha
\end{bmatrix}
\begin{bmatrix}
\cos \beta & \sin \beta & 0 \\
-\sin \beta & \cos \beta & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix}
= \mathbf{C}_{Body}^{ECF} \frac{\mathbf{p}_{ECF}^g - \mathbf{p}_{ECF}^s}{\left\| \mathbf{p}_{ECF}^g - \mathbf{p}_{ECF}^s \right\|}
= \begin{bmatrix}
\xi_x \\
\xi_y \\
\xi_z
\end{bmatrix}
\]

Finally, we can define \( \alpha \) and \( \beta \)

\[
\alpha = \sin^{-1} \left( \frac{\xi_z}{\cos \left( \sin^{-1} (-\xi_y) \right)} \right)
\]

and

\[
\beta = \sin^{-1} (-\xi_y)
\]

The angles calculated above are desired azimuth and elevation angles of antenna w.r.t. the satellite body frame. The calculated angles are preliminary tracking profile, which is further refined to generate corrected profile considering mechanical and electrical constraints.
7.3 Antenna Dynamics

Korea multi-purpose satellite 3 (KOMPSAT-3) has two X-band antennas. The X-band antennas are used for transmitting satellite images to ground station. One of two antennas or both antennas are selected to transmit the images in accordance with satellite orbit, attitude, and position of ground station. The $x$-axis of antenna vector is aligned to the $x$-axis of satellite body coordinate frame when azimuth and elevation of the antenna are at (0, 0) degrees respectively. When azimuth and elevation are at (90, 0) degrees, it corresponds to the $y$-axis of the body coordinate frame. The azimuth angle can rotate from zero to 360 degrees, while elevation angle can rotate from 14.8 degrees to 145 degrees. The antenna is driven by antenna motors, which are controlled by antenna pointing mechanism (APM). The APM is driven by antenna pointing driver (APD), which has command inputs from tracking profile file (TPF). The dynamics of antenna is determined by rigid body motion of antenna and dynamics equation of antenna driving motors. The motion of antenna is affected by flexible vibrations of antenna supporting bars and satellite motion. If the angular velocity vector of antenna body frame is denoted as $\omega = (\omega_x, \omega_y, \omega_z)^T$, then it is related to azimuth and elevation angular rates such as

$$\omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & \sin \alpha \\ 0 & 0 & -\cos \alpha \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ 0 \\ \dot{\beta} \end{bmatrix} = P(\alpha) \dot{\Theta}$$

where $\beta$ is azimuth angle, and $\alpha$ is elevation angle which are system outputs. Eventually, using motor dynamics model and antenna rigid body motion equations, the antenna dynamics
can be obtained as in the work [37] as

\[ \Theta = g(P(\alpha))V + f \]

where \( V \) is the control voltage input vector of a motor, \( g(P(\alpha)) \) is a nonlinear function of \( P(\alpha) \), and the uncertainties, disturbances are included in \( f \), which is a combination of vibration from satellite body, antenna supporting bar, perturbation from reaction wheels, torques from the antenna motors, and various modeling errors.

### 7.4 Current Trajectory Planning and Control Method

This section explains the current method of trajectory planning of the antenna for KOMP-SAT2. The current method approximates the set of the desired target points, which is composed of the desired azimuth and elevation angles by a polynomial. Specifically, given the desired azimuth and elevation angles \( \alpha^d(t) \), \( \beta^d(t) \) at the time of \( t \), we approximate them by polynomial functions such as

\[
\beta^d(t) = a_0 + a_1 t + \ldots + a_n t^n
\]

\[
\alpha^d(t) = b_0 + b_1 t + \ldots + b_n t^n
\]

If there are desired azimuth and elevation angles at discrete time points \( t_0, t_1, \ldots, t_m \), then the coefficients of the polynomials are computed by solving the following equality by a least-square method:

\[
\begin{bmatrix}
1 & t_0 & t_0^2 & \ldots & t_0^n \\
1 & t_1 & t_1^2 & \ldots & t_1^n \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & t_m & t_m^2 & \ldots & t_m^n
\end{bmatrix}
\begin{bmatrix}
a_0 \\
a_1 \\
\vdots \\
a_n
\end{bmatrix}
= \begin{bmatrix}
\beta^d(t_0) \\
\beta^d(t_1) \\
\vdots \\
\beta^d(t_m)
\end{bmatrix}
= \begin{bmatrix}
\alpha^d(t_0) \\
\alpha^d(t_1) \\
\vdots \\
\alpha^d(t_m)
\end{bmatrix}
\]
Then, the antenna is maneuvered to follow the given polynomials, whose coefficients are generated by solving the above equation. Note that the current method is good in providing a smooth trajectory; however, it is a trivial method and does not consider mechanical constraints of the APS. After generating the smooth trajectory, we had to check whether the generated profile satisfies the constraints or not. Thus, the current method is highly dependent on a manual tuning and operators empirical knowledge. Moreover, from the perspective of computational cost, it takes much control effort to create a reference trajectory, especially with a long operation system as satellite antenna. As mentioned, our approach overcomes this problem by only considering the critical points rather than the whole reference polynomials.

7.5 Simulations

The desired azimuths and elevations are calculated as the above method and was given in our previous work [15]. For the purpose of simulation, we borrow the linear state-space
system of the elevation dynamic system from [38] to our demonstration as

\[
A = \begin{bmatrix}
1 & 0.0007 & -0.0310 & 0.0345 & 0.0235 & 0.0272 & -0.0088 & -0.0072 & 0.0205 \\
0 & 0.9963 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.9575 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.9779 & 0.1383 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.1383 & 0.9779 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.9658 & 0.1964 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.1964 & 0.9658 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9504 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2926 & 0.9504
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
0.0003 & -0.0048 & -0.0251 & -0.0259 & -0.0061 & 0.0023 & 0.0168 & 0.0041 & -0.0062 \\
1.0090 & 0.1839 & 0.0133 & -0.0890 & 0.2196 & -0.0411 & 0.0632 & -0.0660 & 0.0395
\end{bmatrix}^T
\]

\[
C = \begin{bmatrix}
1.0090 & 0.1839 & 0.0133 & -0.0890 & 0.2196 & -0.0411 & 0.0632 & -0.0660 & 0.0395
\end{bmatrix}
\]

In the simulation, we consider the disturbances are nonrepeated and random, but bounded. The weight matrices are chosen as \( q = 10^4, r = 0.02, s = 10^{-4} \). It can be seen from the simulations that the outputs track the desired data points in the iteration domain. After 25 iterations, the output converges to the reference trajectory. For comparison, the last figure shows the performance with different weight matrices \( q = 5 \times 10^4, r = 0.01, s = 10^{-3} \).
Figure 7.6: Errors at points
Figure 7.7: Control signals and output of angles in the iteration domain with $q = 10^4$, $r = 0.02$, and $s = 10^{-4}$
Figure 7.8: Control signals and output of angles in the iteration domain with $q = 5 \times 10^4$, $r = 0.01$, and $s = 10^{-3}$.
<table>
<thead>
<tr>
<th>UTC</th>
<th>Azimuth (degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20090713011705</td>
<td>4.3148538e+001</td>
</tr>
<tr>
<td>20090713011710</td>
<td>4.3229475e+001</td>
</tr>
<tr>
<td>20090713011715</td>
<td>4.3310690e+001</td>
</tr>
<tr>
<td>20090713011720</td>
<td>4.3392182e+001</td>
</tr>
<tr>
<td>20090713011725</td>
<td>4.3473948e+001</td>
</tr>
<tr>
<td>20090713011730</td>
<td>4.3555993e+001</td>
</tr>
<tr>
<td>20090713011735</td>
<td>4.3638314e+001</td>
</tr>
<tr>
<td>20090713011740</td>
<td>4.3720915e+001</td>
</tr>
<tr>
<td>20090713011745</td>
<td>4.3803793e+001</td>
</tr>
<tr>
<td>20090713011750</td>
<td>4.3886951e+001</td>
</tr>
<tr>
<td>20090713011755</td>
<td>4.3970387e+001</td>
</tr>
<tr>
<td>20090713011760</td>
<td>4.4054102e+001</td>
</tr>
<tr>
<td>20090713011765</td>
<td>4.4138100e+001</td>
</tr>
<tr>
<td>20090713011870</td>
<td>4.4222377e+001</td>
</tr>
<tr>
<td>20090713011875</td>
<td>4.4306933e+001</td>
</tr>
<tr>
<td>20090713011880</td>
<td>4.4391772e+001</td>
</tr>
<tr>
<td>20090713011885</td>
<td>4.4476895e+001</td>
</tr>
<tr>
<td>20090713011890</td>
<td>4.4562298e+001</td>
</tr>
<tr>
<td>20090713011895</td>
<td>4.4657984e+001</td>
</tr>
<tr>
<td>20090713011900</td>
<td>4.4733953e+001</td>
</tr>
</tbody>
</table>
Chapter 8

Conclusion and Future Work

This thesis presents the optimal tracking problem based on ILC theory. The concept of learning through the experience of ILC to track a desired trajectory has been extensively analyzed in the area of control. However, when there is a mass data point, these ILC approaches have trouble in generating an optimal trajectory, performance, and rate of convergence. Moreover, most ILC algorithms formulate system models in a lift-system representation; thus, the computational cost and time increases whenever the length of the operation time increases. Our approach overcomes these drawbacks by utilizing only the essential information of data points.

The thesis makes three key contributions to this research field. First, we bridge the gap between trajectory planning and ILC tracking control where the reference trajectory could be updated iteratively for a better convergence. Second, we propose an approach to utilize only the essential information of data points without building the desired trajectory in both unconstrained and constrained ILC. Specifically, we have shown that the learning algorithm that investigates critical points can successfully obtain the convergence of error and control energy. By manipulating these parameters, a very good performance is achieved. The results improve upon those obtained by a traditional ILC, being significantly more direct and simple. The constraints and repeating disturbance was also verified in the new algorithm. Lastly, we have demonstrated an norm optimal ILC approach to deal with non-repetitive uncertainties.
at the reference data points. The approach also leads to a new problem, which is multiple interval points tracking.

Future work will extend the theory to more generic scenarios where we consider trajectory planning and tracking control problem for nonlinear systems. The reason is that many tracking applications, especially robotics systems are nonlinear. Moreover, as we have shown that it is possible to incorporate numerical method in the ILC algorithm in order to against both repetitive and non-repetitive disturbances; thus, this approach could be further consider in the coming works to deal with more general learning paradigms such as non-repetitive uncertainties in both system dynamics and reference trajectories. Finally, our objective is to apply the theory works to a practical application in order to verify the effective of our proposed techniques.
References


Acknowledgements

This thesis has only been possible due to the support of numerous people. I would therefore like to express my gratitude to all those that have played a role. More specifically I would like to thank Professor Hyo-Sung Ahn, who gave me the opportunity to join the Distributed Control and Autonomous Systems Laboratory. I especially want to thank him, in his role as my Master advisor, for his support, constant availability, reassurance, and encouragement during my Master program. The numerous discussions we had, and the advices he provided, were invaluable to the success of this work.

I am honored to have had Prof. Jong-Ho Lee, Prof. Hyo-Sung Ahn, and Prof. Hyuk-Sang Kwon as my thesis committees. Thank you for accepting to evaluate this thesis.

Thank you to all my colleagues at the Distributed Control and Autonomous Systems Laboratory for such an enjoyable working atmosphere. They have made my Korean life so memorable and enjoyable: HurHwan, KwangKyo, HanEol, JiHwan, YoungHun, ByeongYeon, YoungCheol, SangChul, SeungJu, Sanghyuk, MyoungChul, Jeyoung, ByungHun, Taekyung, Stefan, Catalin, Ankh Byayar.

I thank profoundly my parents and sister for their unaltering love and support in all that I have done throughout my life. I am grateful to all my friends who have always encouraged and inspired me during my research work. Last but not least, I would like to deeply thank ChiChi for her love and companionship, and the support she has given me despite the physical distance between us.

I gratefully acknowledge the financial support received from the Gwangju Institute of Science and Technology.