UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION
INSTITUTE OF SCIENCE AND TECHNOLOGY

ADAPTIVE ITERATIVE LEARNING CONTROL FOR A LINEAR SYSTEM WITH UNKNOWN PARAMETERS

MASTER THESIS

IBRAHIM F. TAHA ALZAIDI

MECHANICAL AND AERONAUTICAL ENGINEERING DEPARTMENT

MASTER THESIS PROGRAM

DECEMBER 2017

UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION
INSTITUTE OF SCIENCE AND TECHNOLOGY

ADAPTIVE ITERATIVE LEARNING CONTROL FOR A LINEAR SYSTEM WITH UNKNOWN PARAMETERS

MASTER THESIS

IBRAHIM F. TAHA ALZAIDI

1406080010

MECHANICAL AND AERONAUTICAL ENGINEERING
DEPARTMENT
MASTER THESIS PROGRAM

SUPERVISOR
ASSIST. PROF. DR. HABIB GHANBARPOURASL

Ibrahim F. Taha ALZAIDI, having student number **1406080010** and enrolled in the Master program at the institute of Science and Technology at the University of Turkish Aeronautical Association, after meeting all of the required conditions contained in the related regulations, has successfully accomplished, in front of jury, the presentation of the thesis prepared with the title of: "ADAPTIVE ITERATIVE LEARNING CONTROL FOR A LINEAR SYSTEM WITH UNKNOWN PARAMETERS".

Supervisor:	**Assist. Prof. Dr. Habib GHANBARPOURASL** **University of Turkish Aeronautical Association**	[illegible]
Jury Members:	**Assist. Prof. Dr. Habib GHANBARPOURASL** **University of Turkish Aeronautical Association**	[illegible]
	Assist. Prof. Dr. Durmuş Sinan KÖRPE **University of Turkish Aeronautical Association**	[illegible]
	Assist. Prof. Dr. Kerim Youde HAN **Çankaya University**	[illegible]

Thesis Defense Date: 01.12.2017

UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION
INSTITUTE OF SCIENCE AND TECHNOLOGY

I hereby declare that all the information in this study I presented as my Master's Thesis, called "Adaptive Iterative Learning Control for a Linear System with Unknown Parameters" has been presented in accordance with the academic rules and ethical conduct. I also declare and certify on my honor that I have fully cited and referenced all the sources I made use of in this present study.

[illegible]

Ibrahim, Alzaidi

01.12.2017

ACKNOWLEDGMENTS

This is a nice opportunity to show deep thankfulness to all those who helped and supported me in completing this piece of work.

First of all, I would like to show sincere thankful to my supervisor Assist. Prof. Dr. Habib GHANBARPOURASL, who willingly shared his knowledge, experience, and ideas with me, where without his excellent courses, guidance, and remarkable patience I don't think I would ever have finished this work.

I would like to thank all of the academic staff in the Mechanical and Aeronautical Engineering Department for allocating their valuable time and effort during these years of study specially Assist. Prof. Dr. Kerim Youde HAN for his valuable course and his inspiration to study ILC.

I want to express my sincere gratitude to my mother, wife, brother, sisters, and all of my family for their complete reliance, unconditional love, and moral support throughout my academic years.

December 2017 Ibrahim, Alzaidi

LIST OF CONTENTS

LIST OF FIGURES

ABSTRACT

ADAPTIVE ITERATIVE LEARNING CONTROL FOR A LINEAR SYSTEM WITH UNKNOWN PARAMETERS

Alzaidi, Ibrahim
M.Sc., Mechanical Engineering Department
Supervisor: Assist. Prof. Dr. Habib GHANBARPOURASL
December 2017, 92 pages

One of the main disadvantages of iterative learning control (ILC) systems is that the learning gain is static and does not update itself by using the current data of iteration. For a dynamic system has uncertain parameters, although the controller will update its input, but the learning gain will not be adapted according to the new changes in parameters, so that, it will still treat the dynamic system as if no change has occurred, which leads to the controller failure in the tracking process.

In this thesis, the analysis and design of adaptive iterative learning control (AILC) algorithm is implemented by using least squares approximation. A new method for calculating ILC learning gain matrix is presented. The ILC algorithm is applied on a SISO linear time-invariant (LTI) dynamic system with unknown parameters, and a parameter identificator is designed to optimize the accurate values of that unknown parameters and minimize the tracking error. The simulation procedure with the response results is placed in the last part of this monograph. The simulation, which represents the proposed algorithm, is suitable for linear systems that have unknown parameters but the bounds of these parameters are limited. Furthermore, the controller is able to compensate the internal error value.

Keywords: Adaptive Iterative Learning Control, Parameter Identification, Unknown Parameters, Repetitive System, Learning Gain, Single Input Single Output (SISO).

ÖZET

BİLİNMEYEN PARAMETRELİ LİNEER BİR SİSTEM İÇİN ADAPTİF İTERATİF ÖĞRENME KONTROLÜ

Alzaidi, Ibrahim
Yüksek Lisans, Makine Mühendisliği Bölümü
Danışman: Yrd. Doç. Dr. Habib GHANBARPOURASL
Aralık 2017, 92 sayfa

Tekrarlayan Öğrenme Kontrolü (ILC) sistemlerinin en büyük dezavantajlarından biri, öğrenme kazancının statik olması ve iterasyonun mevcut verilerini kullanarak kendini güncellemez olmasıdır. Belirsiz parametreleri olan dinamik bir sistemde kontrol elemanının kendi girdisini güncellemesine rağmen öğrenme kazancının parametrelerdeki yeni değişikliklere uyarlanamaması anlamına gelir ki bunun sonucunda öğrenme kazancı dinamik sistemi değişime uğramamış gibi işleme tabi tutacak ve bu da izleme sürecinde denetleyici arızasına neden olacaktır.

Bu tezde, uyarlamalı iteratif öğrenme kontrol (AILC) algoritmasının analizi ve tasarımı, en küçük kareler yaklaşımı kullanılarak gerçekleştirilmiştir. ILC öğrenme kazanç matrisini hesaplamak için yeni bir yöntem sunulmuştur. ILC algoritması bilinmeyen parametreleri olan bir SISO doğrusal zamanla-değişmeyen (LTI) dinamik sistem üzerine uygulanır, ve bir parametre tanımlayıcı bilinmeyen parametrelerin doğru değerlerini optimize etmek ve izleme hatasını en aza indirmek için tasarlanmıştır. Yanıt sonuçları ile simülasyon prosedürü bu monografinin son bölümüne yerleştirilmiştir. Önerilen algoritmayı simgeleyen simülasyon, bilinmeyen parametreleri olan doğrusal sistemler için uygundur ancak bu parametrelerin sınırları sınırlıdır. Ayrıca, kontrolör dahili hata değerini telafi edebilmektedir.

Anahtar Kelimeler: Uyarlamalı Yinelemeli Öğrenme Kontrolü, Parametrelerin Belirlenmesi, Bilinmeyen parametreler, Tekrarlayan Sistem, ILC Sisteminin Kazancı, Tek Giriş Tekli Çıkış (SISO).

CHAPTER ONE

INTRODUCTION

A general introduction to automatic control branches will be introduced in this chapter, the position of *Iterative Learning Control* (ILC) is discussed in this thesis with presenting some usage examples, a literature review of adaptive iterative learning control (AILC) and the problem to be solved in this thesis is presented, and then the organization of the thesis is placed at the end of this chapter.

1.1 Introduction

Any field of engineering and science has an automatic controller as an essential part of its systems and components. It is a necessary part of many fields like robotic systems, space-vehicle systems, modern manufacturing systems, and any operation involving control of humidity, pressure, flow, temperature, etc. [1]. Using feedback signals represents the main feature in control systems engineering to rise up the performance of the system to be controlled [2]. The common control theories which are used in current time are the conventional (classical) control theory and the modern control theory [1]. Under the second theory, there are many branches of theories like optimal control, nonlinear control, adaptive control, estimation control, intelligent control and robust control see Figure 1.1 [2]. The main differences between the conventional control theory and the modern control theory are related to some conditions that the control system can deal with such as the domain approach, the number of the inputs and outputs, and linear or non-linear systems. While the conventional control theory controls only Frequency domain approach and single-input single-output (SISO) linear time-invariant systems, the modern control theory is applicable for multi-input multi-output (MIMO) linear or non-linear systems and time-varying or invariant systems.

The connection between the performance of the industrial sector and the success in the market is highly bonded. The development of advanced industrial control systems is an important factor in providing the technological base that is necessary to achieve improving quality and competitiveness in the industrial sector.

The advanced control systems can deal with a lot of problems that may happen in the dynamic systems, one of these problems is trajectory tracking, in this case, the controller must follow the path as fit as possible. This problem appeared clearly in industrial lines or robotic field which need to follow critical lines accurately, even though, it is important to other applications which also related to motion control [3].

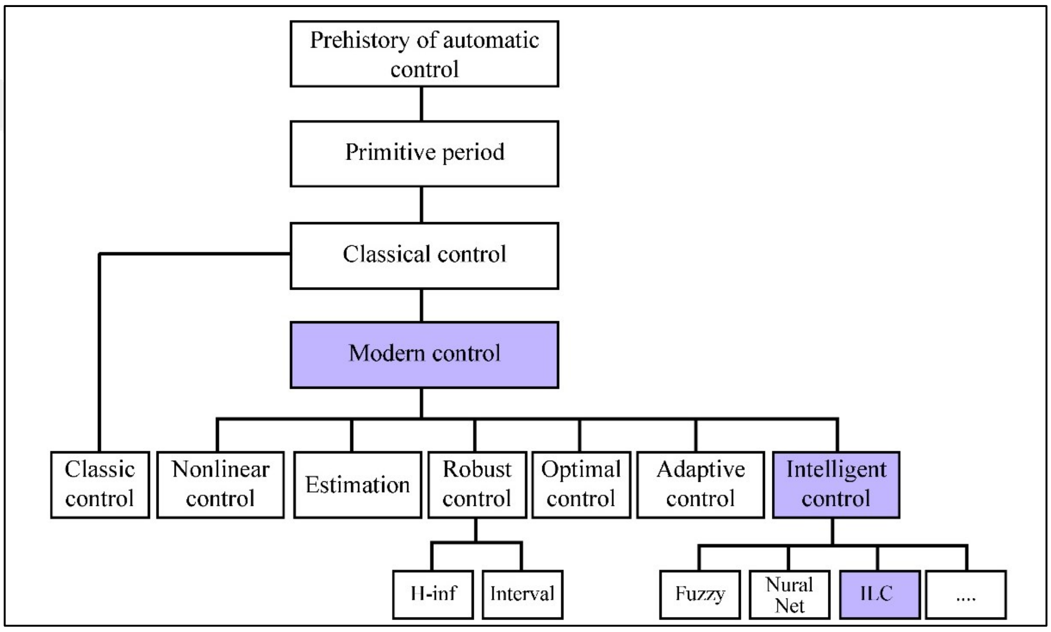


Figure 1.1: Control Engineering - History and ILC [4].

In spite of using several numerous design tools in the conventional control theories in order to improve the response of the dynamic systems in trajectory tracking field, but reaching the desired results not always available. If a trajectory track has been given to a machine controlled by a conventional control algorithm and this machine made some errors while achieving the desired path due to some outer or inner disturbances, these errors will be repeated in each iteration until the end and there is a possibility to be worse due to some unknown parameters or cumulative errors in the system unless get manual adjustment, in this case, achieving perfect tracking by using the traditional methods is not easy, so that, using modern control theories in the trajectory tracking control is preferable.

“Repetition is the mother of all learning”, with this quotation, it can be deduced the way in which a person acquired his knowledge and experience. This maxim is applicable to many things in life, for instance, the chef while cooking food he tests the food repetitively to check the seasoning and the other gradients till reach to the best result, and as he repeats this procedure more times, as he reaches to the best result in shorter time. This example can be called simply as the human version for cooking of what is called iteration control system algorithm (ILC). Here the taste variation of food from the desired taste represents the error. Chef’s experience represents the feedback controller which will still check the taste till reach to the wanted result. The number of making the same dish by the chef denote the iteration numbers in ILC, this is how the chef learned to optimize the seasoning after some iterations. This concept of ILC is transformed later into mathematic models [3].

The iterative learning control (ILC) is one of the hopeful algorithms for the control systems that working according to self-learning concept. It is an algorithm capable to track the required trajectory within a certain period of time [5]. The intelligent control has many branches, one of them is the iterative learning control (ILC) as seen in Figure 1.1, which can be defined as an *effective control tool for improving the response of the repetitive motion performance of uncertain dynamic systems* [2].

The idea of the iterative learning control systems is to apply a simple algorithm repetitively to an unknown plant until reach perfect tracking [6]. As seen in Figure 1.2, the error is reduced from trail to trail by increasing the input signal until approach zero error in the output and the desired graph. It is similar to the human example expressed before, as much repeat the activity as much get more experience and rise up his physical performance. Impression and similarity are the qualities that enable human to get his knowledge. In machines, the matter is not far away from this idea, where the initial setup, uniform sampling, fixed time point, repetitive desired trajectory and another setup could be taken similar to what mentioned in case of humane qualities [2].

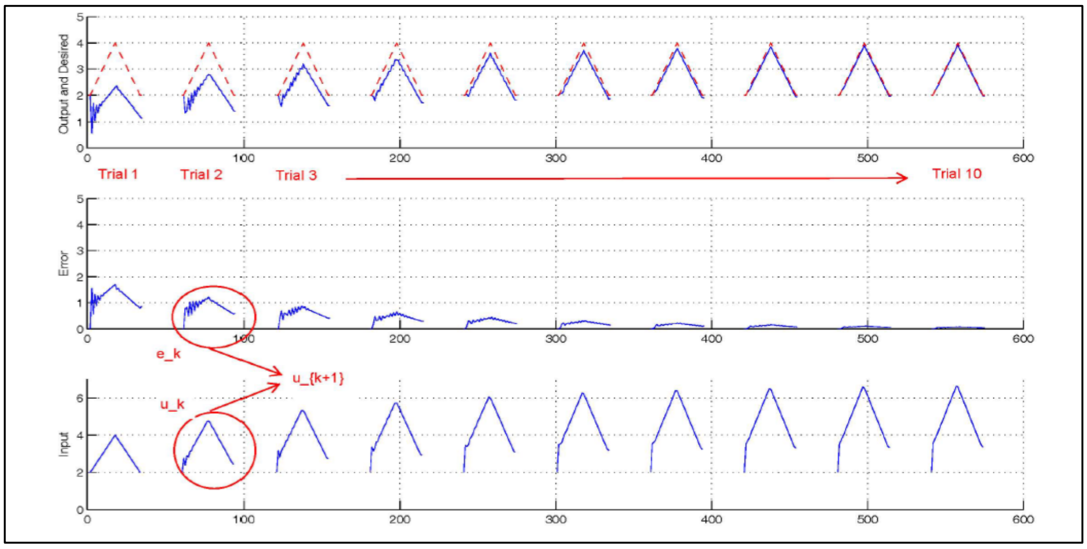


Figure 1.2: The Convergence to the Zero Error Through the Iterations [4].

Even the iterative learning control idea started before the 1970 but the first publication was at 1974, and the first paper was in Japanese language (Formation of High-Speed Motion Pattern of Mechanical Arm by Trial) by Masaru Uchiyama in 1978 [7]. In the eighties, Arimoto et al. rigorously formulated the Iterative Learning control problem [8]. Since 1992, the study and researchers in ILC have progressed very fast. On one hand, important work has been showed and stated in the main area of developing and analyzing new algorithms of ILC. On the other hand, researchers also have recognized that making integration between ILC and other control theories might give better controllers that show the desired performance which is impossible to be shown by any individual approach [9].

Since the birth of ILC idea in early 1980's, the history of ILC can be separated into two periods. The first period was between the early 1980s' and early 1990's which represents the linearly increasing period of ILC, whether in terms of publications and reports in theory or in applications. The second period was from early 1990's so far, nevertheless, the activities of research in ILC undergo a nonlinear (exponential) increase [10].

1.2 Usage of ILC

This control system is commonly used in the repetitive operated dynamic systems like the robotic manipulators in the production lines, also it is used in several numerous applications in different fields like automation, machine tool control, military, transportation systems, economic systems, medical engineering …etc. [2].

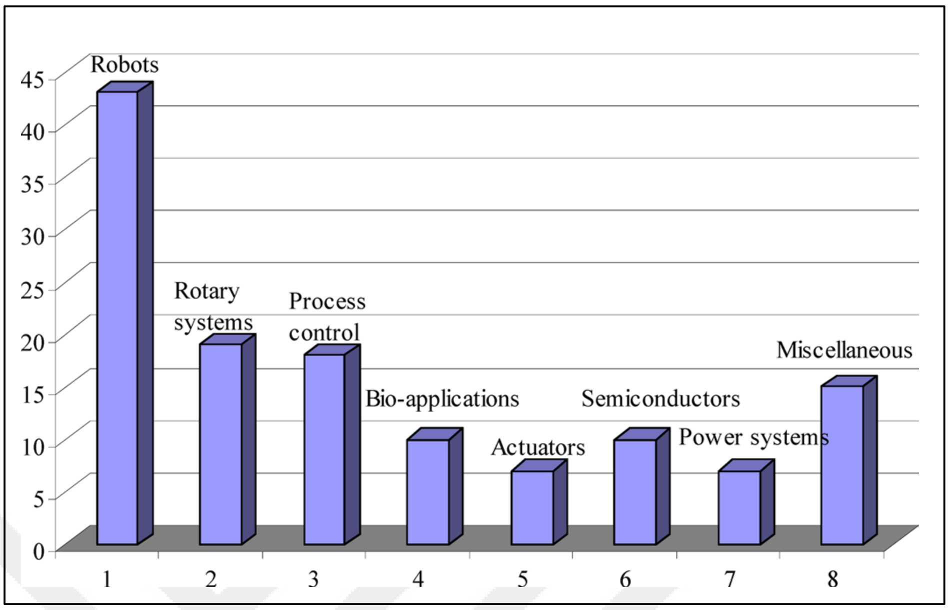


Figure 1.3: ILC Applications [2].

Although the robotic arms and manipulators are the most applications controlled by ILC, there are other applications controlled by ILC too such as:

1.2.1 Automated ploughing

The concept behind the automated agricultural ploughing control is creating parallel lines of grooves depending on a sequence, this sequence takes the initial groove line as a reference signal for the future iterations k. Using iterative learning control system in this system is valuable to be sure that the repetition to the groove line sequence is acceptable and by repeating the iterations the error will be less and less till reach to error very near to zero [11].

1.2.2 Automated coal cutting

The equipment used for cutting the underground walls in coal extraction process shows another example of the machines that use ILC controller in their work. This equipment works in a multipass process while cutting the walls through trajectories. The tool used for cutting the underground walls basically is a rotating cutter, this cutter moves along the face of the wall through trajectories defined by the same face of that wall, when the one pass along the wall's face has been finished, the equipment will repeat the movement from new set point which revealed by cutting to start the second iteration depending on the previous iteration data as a new initial data. As a result, the controller has the ability to track the coal layer as the progress of the trail, and also it can set the new starting positions at each iteration [11].

1.3 Literature Review

Comparatively, slow convergence rate is the problem of the classical ILC algorithms, where they are not able to adjust any changes could happen in parameters of dynamic systems due to the fixed laws on which the algorithm depends on [5]. That property represents one of the main weak points of standard ILC algorithms because they use fixed learning laws which are limited to a prior system knowledge. This means that even though more system knowledge may be gained each iteration, the learning law remains static. One way to address this limitation is to adaptively adjust the learning law after each iteration [12], so that, iterative learning control systems merged to adaptation algorithms are growing into a promising research area [13].

ILC was first popularized by Arimoto et al. (1984) and several recent surveys of learning control suggest its recent surge in interest (see [2], [14]). The major improvement of ILC is that act of unknown systems can be enhanced with little dependency on the system model [8].

Messner et al. (1990), used a new adaptive learning law based on integral transforms, though in a repetitive control environment [15].

French and Rogers (1998) consider a system which included adaptively estimated parameters which can decrease parameters in a finite time horizon [16].

Bien, Z. et al. (1998) tried to guide readers to a good work related to the adaptive iterative learning control in discrete time. Also, there is plenty of space for improving ILC algorithms, without the need to former information of the system [17].

Owens et al. (1998), used the current trail feedback to implement convergence/stability norms for common adaptive learning control system and by using high gain for linear MIMO state space system [18]. French et al. (1999) also provided a scheme of learning control which based on an adaptation of the learning gain by using only the data of sign (BC) [19].

Phan and Frueh (1999) present a new algorithm of ILC, in this algorithm they used a reference model as a leader to the learning process, where this model enables the controller to achieve the desirable properties by choosing good previous knowledge for that [20].

Choi and Lee (2000) estimated uncertain parameters for AILC scheme by using a different advantage in the time domain when repetitive troubles are recognized and reduce the error percent in the next iteration. There is no need to know the limits of parameters that the algorithm should work with because the learning gains will be updated self-reliantly and the parameters will be adapted [21].

Early approaches of ILC focused on proportional (P-) type and proportional plus derivative (PD-) type learning algorithms similar to the feedback controller equivalents. Lately, DeRoover et al. (2000) and Gunnarsson et al. (2001) gained model-based learning algorithms popularity. These are based on the point that the ideal learning law is based on the inverse system dynamics. These model-based learning laws are limited by how well the system is modeled which begs the following question: is it necessary to spend effort obtaining a better system model before running the iterative control system? [12] [22] [23].

Lee et al. (2002) developed a direct model reference adaptive control scheme which will be converted to produce two laws, the first one is direct model reference repetitive control law, and the second one is discrete-time direct model reference learning control law. These two laws can definitely gain the zero error of converging by adjusting the input command. They produced this algorithm by using linear time-invariant systems with no disturbances and applied the original adaptive control ideas and also by developing the learning laws with repetitive disturbances [24].

Chen and Jiang (2002) used (SISO) nonlinear systems and apply an adaptive iterative control algorithm on it after assigning fully unknown feedback high-frequency gain, where the convergence will be obtained by manipulating that gain in a Nussbaum-type function [25].

Yang et al (2002) presented an adaptive robust iterative learning control scheme with structured and unstructured uncertainty by using the Lyapunov method, they used an inexact model of a robotic system, which decomposed into both situations, whether were repetitive and non-repetitive parts [26].

Owens and Feng (2003) presented a new method of ILC law, in this law they used quadratic performance index to estimate the new parameters, this algorithm is designed to work with discrete-time LTI dynamic system to guarantee the convergence to zero error [27].

Miyasato et al. (2003) suggested a hybrid AILC and applied it to robotic manipulators to show how can adapt their actions to the changes and improve their skills [28]. Shou et al. (2003) used a type of nonlinear time-varying systems under some assumptions and conditions and applied open-closed-loop D-type ILC algorithm to ensure the convergence of the given learning gain [29].

Chien et al. (2004) developed an offline nonlinear ILC systems, this system is based on output-feedback linearization [30]. Chiang et al. (2004) developed an output tracking error model, which used filtered signals from plant output and input, then generated a new output-based AILC for a linear system which has repetitive motion with unknown parameters, high relative degree, initial resetting error input disturbance and output noise [31].

Ashraf et al. (2008) proposed a new merged algorithm between ILC and identification techniques and applied this algorithm on a practical simulation to test its robustness. The optimal gain matrices values are calculated by using steepest descent approach [5].

Stearns et al. (2009) present an iteration varying learning filter which is based on identification techniques. This learning filter should be calculated every iteration to make the closed-loop system inverse based on a least squares approximation. They showed that the iteration which changes the learning filter successfully decreases the 2-norm error speedily like a static, model-based learning filter and better than the learning filter of P-Type [12]. Dong et al. (2009) developed an AILC based on a theory of two operational modes, single and repetitive modes, and guarantee the convergence in both of them [32].

Bu et al. (2013) proposed a model-free adaptive iterative learning control (MFAILC) algorithm in discrete time form to track a path of farm vehicle. This algorithm based on the dynamical form linearization model for a farm vehicle kinematic model, which designed to depend on the input/output data of the farm vehicle [33].

Oh et al. (2015) presented an AILC applicable on discrete linear time-invariant LTI, also it can be applicable to a stochastic system with batch-varying reference trajectories BVRT. And they proposed two approaches based on batch-domain and time-domain Kalmen filter to deal with the problem of state and measurements noises [34].

Li et al. (2015) examined a distributed coordination problem by using iterative learning control and applied it to leader-follower multi-agent systems with the second-order nonlinear dynamic system [35]. Xu (2016) presented a new distributed AILC algorithm applicable for a type of high-order nonlinear multi-agent systems (MAS) [36].

Yu et al. (2016) proposed a new AILC algorithm applicable to nonlinear systems with both state and input constraints and took into consideration the external disturbances with random initial errors and time-varying parametric uncertainties [37].

Wei (2017) presented an AILC algorithm which can be applied to the nonlinear dynamic system, this system has unknown input dead-zone and unknown time-varying delays by using many theories like Lyapunov-Krasovskii, Young's inequality and radial basis function neural networks [38].

1.4 Thesis Statements

Iterative learning control algorithms are very effective method for control repetitive motioned dynamic systems, but their static learning gains made them inefficient to deal with the systems which have uncertain parameters, in this case, ILC system will fail to adapt to the changes in these parameters, then the dynamic system will fail to respond the reference signal as well.

1.5 Thesis Objective

The objective of this thesis is to design a control algorithm for dynamic systems that move repetitively and have some unknown parameters need to be adapted optimally. An iterative learning algorithm with Least squares approximation will be used to design an Adaptive Iterative Learning Control (AILC) algorithm to control a linear single-input single-output system with unknown parameters.

1.6 Thesis Organization

In the first chapter as seen above, a general background about the automatic control systems and the ILC control systems is illustrated, some examples of ILC applications and the literature review also have placed in this chapter. In the second chapter, a general description of ILC theories and the relationships between these theories and the conventional control theories has presented. In chapter three, least squares approximation method and its branches have displayed. Chapter four discusses the adaptive iterative learning control (AILC) algorithm which depended on the ILC algorithm and the identification algorithm. In chapter five, the simulation and its results have presented. Finally, chapter six which contains the conclusion and recommendations.

CHAPTER TWO

ITERATIVE LEARNING CONTROL

The ideas on iterative learning control (ILC) presented in this chapter were developed over a period of several years and have performed in many publications, with changing degrees of fullness.

2.1 General Overview of ILC

The basic overall idea of iterative learning control can sum up in Figure 2.1. The signals described in this figure belong to the finite interval $t \in [0, t_f]$, where k is the trail number, $u_k$is the input signal, y_k is the output of the system, $u_{k+1}$is the updated input signal, and y_d the desired output reference. The work of this system can be summarized as follows: when the input $u_k(t)$ is applied to the cycle at the trail k, the output of the system will be $y_k(t)$, a copy of these two values will be sent to memories in order to use them in the next trail for learning, and then the new updated input $u_{k+1}(t)$ will be calculated after finding the error value between the output and the desired signals $\left(e_k(t) = y_d(t) - y_k(t)\right)$, the new generated input $u_{k+1}(t)$, also it should be saved in a memory to be used as a new input to the next trail. So that, the main point of ILC input design is to obtain smaller error rate at each trail in order to approach zero.

The approach of ILC can be explained by giving some notations, Let $f: U \longrightarrow Y$, where f is a nonlinear operator working on mapping elements in the vector space U to the vector space Y, so $y = f(u)$ where $u \in U$ and $y \in Y$. So, the assumption is suitable norms that can be defined on u and Y as well as a norm on $f(\cdot)$. Let $y(t) = f_S(u(t), t)$ for a system called S, and let the dynamic system be $f_S(\cdot, t)$ where, f_S is the I/O operator.

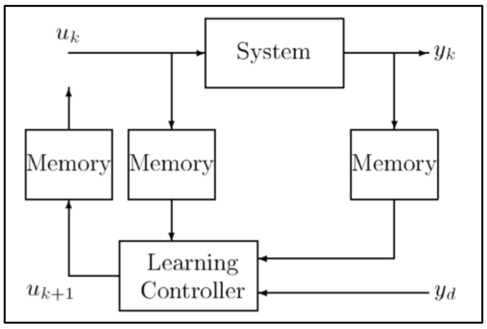


Figure 2.1: Iterative Learning Control Idea [39].

The aim of this system is obtaining the closest output value y_k to the desired value y_d, when that matching is reached, the input signal at that point called the optimal input $u^*(t)$, the representation of these assumptions can satisfies:

$$\min_{u(t)} \parallel y_d(t) - fs(u(t),t) \parallel = \parallel y_d(t) - fs(u^*(t),t) \parallel \qquad (2\text{-}1)$$

ILC algorithm works to calculate the optimal input $u^*(t)$ when all of the signals belong to the finite interval $[0, t_f]$. ILC can reach to this optimal input by generating a sequence of inputs $u_k(t)$ then convert this sequence to that optimal input, the converting idea can be represented as:

$$\begin{aligned} u_{k+1}(t) &= f_L(u_k(t'), y_k(t'), y_d(t'), t) \\ &= f_L(u_k(t'), f_S(u_k(t')), y_d(t'), t), \qquad t' \in [0, t_f], \end{aligned} \qquad (2\text{-}2)$$

such that

$$\lim_{k\to\infty} u_k(t) = u^*(t) \qquad for\ all\ t \in [0, t_f] \qquad (2\text{-}3)$$

2.1.1 Some explanations about ILC

1. If ILC algorithm works successfully, the next error signal must be reduced, and that reduction will affect the input signal for the next iteration as well which will give closer output to the desired signal.
2. ILC signals have been defined by two variables k and t, where the integer k denote the trial index while the variable t represents the discrete or the continuous time.
3. When constructing the input $u_{k+1}(t_o)$, there is a possibility to use information about what happened after applying the input $u_k = (t_o)$, because after the trail is completed there is no reasonable constraint in the operator f_L of ILC algorithm prevents it. This fact can be expressed as a new variable called t' in

the algorithm of ILC. There is just one possibility to prevent that when $t = t_f$, though it is $t' \in [t, t_f]$ it can be assumed as $t' \in [0, t_f]$.

4. The difference between iterative learning control and conventional control have been illustrated. Figure 2.2 (a) shows the approach of ILC while preserving data about inputs effect at every instant throughout the iteration and calculate the corrected signal by depending on that data during the next trail. Figure 2.2 (b) shows that the conventional control calculates the error by using the same data in that current step.

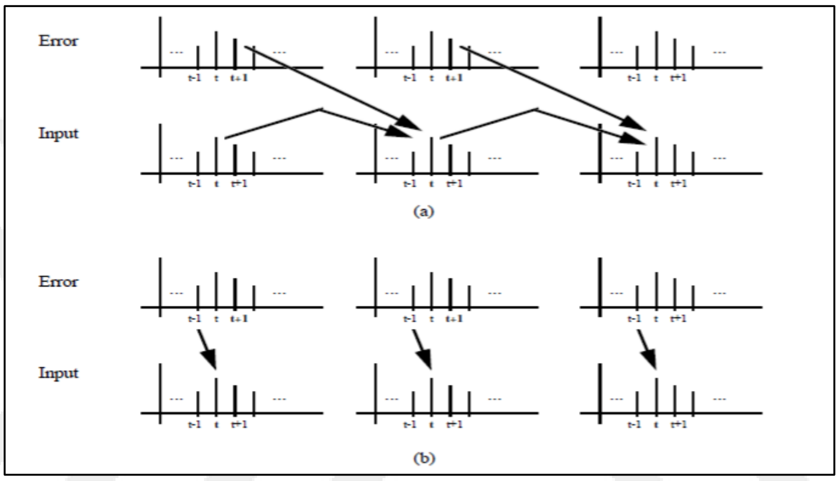


Figure 2.2: (a) (ILC), (b) Conventional Control [39].

5. Assume that the initial conditions of the system will be reset when the iteration is started again to its initial values.
6. Trail length t_f should be assumed as fixed. However, and it could be taken as $t_f \rightarrow \infty$ for analysis purposes.
7. The properties of ILC's convergence must not be affected by the desired signal $y_d(t)$. If the desired signal is introduced, the ILC controller will not change any of its algorithm and it will learn the new optimal input simply.
8. The system S must be stable to show the convergence, if it is not, so one of the conventional techniques should be used to obtain stability then ILC algorithm be applicable. Because ILC's work is improving performance [39].

2.2 Connection to Other Control Paradigms

Before going ahead in discussing ILC algorithms, it is beneficial to talk about the relationship between ILC and the other conventional algorithms.

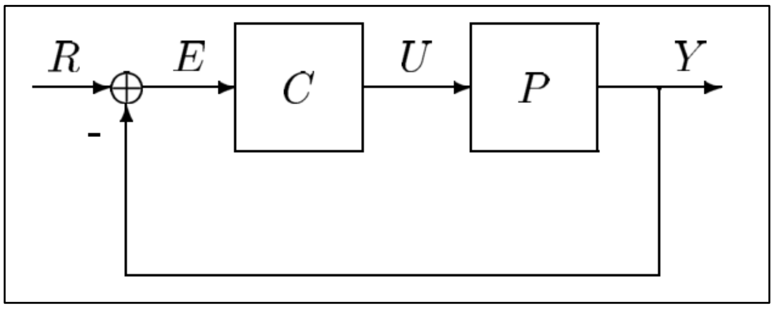


Figure 2.3: Unity Feedback Control System [39].

In unity feedback control system shown in Figure 2.3, the block P denote system plant which will be controlled by the controller shown in block C, and the block Y value will be calculated by the control system. The control system is operated by calculating the value of the output signal Y, then comparing it with the desired signal (reference signal) R. The difference between them $(R - Y)$ represents the error E which enters to the controller in order to calculate the new suitable actuating signal U, which will be used to drive the plant. Many control problems are subjected to Figure 2.3 as follow:

2.2.1 Feedback control systems

General feedback control algorithms are similar to each other whether frequency domain techniques or pole placement technique, they have block P as plant, block C as a controller, the reference signal R as input signal to the controller, and Y is the output of the system, where the feedback of close-loop will arrange frequency characteristics, the poles in case of pole placements, or even the steady-state error properties.

It becomes clear that ILC is not like the technique explained above because ILC technique cannot affect the poles of the system, not like pole placement or frequency domain technique.

2.2.2 Optimal control systems

Minimizing the error in optimal control systems can be explained as $\min_{C} \|E\|$, the optimal controller will not be optimal anymore if the plant P has changed in comparison to the model of the dynamic system. In case of ILC, it has similarity somehow to the optimal controller, where if ILC designed in order to get stability of a dynamic system that could be possible because ILC can produce an output similar to the output of the optimal controller and both of them working to minimize the error. The difference between them is that ILC can obtain that output by adding the optimal input U^* to the system instead of the real-time error processing, see Figure 2.4.

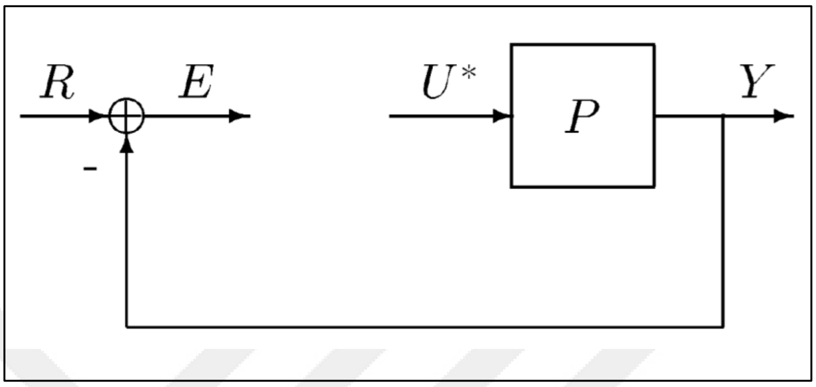


Figure 2.4: ILC Approach In Another Representation [39].

2.2.3 Adaptive control systems

When someone starts studying ILC, he might think that there is no difference between it and the adaptive control. The difference between them can be described as that between Figure 2.3 and Figure 2.4. ILC is not like the known conventional adaptive control, where, in the second type the adjustment of the parameters is done by online algorithms till reach the steady state equilibrium. Definitely, if any change happens in the parameters of the dynamic system the learning controller will be adapted to the new situation by changing the values of the parameters optimally and will change the input for the next trail [39].

2.2.4 Robust control systems

As ILC algorithm can deal with unknown plants, so it is similar to the robust control system by this feature because the second one can treat with uncertain plants [39].

2.2.5 Intelligent control systems

Fuzzy logic, neural networks, and expert systems, all of them have developed control paradigms which can be met under the name of intelligent control techniques.

All of them have a common thing which can be described as involving learning somehow.

Although the word "learning" has been widely used in the field of control, some confusion has occurred in understanding this word. In general, the control system has the ability to adapt the input signal in case of any change happened in the input-output observation, this feature refers to the word learning. But this feature is not only used in one control method, many control systems have the ability to respond to the changes happened in their environments. Thus the word “learning” needs to be more specified. So that, for (ILC) the word learning be used to point to generating a sequence of input trajectory iteratively, the detailing is shown in Figure 2.1 [39].

2.3 ILS’s algorithms

In this part, the development of ILC algorithms will be presented.

2.3.1 Continuous-Time ILC

It is too hard to talk about all of ILC kinds because its research field is too wide, due to that the iterative learning control has two kinds of developing and analyzing algorithms, developing main algorithms and making integration between ILC and other control theories like nonlinear control, estimation control, robust control, optimal control and adaptive control, each of these algorithms has combined with the ILC giving out new algorithms for repetitive systems.

The controller task makes y_k approach y_d , as the increasing of the iteration k as the approach happen within a fixed interval $t \in [0, T]$. Consider the linear continuous-time system below:

$$\dot{x}_k(t) = Ax_k(t) + Bu_k(t) \qquad (2\text{-}4)$$

$$y_k(t) = Cx_k(t) \qquad (2\text{-}5)$$

where $x \in R^n, u \; and \; y \in R^r, A \in R^{n\times n}, B \in R^{n\times r}, and \; C \in R^{r\times n}$

For the system above and under some assumption stated by (S. Arimoto), he made the standard algorithm and its error equation:

$$u_{k+1} = u_k + \Gamma e_k \qquad (2\text{-}6)$$

$$\mathrm{e}_k(t) = y_d(t) - y_k(t) \qquad (2\text{-}7)$$

Where Γ is a diagonal learning gain matrix, and the convergence will be achieved when $\|I - CB\Gamma\|_i < 1$ [8].

Starting from the classical ILC algorithm of Arimoto, many general expressions can be developed. For example, the combination of ILC and PID algorithms gives the expression shown in equation (2-8):

$$u_{k+1} = u_k + \phi e_k + \Gamma \dot{e}_k + \psi \int e_k \, dt \tag{2-8}$$

Where Φ, Γ, and Ψ denotes to the learning gain matrices [40]. Among possible usable update forms, the form (2-9) could be the simplest one:

$$u_{k+1}(t) = u_k(t) + \phi \Delta y_k(t) \tag{2-9}$$

Where, Φ is a positive constant matrix. The next form is considered as the second simplest form:

$$u_{k+1}(t) = u_k(t) + \Gamma \frac{d}{dt} \Delta y_k(t) \tag{2-10}$$

As the right-hand side of the form (2-10) has differentiation term with time, so this law will be considered as non-causal one, in spite of that, it is not used in real-time when the trail is k^{th}, but it is used at the trail $(k+1)^{th}$. In order to find an appropriate value of $\Delta y_k(t)$, sufficient method can be applied for that [9].

There are some basic assumptions for the standard ILC contain:

1. Some types of Lipschitz condition or stable dynamics.
2. Each trial has the same length.
3. At the start of each trial, the system goes back to the same initial conditions.
4. Undefined amount of time can elapse between trials.

* The last two features distinguish ILC from repetitive control (RC) [41].

This theory developed by D. H. Owens who studied this algorithm by updating the following expression:

$$u_{k+1}(t) = \alpha u_k(t) + K e_{k+1}(t) \tag{2-11}$$

and its steady-state error is given by $e_\infty = (I + GK_{eff})^{-1} y_d$, where the matrix G represents a dynamic system, the desired trajectory is considered as y_d, and $K_{eff} = K/(1 - \alpha)$ [42]. Goldsmith has presented that if the update algorithm ($u_k = F\, u_{k-1} + Ce_k + Ce_{k-1}$) applied on a learning control system, its output equals to a feedback control system updated by the equation $u(t) = Ke(t - 1)$ if the term K is calculated by $K = (I - F)^{-1}(C + D)$ [43].

A lot of work has been done for nonlinear ILC, leading to what is called (Energy Function Approach) by using what is called control Lyapunov function. Also, a lot of work has been done using Fuzzy-Logic and Artificial neural networks [41].

2.3.2 Optimization-Based Approaches

Some researchers [44], in order to choose T_e and guide the convergence to follow some error's gradient, they took an approach similar to the discrete-time learning control expression explained below:

$$u_{k+1}(t) = u_k(t) + Ge_k(t+1) \tag{2-12}$$

To reduce the quadratic cost of the error and to calculate the gain G, optimization gradient method have been used:

$$J = \frac{1}{2} e_k^T(i+1)Qe_k(i+1) \tag{2-13}$$

To take place between successive trails. Several techniques have been illustrated by the authors for implementing G, and the most optimization methods used are Newton-Raphson, Gauss-Newton, and steepest-descent methods. A time-varying gain G_k is various for every trail is as result of the method of Newton-Raphson.

The form exposed in equation (2-14) clarifies the update algorithm.

$$u_{k+1} = u_k + \epsilon_k T_p^* e_k \tag{2-14}$$

where T_p^* is the adjoint operator of the system and ϵ_k is a time-varying gain [39].

2.3.3 Norm-Optimal ILC

This approach has been developed in order to reduce the error rate by calculating the changes in the input signal.

$$J_{k+1} = \|e_{k+1}\|^2 + \lambda\|u_{k+1} - u_k\|^2 \tag{2-15}$$

The results will be similar to:

$$u_{k+1} = u_k + G^* e_{k+1} \tag{2-16}$$

Where G^* denote the adjoint operator of the system [45] [46].

2.3.4 Frequency-Domain approaches

Iterative learning control from the frequency domain viewpoint has been considered by several researchers like [47] where the input update law defends as:

$$U_{k+1}(s) = L(s)[U_k(s) + aE_k(s)] \tag{2-17}$$

A non-zero error will be produced by using this algorithm according to what illustrated in [40], in spite of that, Arimoto took his algorithm in the frequency domain [48]. In the same last domain, Luce et al. [49] have designed the operators T_e and T_u

which considered as complete free operators for ILC algorithms, and there are many frequency domains results have been considered by other researchers [50] and [51].

2.3.5 Discrete-Time systems

Iterative learning control algorithm has treated by some researchers as a system belongs to discrete-time form, all of its simulations are done depending on this consideration [39].

By consideration of Figure 2.1 the discrete-time definition will be:

$$u_k = \big(u_k(0), u_k(1), \cdots, u_k(N-1)\big), \tag{2-18}$$

$$y_k = \big(y_k(m), y_k(m+1), \cdots, y_k(N-1+m)\big), \tag{2-19}$$

$$y_d = \big(y_d(m), y_d(m+1), \cdots, y_d(N-1+m)\big), \tag{2-20}$$

For the linear plant, the symbol k denote to the number of trial, the symbol m denotes to the relative gain, and the symbol N denotes to trial's length. Let $m = 1$, the truncated $l_\infty - norm,$ will be taken as:

$$\|x\|_\infty = \max_{1\le i\le N} |x_i| \tag{2-21}$$

As a result, the norm of the matrix H will be found by:

$$\|H\|_i = \|H\|_\infty = \max_i \left(\sum_{j=1}^{N} |h_{ij}| \right) \tag{2-22}$$

The linear dynamic system can be written as $y_k = Hu_k$, the matrix H is Markov parameters elements matrix which represents the matrix of rank N as seen below:

$$H = \begin{bmatrix} h_1 & 0 & 0 & \cdots & 0 \\ h_2 & h_1 & 0 & \cdots & 0 \\ h_3 & h_2 & h_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_N & h_{N-1} & h_{N-2} & \cdots & h_1 \end{bmatrix} \tag{2-23}$$

ILC algorithm for this situation can be considered as:

$$u_{k+1} = u_k + Ae_k \tag{2-24}$$

$$A = \begin{bmatrix} \alpha_1 y_d(1) & 0 & \cdots & 0 \\ \alpha_2 y_d(2) & \alpha_1 y_d(1) & \cdots & 0 \\ \alpha_3 y_d(3) & \alpha_2 y_d(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_N y_d(N) & \alpha_{N-1} y_d(N-1) & \cdots & \alpha_1 y_d(1) \end{bmatrix} \tag{2-25}$$

So the algorithm (2-24) could be arranged in the form:

$$u_k = A_k y_d \tag{2-26}$$

where A_k is took as an inverse matrix approximately, which be updated at the end of each trial according to the expression below:

$$A_{k+1} = A_k + \Delta A_k \tag{2-27}$$

Where ΔA_k is:

$$\Delta A_k = \begin{bmatrix} \alpha_1 e_k(1) & 0 & 0 & \cdots & 0 \\ \alpha_1 e_k(2) & \alpha_2 e_k(1) & 0 & \cdots & 0 \\ \alpha_1 e_k(3) & \alpha_2 e_k(2) & \alpha_3 e_k(1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_1 e_k(N) & \alpha_2 e_k(N-1) & \alpha_3 e_k(N-2) & \cdots & \alpha_N e_k(1) \end{bmatrix} \tag{2-28}$$

CHAPTER THREE

LEAST SQUARES PARAMETER IDENTIFICATION

The estimation theory is one of the statistics branches which deal with evaluating the values of parameters relied on measured empirical data which has a random component [52]. Although estimation theories are related to statistics, but also they are used in several fields like signal processing, control, and trajectory tracking. There are several theories for estimation parameters like maximum likelihood method, least squares method, estimating equations, and output method. In this chapter, the least squares approximation method will be described with its branches.

Least squares approximation can be used in extensive various applications, including parameter identification, for instance computing the properties of damping of a damper, for fluid-filled and works as a function of temperature, the dynamic identification of air plant and coefficients of static aerodynamic, and determination of orbit and attitude. Even the strategies of modern control, for example, some of the adaptive controllers, use the least squares approximation in order to update the parameters in the controller of the dynamic system. There are three quantities of interest for any parameter or variable in estimation:

1. The true value: or "truth", this value denotes the real value required of the quantity being approximated by the parameter estimator. The true values will be represented as unadorned symbols like x, y, z … etc.
2. The measured value: it is the coming signal from a sensor directly. Since measurements always contain errors, they are never perfect. So, measured values generally can be considered as the true values plus some error. The measured values are represented by the symbol $(\sim)$ like $\tilde{x}, \tilde{y}, \tilde{z}$ … etc.
3. The estimated value: this value can be calculated from the estimation process itself, which are represented as $\hat{x}, \hat{y}, \hat{z}$ … etc.

Other values used commonly in parameter estimation, measurement error and residual error. Where the measurement error υ is the difference between measurement value $\tilde{x}$ and the true value x:

$$\tilde{x} = x + \upsilon \qquad (3\text{-}1)$$

and the residual error e is the difference between measurement value $\tilde{x}$ and the estimated value $\hat{x}$.

$$\tilde{x} = \hat{x} + e \qquad (3\text{-}2)$$

The true value and the real measured value cannot be known practically, the generation of the mechanism's errors happening according to some of the statistical properties. The residual error e can be founded easily just after finding the estimated value, also it is used to drive the algorithm of the estimator itself. It has to be taken into consideration that both of the errors have importance in determining the features of the estimator [53].

3.1 Linear Batch Estimation Method

In order to make central principles to solve wide field of estimation problems, the estimation method of linear least squares approximation of Gauss will be stated in this part of this monograph. By taking known discrete-time period t_k, a batch of measured values $\tilde{y}_k$ could be found of the process $y(t)$:

$$\{\tilde{y}_1, t_1;\ \tilde{y}_2, t_2; \dots;\ \tilde{y}_m, t_m\} \qquad (3\text{-}3)$$

and a suggested mathematical model of the form

$$y(t) = \sum_{i=1}^{n} x_i\, h_i(t), \qquad m \geq n \qquad (3\text{-}4)$$

where

$$h_i(t) \in \{h_1(t), h_2(t), \dots, h_n(t),\} \qquad (3\text{-}5)$$

are a batch of independent identified base functions. From equation (3-4), it results that the two variables x and y are related to each other.

Depending on measures of "how well' expression in (3-4) to find the optimal value of x been clearly reasonable from the measurements in (3-3). A batch of estimates have been looked for, this batch donated by the values $\{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n\}$ which will be used in the equation (3-4) to estimate $y(t)$, even though, there are several ways like measurements errors, choosing wrong value of x, or modeling error that can rise up errors between the true and the estimated values. From the equation (3-4), it has been considered that both of the estimated output $\hat{y}_k$ and the measured values $\tilde{y}_k$ are related to the true and the estimated values.

$$\tilde{y}_k \equiv \tilde{y}(t_k) = \sum_{i=1}^{n} x_i h_i(t_k) + v_k, \qquad k = 1, 2, \dots, m \tag{3-6}$$

$$\hat{y}_k \equiv \hat{y}(t_k) = \sum_{i=1}^{n} \hat{x}_i h_i(t_k) \qquad k = 1, 2, \cdots, m \tag{3-7}$$

where v_k represents the measurement error. The measurement process which has been assumed is modelled by Equation (3-6).

Consider the next equation:

$$\tilde{y}_k = \sum_{i=1}^{n} \hat{x}_i h_i(t_k) + e_k, \qquad k = 1, 2, \cdots, m \tag{3-8}$$

where the residual error e_k is defined as

$$e \equiv \tilde{y}_k - \hat{y}_k \tag{3-9}$$

Equation (3-8) is rewritten in compact matrix form as:

$$\tilde{\mathbf{y}} = H\tilde{\mathbf{x}} + \mathbf{e} \tag{3-10}$$

where

The measured values of y	$\longrightarrow$	$\tilde{\mathbf{y}} = [\tilde{y}_1\ \tilde{y}_2 \cdots \tilde{y}_m]^T$	(3-11)
The residual errors	$\longrightarrow$	$\mathbf{e} = [e_1\ e_2 \cdots e_m]^T$	(3-12)
The estimated values of x	$\longrightarrow$	$\hat{\mathbf{x}} = [\hat{x}_1\ \hat{x}_2 \cdots \hat{x}_m]^T$	(3-13)

$$H = \begin{bmatrix} h_1(t_1) & h_2(t_1) & \cdots & h_n(t_1) \\ h_1(t_2) & h_2(t_2) & \cdots & h_n(t_2) \\ \vdots & \vdots & & \vdots \\ h_1(t_m) & h_2(t_m) & \cdots & h_n(t_m) \end{bmatrix} \tag{3-14}$$

Equations (3-6) and (3-7) can be summarized as:

$$\tilde{\mathbf{y}} = H\mathbf{x} + \mathbf{v} \tag{3-15}$$

$$\hat{\mathbf{y}} = H\hat{\mathbf{x}} \tag{3-16}$$

where

The measured values of y	$\longrightarrow$	$\tilde{\mathbf{y}} = [\tilde{y}_1\ \tilde{y}_2\ \cdots\ \tilde{y}_n]^T$	(3-17)
The true values x	$\longrightarrow$	$\mathbf{x} = [x_1\ x_2\ \cdots\ x_n]^T$	(3-18)
The estimated values of y	$\longrightarrow$	$\hat{\mathbf{y}} = [\hat{y}_1\ \hat{y}_2\ \cdots\ \hat{y}_n]^T$	(3-19)
The measurement errors	$\longrightarrow$	$\mathbf{v} = [v_1\ v_2\ \cdots\ v_n]^T$	(3-20)

Both of the mathematical expressions, (3-10) and (3-15), which normally denoted to what called “observation equations” if $\hat{\mathbf{x}} = \mathbf{x}$, and if there is available zero model errors [53].

3.1.1 Linear least squares method

One of the famous methods to find the optimum choice for the unknown parameters is linear least squares method by Gauss. In this method, $\hat{x}$ minimizes the sum squares of the residual errors that observed in:

$$J = \frac{1}{2}\mathbf{e}^{T}\mathbf{e} \tag{3-21}$$

To find the error **e** sub-equation (3-10) in (3-21) and take in the calculation that a scalar equals its swap yields:

$$J = J(\hat{\mathbf{x}})\frac{1}{2}(\tilde{\mathbf{y}}^{T}\tilde{\mathbf{y}} - 2\tilde{\mathbf{y}}^{T}H\hat{\mathbf{x}} + \hat{\mathbf{x}}^{T}H^{T}H\hat{\mathbf{x}}) \tag{3-22}$$

In order to minimize J, a suitable $\hat{\mathbf{x}}$ should be found. Using the matrix calculus differentiation instructions [53], some requirements will be got by following that for a global minimum of the quadratic function of the above expression (3-22) as follow:

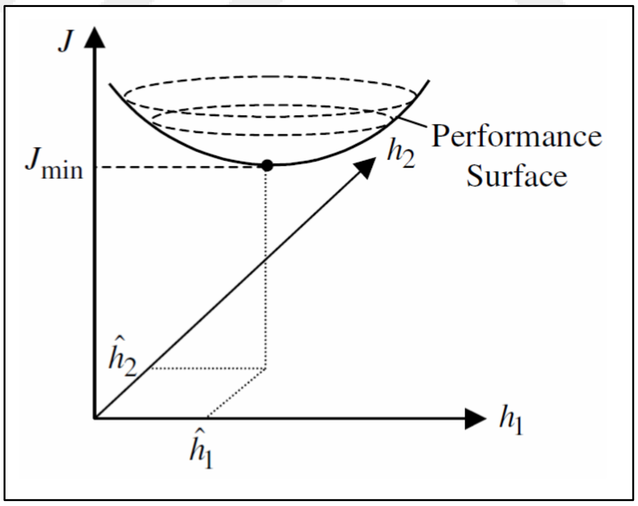


Figure 3.1: Surface of the Convex Performance Order $n = 2$ Problem [53].

necessary condition

$$\nabla_{\hat{\mathbf{x}}}J \equiv \begin{bmatrix} \dfrac{\partial J}{\partial \hat{x}_1} \\ \vdots \\ \dfrac{\partial J}{\partial \hat{x}_n} \end{bmatrix} = H^{T}H\hat{\mathbf{x}} - H^{T}\tilde{\mathbf{y}} = 0 \tag{3-23}$$

sufficient condition

$$\nabla_{\hat{\mathbf{x}}}^2 J \equiv \frac{\partial^2 J}{\partial \hat{\mathbf{x}} \partial \hat{\mathbf{x}}^T} = H^T H \tag{3-24}$$

equation (3-24) must be positive definite, where $\nabla_{\hat{\mathbf{x}}} \mathrm{J}$ represents the Jacobian matrix and $\nabla_{\hat{\mathbf{x}}}^2 \mathrm{J}$ represents the Hessian matrix. By considering the sufficient condition, the positive semi-definite called on any B matrix like that in $\mathbf{x}^T B \mathbf{x} \geq 0$ for all $\mathbf{x} \neq \mathbf{0}$. By setting and squaring $\mathbf{h} = H\mathbf{x}$, the scalar $h^2 = \mathbf{h}^T \mathbf{h} \geq 0$ can easily be obtained, where, $H^T H$ is always positive semi-definite, and when $H$is maximum rank (n) then it becomes positive definite.

The space performance at $n + 1$ represents the function J, which has a convex shape of parabola, the dimension of this parabola is (n), and it has one distinct minimum point, for instance to what mentioned before, the bowl-shape surface with $n = 2$ just like shown in Figure 3.1.

From the equation (3-23), and by considering its necessary condition, the "normal equation" is existed now:

$$(H^T H)\hat{\mathbf{x}} = H^T \tilde{\mathbf{y}} \tag{3-25}$$

The result of $H^T H$ will be positive definite when the rank of the matrix H equal to n, at this situation the inverse of this matrix can be implemented in order to get the clear result of optimal estimate below:

$$\hat{\mathbf{x}} = (H^T H)^{-1} H^T \tilde{\mathbf{y}} \tag{3-26}$$

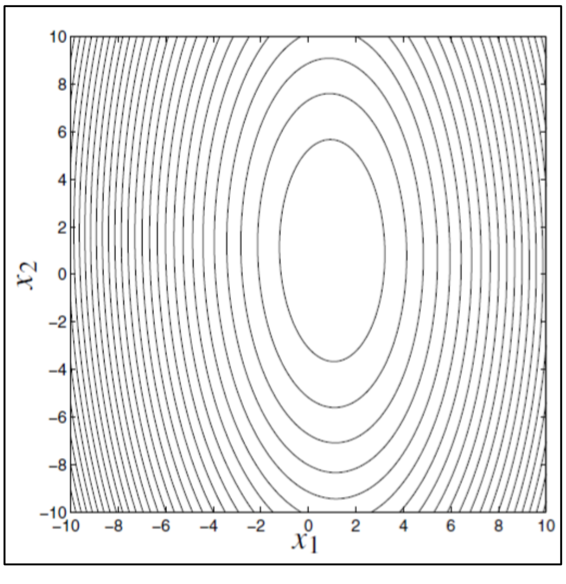


Figure 3.2: (a) Outline Plots for Observable System.

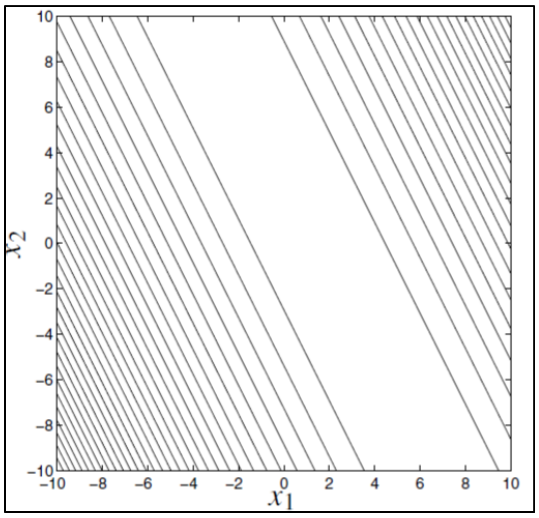


Figure 3.2: (b) Outline Plots for Unobservable System.

The difference between the equations (3-10) and (3-26) is that the first one is Gauss' "equations of condition" but the last one represents one of the most useful algorithms in solving easy least squares problems. The inverse of $H^T H$ is useful in calculating $\hat{\mathbf{x}}$, and that inverse is only available when the observations number is equivalent to or larger than the number of unknown x_i.

If the two expressions H_1 and H_1 took in consideration as $[\sin t \quad 2\cos t]$ and $[\sin t \quad 2\sin t]$ respectively, and **x** took as$[1 \quad 1]^T$, that will lead clearly to understand that H_1 can provide a set of independent and linear functions, not like the other expression H_2 which did not do that due to the second column of its matrix which twice of its first one. The Figure 3.2 (a) shows the minimum value of $\mathbf{x} = [1 \quad 1]^T$ when H_1 is used, not like the case of using H_2 which could not implement the minimum values and it could make infinite of solutions as seen in Figure 3.2 (b).

The least squares method has some advantages, one of these advantages is the equality between the number of unknown parameters and the order of the matrix inverse.

3.1.2 Weighted least squares

The method of least square shown in (3-21) is minimized in order to calculate $\hat{\mathbf{x}}$, Implies an equal concentration on each $\tilde{y}_k$ measurement. Because of what is commonly reported that measurements occur in with inaccuracy, so the question has become how to choose the appropriate values of these weights which can be selected instinctively for each amount, where these weights are inversely proportional to the measure the estimated accuracy, for instance, infinity weight is suitable for zero error measurement, and zero weight is suitable for infinite error measurement. For the sake of combine appropriate weighting, a standard form of lest square has been set as below:

$$J = \frac{1}{2}\mathbf{e}^{T} W \mathbf{e} \tag{3-27}$$

We now seek to calculate $\hat{\mathbf{x}}$ that minimize J, where W is an $m \times m$ symmetric matrix because the terms $e_i e_j, i \neq j$, are always weighted equally with the corresponding $e_j e_i$ terms. In order that $\hat{\mathbf{x}}$ yield a minimum of equation (3-27), we have the requirements:

necessary condition

$$\nabla_{\hat{\mathbf{x}}} J = H^T W H \mathbf{x} - H^T W \tilde{\mathbf{y}} = \mathbf{0} \tag{3-28}$$

sufficient condition

$$\nabla^2_{\hat{\mathbf{x}}} J = H^T W H \tag{3-29}$$

where the equation (3-29) must be positive definite.

The solution for $\hat{\mathbf{x}}$ has been obtained from the necessary conditions in the equation (3-28), the obtaining of $\hat{\mathbf{x}}$ is given by:

$$\hat{\mathbf{x}} = (H^T W H)^{-1} H^T W \tilde{\mathbf{y}} \tag{3-30}$$

Also, equation (3-29) shows that the weight W have to be positive definite.

3.2 Linear Sequential Estimation Method

An implicit assumption is present in developing the preceding section, this means that all measurements exist for the processing at the same time. The measurements exist successively in subsets in many application in the world, it may be recommendable to identify new estimates relied on all preceding measurements.

Only the following two subsets are considered to simplify the initial discussion:

$$\begin{aligned}\tilde{\mathbf{y}}_1 &= [\tilde{y}_{11} \quad \tilde{y}_{12} \quad \cdots \quad \tilde{y}_{1m_1}]^T \\ &= \text{an } m_1 \times 1 \text{ vector of measurments}\end{aligned} \tag{3-31}$$

$$\begin{aligned}\tilde{\mathbf{y}}_2 &= [\tilde{y}_{21} \quad \tilde{y}_{22} \quad \cdots \quad \tilde{y}_{2m_2}]^T \\ &= \text{an } m_2 \times 1 \text{ vector of measurments}\end{aligned} \tag{3-32}$$

and the associated observation equations

$$\tilde{\mathbf{y}}_1 = H_1 \mathbf{x} + \mathbf{v}_1 \tag{3-33}$$

$$\tilde{\mathbf{y}}_2 = H_2 \mathbf{x} + \mathbf{v}_2 \tag{3-34}$$

Where:

H_1 = an $m_1 \times n$ known coefficient matrix of maximum rank $n \leq m_1$

H_2 = an $m_2 \times n$ known coefficient matrix

$\mathbf{v}_1, \mathbf{v}_2$ = vector of measurement errors

$\mathbf{x}$ = the $n \times 1$ vector of unknown parameters

The least squares estimate, $\hat{\mathbf{x}}$, of $\mathbf{x}$ based upon the first measurement subset (3-31) follows from equation (3-30) as:

$$\hat{\mathbf{x}}_1 = (H_1^T W_1 H_1)^{-1} H_1^T W_1 \tilde{\mathbf{y}}_1 \tag{3-35}$$

Where W_1 is an $m_1 \times m_1$ symmetric, positive matrix related with measurements $\tilde{\mathbf{y}}_1$. It is possible to consider $\tilde{\mathbf{y}}_1$and $\tilde{\mathbf{y}}_2$ in the same time and determine an estimate $\hat{\mathbf{x}}_2$ of $\mathbf{x}$, that estimation based on the subsets of measurement illustrated in (3-31) and (3-32). So that, the compound observation equations have been formed as:

$$\tilde{\mathbf{y}} = H\mathbf{x} + \mathbf{v} \tag{3-36}$$

where

$$\tilde{\mathbf{y}} = \begin{bmatrix} \tilde{\mathbf{y}}_1 \\ \cdot\cdot \\ \tilde{\mathbf{y}}_2 \end{bmatrix}, \quad H = \begin{bmatrix} H_1 \\ \cdot\cdot \\ H_2 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \cdot\cdot \\ \mathbf{v}_2 \end{bmatrix} \tag{3-37}$$

Then, the compound matrix of weight is assumed to take block diagonal shape, so its shape will be:

$$W = \begin{bmatrix} W_1 & \vdots & 0 \\ \cdots & & \cdots \\ 0 & \vdots & W_2 \end{bmatrix} \tag{3-38}$$

Then, by depending on the first two subsets of measurement which follows from the equation (3-30), the optimal least squares estimate will be:

$$\hat{\mathbf{x}}_2 = (H^T W H)^{-1} H^T W \tilde{\mathbf{y}} \tag{3-39}$$

By taking into consideration that the matrix W is a diagonal matrix, and by expanding the equation (3-39) will lead to:

$$\hat{\mathbf{x}}_2 = [H_1^T W_1 H_1 + H_2^T W_2 H_2]^{-1} (H_1^T W_1 \tilde{\mathbf{y}}_1 + H_2^T W_2 \tilde{\mathbf{y}}_2) \tag{3-40}$$

By using the above procedure, it is clearly conceivable that it is possible to keep on founding compound equations and solving them to find the new optimal estimates as in equation (3-40). Though, it is not taken into consideration the benefit of calculations that done on with the previous subsets of data effectively in the method that illustrated above, the main point of using least square is basically to make efficient use of earlier estimates and connected side calculations by calculations arrangement to find the next estimate (e.g., $\hat{\mathbf{x}}_{\mathbf{2}}$).

Let's be the defining the following variables is the start of this approach:

$$P_1 \equiv [H_1^T W_1 H_1]^{-1} \tag{3-41}$$

$$P_2 \equiv [H_1^T W_1 H_1 + H_2^T W_2 H_2]^{-1} \tag{3-42}$$

By (assuming that both ${P_1}^{-1}$ and ${P_2}^{-1}$ exist)

$${P_2}^{-1} = {P_1}^{-1} + H_2^T W_2 H_2 \tag{3-43}$$

We now rewrite equation (3-35) and (3-36) using the definitions in equation (3-41) and (3-42) as:

$$\hat{\mathbf{x}}_{\mathbf{1}} = P_1 H_1^T W_1 \tilde{\mathbf{y}}_{\mathbf{1}} \tag{3-44}$$

$$\hat{\mathbf{x}}_{\mathbf{2}} = P_2 (H_1^T W_1 \tilde{\mathbf{y}}_{\mathbf{1}} + H_2^T W_2 \tilde{\mathbf{y}}_{\mathbf{2}}) \tag{3-45}$$

Pre-multiplying equation (3-44) by ${P_1}^{-1}$ yields

$${P_1}^{-1} \hat{\mathbf{x}}_{\mathbf{1}} = H_1^T W_1 \tilde{\mathbf{y}}_1 \tag{3-46}$$

Next, from equation (3-43) we have

$${P_1}^{-1} = {P_2}^{-1} - H_2^T W_2 H_2 \tag{3-47}$$

Putting the equation (3-47) into equation (3-46) leads to:

$$H_1^T W_1 \tilde{\mathbf{y}}_1 = {P_2}^{-1} \hat{\mathbf{x}}_1 - H_2^T W_2 H_2 \hat{\mathbf{x}}_1 \tag{3-48}$$

At last, putting equation (3-48) into equation (3-45) and gathering terms gives:

$$\hat{\mathbf{x}}_2 = \hat{\mathbf{x}}_1 + K_2 (\tilde{\mathbf{y}}_2 - H_2 \hat{\mathbf{x}}_1) \tag{3-49}$$

where

$$K_2 \equiv P_2 H_2^T W_2 \tag{3-50}$$

We now have a mechanism to consecutively provide an updated estimate, $\hat{\mathbf{x}}_2$, depended on the previous estimate, $\hat{\mathbf{x}}_1$, and connected side calculations. We can easily generalize equation (4-1) and (4-2) to use the k^{th} estimate to determine estimate at k+1 from the k+1 subset of measurements, which leads to a most important result in sequential estimation theory:

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k + K_{k+1} (\tilde{\mathbf{y}}_{k+1} - H_{k+1} \hat{\mathbf{x}}_k) \tag{3-51}$$

where:

$$K_{k+1} = P_{k+1} H_{k+1}^T W_{k+1} \tag{3-52}$$

$$P_{k+1}^{-1} = P_k^{-1} + H_{k+1}^T W_{k+1} H_{k+1} \tag{3-53}$$

Equation (4-1) modifies the previous best correction $\hat{\mathbf{x}}_k$ by an additional correction to account for the information contained in the $k+1$ measurement subset. This equation is a Kalman update equation for computing the improved estimate $\hat{\mathbf{x}}_{k+1}$.

Equation (4-2) is the correction term, known as the Kalman gain matrix. The sequential least squares algorithm plays an essential role for linear (and nonlinear) dynamic state estimation. Equation (3-51) is in fact a linear difference equation, usually found in digital control analysis. This equation may be rearranged as:

$$\hat{\mathbf{x}}_{k+1} = [I - K_{k+1}H_{k+1}]\hat{\mathbf{x}}_k + K_{k+1}\tilde{\mathbf{y}}_{k+1} \tag{3-54}$$

Which clearly is in the form of a time-varying dynamical system. Therefore, linear tools can be used to check stability, dynamic response times, etc.

The specific form for P^{-1} in equation (3-53) is known as the information matrix recursion. The current approach for computing P_{k+1} includes computing the inverse of equation (3-53) which offers no advantage over upsetting the normal equations in their original batch processing in equation (3-39). This is because of the fact that an $n \times n$ inverse must still be implemented. We may think about whether there is a less demanding approach to P_{k+1} given that we have calculated P_k before. As shown before, when the measurements' number m in the new data subset is small compared to n (as is usually the case), a small rank adjustment to the already computed P_k can be calculated efficiently using the Sherman-Morrison-Woodbury matrix inversion lemma. Let

$$F = [A + BCD]^{-1} \tag{3-55}$$

where:

$$F = an\ arbitrary\ n \times n\ matrix$$

$$A = an\ arbitrary\ n \times n\ matrix$$

$$B = an\ arbitrary\ n \times m\ matrix$$

$$C = an\ arbitrary\ m \times m\ matrix$$

$$D = an\ arbitrary\ m \times n\ matrix$$

Then, assuming all inverses exist

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

The matrix inversion lemma can be proved by showing that $F^{-1}F = I$. Brute force calculation of $F^{-1}F$ gives

$$F^{-1}F = I - B[(DA^{-1}B + C^{-1})^{-1} - C + CDA^{-1}B(DA^{-1}B + C^{-1})^{-1}]DA^{-1} \tag{3-56}$$

It is sufficient to demonstrate that the amount of the square sections of equation (3-56) is indistinguishably zero, in order to verify the matrix inversion lemma. So, we need to verify that

$$(DA^{-1}B + C^{-1})^{-1} = C - CDA^{-1}B(DA^{-1}B + C^{-1})^{-1} \tag{3-57}$$

Right multiplying both sides of equation (3-57) by $(DA^{-1}B + C^{-1})$ reduce equation (3-57) to

$$I = C(DA^{-1}B + C^{-1}) - CDA^{-1}B \tag{3-58}$$

This finishes the proof.

Applying the matrix reversal lemma to equation (3-53) is the subsequent step. The F, A, B, C, and D are the "judicious choices" which can be calculated by:

$$F = P_{k+1} \tag{3-59}$$

$$A = P_k^{-1} \tag{3-60}$$

$$B = H_{k+1}^T \tag{3-61}$$

$$C = W_{k+1} \tag{3-62}$$

$$D = H_{k+1} \tag{3-63}$$

The matrix information recursion now becomes

$$P_{k+1} = P_k - P_k H_{k+1}^T (H_{k+1} P_k H_{k+1}^T + W_{k+1}^{-1})^{-1} H_{k+1} P_k \tag{3-64}$$

Thus, P_{k+1}, which is used in equation (3-52), can be obtained by "updating" P_k, and the update process usually required inverting a matrix with rank less than *n*. A large number of following applications of the recursion (3-64) rarely introduces mathematics errors which can cancel the estimates.

The "update equation" (3-51) can also be rearranged in several alternate forms. One of the more common is obtained by substituting equation. (3-64) into the equation. (3-52) to obtain

$$\begin{aligned} K_{k+1} = [P_k - P_k H_{k+1}^T (H_{k+1} P_k H_{k+1}^T + W_{k+1}^{-1})^{-1} H_{k+1} P_k] \\ \times H_{k+1}^T W_{k+1} \end{aligned} \tag{3-65}$$

$$= P_k H_{k+1}^T [I - (H_{k+1} P_k H_{k+1}^T + W_{k+1}^{-1})^{-1} H_{k+1} P_k H_{k+1}^T] W_{k+1} \tag{3-66}$$

Now, factoring $(H_{k+1} P_k H_{k+1}^T + W_{k+1}^{-1})^{-1}$ outside of the square brackets leads directly to

$$\begin{aligned} K_{k+1} = P_k H_{k+1}^T (H_{k+1} P_k H_{k+1}^T + W_{k+1}^{-1})^{-1} \\ \times [W_{k+1}^{-1} + H_{k+1} P_k H_{k+1}^T - H_{k+1} P_k H_{k+1}^T] W_{k+1} \end{aligned} \tag{3-67}$$

This leads to the *covariance recursion form*, given by

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_k + K_{k+1}(\tilde{\mathbf{y}}_{k+1} - H_{k+1}\hat{\mathbf{x}}_k) \tag{3-68}$$

where

$$K_{k+1} = P_k H_{k+1}^T [H_{k+1} P_k H_{k+1}^T + W_{k+1}^{-1}]^{-1} \tag{3-69}$$

$$P_{k+1} = [I - K_{k+1} H_{k+1}] P_k \tag{3-70}$$

The covariance type of successive least squares is most usually utilized as a part of training since it is all the more computationally productive. Be that as it may, the data shape might be numerically prevalent in the introduction arrange. The procedure may begin at any progression by form the earlier estimate $\hat{\mathbf{x}}_1$ and estimate P_1. For initialization, the first data subset can be used if a previous estimates are not obtainable, that could be done by using a batch least squares to determine $\hat{\mathbf{x}}_q$ and P_q, where $q \geq n$. Then the sequential least squares algorithm can be invoked for $k \geq q$. However, sequential least squares can still be used for$k = 1, 2, \cdots, q-1$ if one uses:

$$P_1 = \left[\frac{1}{\alpha^2} I + H_1^T W_1 H_1\right]^{-1} \tag{3-71}$$

$$\hat{\mathbf{x}}_1 = P_1 \left[\frac{1}{\alpha} \beta + H_1^T W_1 \tilde{\mathbf{y}}_1\right] \tag{3-72}$$

Where α is a very “large” number and β is a vector of very “small” numbers. It can be shown that the resulting recursive least squares values of P_n and $\hat{\mathbf{x}}_n$ are very close to the corresponding batch values at time t_n.

If the model is, in fact, linear and if there is no correlation between measurement errors of different measurement subsets (so that the assumed block structure of W is strictly valid), then the sequential solution for $\hat{\mathbf{x}}$ in equation (3-51) will agree exactly with the batch solution in equation (3-30), to within arithmetic errors. This is because equation (3-51) is simply an algebraic rearrangement of the normal equations (3-30).

3.3 Error Output Method (Implementation of Least Squares Method for Dynamic System)

For nonlinear dynamic systems like aircraft dynamics, error output of unknown aerodynamic coefficients or stability and control derivatives is used to quantify the performance of these particular dynamic systems. These models are often used to design control systems to provide increased maneuverability and for use in the design of automated unpiloted vehicles. Generally, such coefficients are typically first determined using experimental methods like wind tunnel applications for the aircraft model, and as a newer approach, using computational fluid dynamics. Error output using the dynamic system measurement data is beneficial to provide a final verification of these coefficients, and also update models for other applications such as adaptive control algorithms. This section introduces the basic ideas which incorporate estimation principles for the dynamic system error output from the measured data [54].

Identification methods' applications for the coefficients of aircraft goes back to the mid-1920s, which included the fundamental location of ratios of damping and frequencies. In the 1940s and mid-1950s, frequency response data have fitted by these coefficients (magnitude and phase). Around a similar time, linear least squares were applied using flight data, but gave poor results in the presence of measurement noise and gave biased estimates. Iliff (1989) described another method like time vector techniques and analog matching [54].

The most popular approaches today for aircraft or similar dynamic models coefficient identification are based on maximum likelihood techniques. The desirable attributes of these techniques, such as asymptotically unbiased and consistent estimates, are especially useful for the estimation of aircraft coefficients in the presence of measurement errors associated with flight data

The aircraft model equations of motion can be written in continuous- discrete form as shown below:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{p}) \tag{3-73}$$

$$\tilde{\mathbf{y}}_{\boldsymbol{k}} = \mathbf{h}(t_k, \mathbf{x}_{\boldsymbol{k}}) + \mathbf{v}_{\boldsymbol{k}} \tag{3-74}$$

Where $\mathbf{x}$ is the $n \times 1$ state vector (e.g., angle of attack, pitch angle, body rates, etc.), $\mathbf{p}$ is the $q \times 1$ vector of aircraft coefficients to be determined, y is the $m \times 1$ measurement vector, and $\mathbf{v}$ is the $m \times 1$ measurement-error vector which is assumed to be represented by a zero-mean Gaussian noise process with covariance R. Note that there is no noise associated with the state vector model. Modelling errors may also be present, which lead to several obvious complications. However, the most common approach is to ignore it; any modelling error is most often treated as state or measurement noise, or both, in spite of the fact that the modelling error may be predominately deterministic rather than random.

The maximum likelihood estimation approach minimizes the following loss function:

$$J(\hat{\boldsymbol{p}}) = \frac{1}{2}\sum_{k=1}^{N}(\tilde{\boldsymbol{y}}_k - \hat{\boldsymbol{y}}_k)^T R^{-1}(\tilde{\boldsymbol{y}}_k - \hat{\boldsymbol{y}}_k) \tag{3-75}$$

where $\hat{\boldsymbol{y}}_k$ is the estimated response of $\mathbf{y}$ at time t_k for a given value of the unknown parameter vector $\boldsymbol{p}$, and *N* is the total number of measurements. A common approach to minimize equation (4-1) for dynamic system parameter identification involves using the Newton-Raphson algorithm. If i is the iteration number, then the $i+1$ estimate of $\boldsymbol{p}$, denoted by $\hat{\boldsymbol{p}}$, is obtained from the i^{th} estimate by:

$$\hat{\boldsymbol{p}}_{i+1} = \hat{\boldsymbol{p}}_i - \left[\nabla^2_{\hat{\boldsymbol{p}}} J(\hat{\boldsymbol{p}})\right]^{-1} \left[\nabla_{\hat{\boldsymbol{p}}} J(\hat{\boldsymbol{p}})\right] \tag{3-76}$$

where the first and second gradients are defined as:

$$\left[\nabla_{\hat{\boldsymbol{p}}} J(\hat{\boldsymbol{p}})\right] = -\sum_{k=1}^{N} \left[\nabla_{\hat{\boldsymbol{p}}} \hat{\boldsymbol{y}}_k\right]^T R^{-1} (\tilde{\boldsymbol{y}}_{\boldsymbol{k}} - \hat{\boldsymbol{y}}_{\boldsymbol{k}}) \tag{3-77}$$

$$\left[\nabla^2_{\hat{\boldsymbol{p}}} J(\hat{\boldsymbol{p}})\right] = \sum_{k=1}^{N} \left[\nabla_{\hat{\boldsymbol{p}}} \hat{\boldsymbol{y}}_k\right]^T R^{-1} \left[\nabla_{\hat{\boldsymbol{p}}} \hat{\boldsymbol{y}}_k\right] - \sum_{k=1}^{N} \left[\nabla^2_{\hat{\boldsymbol{p}}} \hat{\boldsymbol{y}}_k\right] R^{-1} (\tilde{\boldsymbol{y}}_{\boldsymbol{k}} - \hat{\boldsymbol{y}}_{\boldsymbol{k}}) \tag{3-78}$$

The Gauss-Newton approximation to the second gradient is given by

$$\left[\nabla^2_{\hat{\boldsymbol{p}}} J(\hat{\boldsymbol{p}})\right] \approx \sum_{k=1}^{N} \left[\nabla_{\hat{\boldsymbol{p}}} \hat{\boldsymbol{y}}_k\right]^T R^{-1} \left[\nabla_{\hat{\boldsymbol{p}}} \hat{\boldsymbol{y}}_k\right] \tag{3-79}$$

This approximation is easier to compute than an equation. (3-78), and has the advantage of possible decreased convergence time.

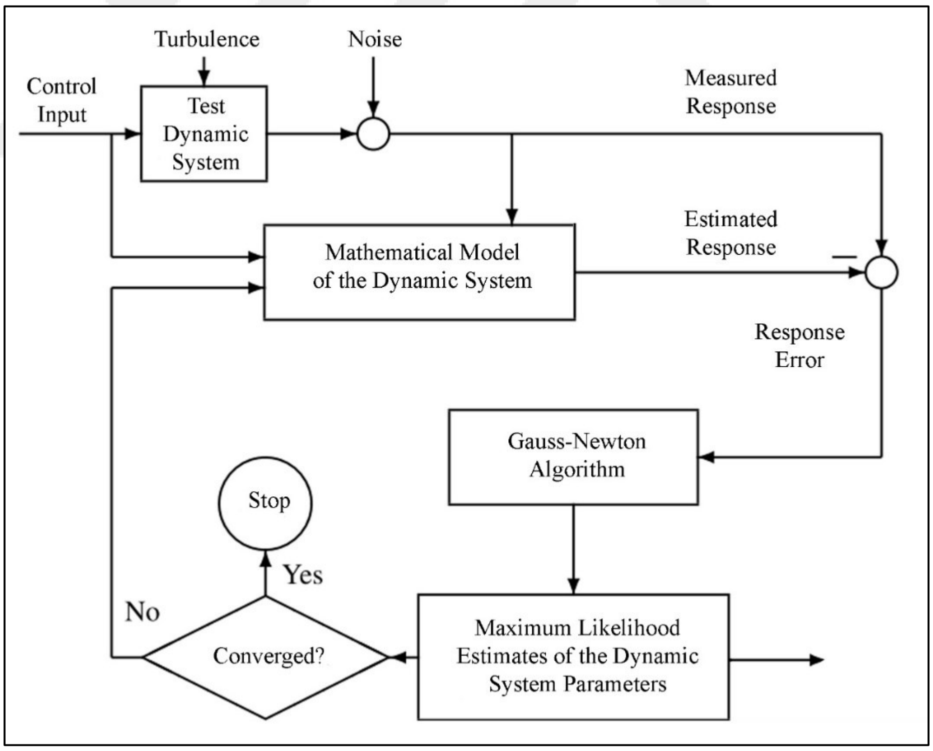


Figure 3.3: Dynamic System Parameter Identification [54].

The dynamic system parameter identification process using maximum-likelihood. First of all, a control input is introduced to stimulate the motion. This input should be “rich” enough so that the test dynamic system undergoes a general motion to allow sufficient observability of the to-be-identified parameters. For most applications, it is assumed that the control system inputs adequately lead the motion in comparison to the effects of the turbulence and other unknown disturbances. An estimated response from the mathematical model is computed first using some initial guess of the dynamic system parameters, which are usually obtained from ground-based wind tunnel data or by other means. A response error is computed from the estimated response and measured response. Then equations (3-76), (3-77), and (3-79) are used to provide a Gauss-Newton update of the dynamic system parameters. Next, the convergence is checked using some stopping criterion. If the procedure has not converged then the previous dynamic system parameters are replaced with the newly calculated ones. These newly obtained dynamic system parameters are used to calculate a new estimated response from the mathematical model. The process continues until convergence is achieved. The error-covariance of the estimated parameters is given by the inverse of equation (3-79), which is also alike to within first-order terms to the Cramer-Rao lower bound. Experiments are frequently repeated to confirm consistency. If the results are found to be consistent, then the measurements can be combined to obtain improved estimates.

CHAPTER FOUR

ADAPTIVE ITERATIVE LEARNING CONTROL (AILC) ALGORITHM

Iterative learning control algorithm can control dynamic systems with a repetitive motion which have fixed parameters, if a dynamic system has one or more of these parameters changed, ILC will fail to control the system. So that, the necessity to develop ILC algorithms to have the ability to adapt itself and control such a systems has appeared.

The algorithm that will be illustrated in this chapter represents a combination between Arimoto's *ILC control* algorithm and the *Error output method*, this combined algorithm enables the controller to control the linear dynamic systems even when this system has some unknown parameters after adapting itself to suit the new situation.

In this chapter, the linear dynamic system's equation and its discretization will be illustrated, ILC algorithm and the method of determining its learning gain matrix will be presented, at last, the illustration of AILC algorithm by combining ILC algorithm and least squares approximation will be placed.

4.1 Linear Dynamic System Description

The linear system which will be controlled in this monograph is the (spring-mass-damper) system shown in Figure 4.1:

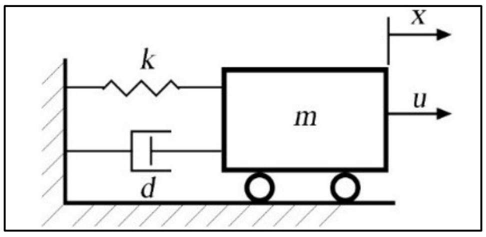


Figure 4.1: Spring-Mass-Damper Dynamic System.

The differential equation of this linear dynamic system and the transfer function shown in the equations (4-1) and (4-2) respectively.

$$\ddot{x} + 2\zeta\omega_n\dot{x} + {\omega_n}^2 x = u \qquad (4\text{-}1)$$

$$G(s) = \frac{1}{S^2 + 2\zeta\omega_n S + \omega_n^2} \qquad (4\text{-}2)$$

Where ζ donates the dimensionless damping ratio, and ω_n is the natural frequency of the system. The pols of the characteristic equation given as described in (4-3).

$$S_1, S_2 = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \qquad (4\text{-}3)$$

When $\zeta < 1$, the roots are complex and the system is underdamped:

$$S_1, S_2 = -\zeta\omega_n \pm j\omega_n\sqrt{\zeta^2 - 1} \qquad (4\text{-}4)$$

When $\zeta = 1$, the roots are repeated and real, and the condition is called critical damped and the S-plane plot of poles and zeros of $y(S)$ is shown in Figure 4.2 where $\theta = cos^{-1}\zeta$. As ζ varies with ω_n constant locus as shown in Figure 4.3.

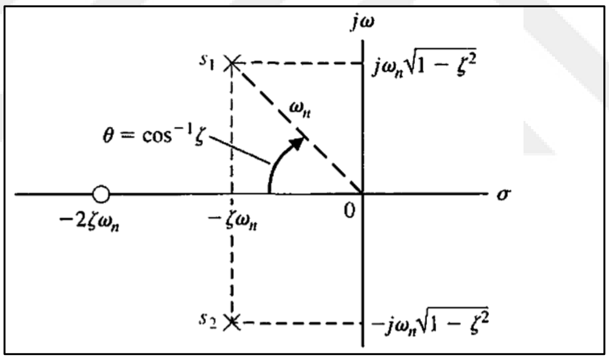

$j\omega$
s_1
$j\omega_n\sqrt{1-\zeta^2}$
ω_n
$\theta = \cos^{-1}\zeta$
σ
$-2\zeta\omega_n$
$-\zeta\omega_n$
0
s_2
$-j\omega_n\sqrt{1-\zeta^2}$

Figure 4.2: An S-plane Plot of The Poles and Zeros of Y(s) [55].

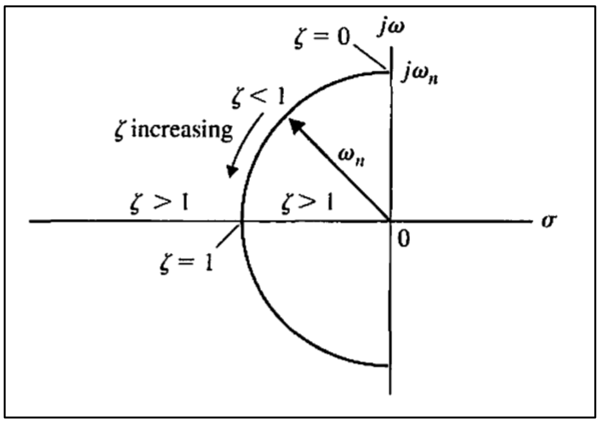


Figure 4.3: The Locus of Roots as $\boldsymbol{\zeta}$ Varies with $\boldsymbol{\omega_n}$ Constant [55].

The transient response is increasingly oscillatory as the roots approach the imaginary axis when ζ approaches zero [55]. The state space form of the system is:

$$\begin{bmatrix}\dot{x}\\ \dot{v}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\omega_n^2 & -2\zeta\omega_n\end{bmatrix}\begin{bmatrix}x\\ v\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}[u] \tag{4-5}$$

The controlled output of the system will be the distance x as expressed in the next equation:

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x\\ v\end{bmatrix} \tag{4-6}$$

4.2 System Discretization

The applied ILC and identification algorithms in this monograph are taken for discrete-time and linear time-invariant (LTI) system, to apply the adaptive iterative learning control (AILC) algorithm on it. The continuous-time system expressed in (4-1) needs to be converted to a discrete-time system equation by applying a discretization method [56]. The continuous-time state space model of the system (4-5) and (4-6) is expressed as:

$$\dot{x} = Ax(t) + Bu(t) \tag{4-7}$$

$$y = Cx(t) \tag{4-8}$$

In general, solving discrete-time equations are easier than differential equations, because it can be solved easily by means of a recursion procedure. The recursion procedure is quite simple and convenient for digital computations.
Consider the following state space equation and output equation in the discrete-time form:

$$x_{k+1} = \Phi x_k + \Theta u_k \tag{4-9}$$

$$y_k = Cx_k \tag{4-10}$$

The solution of equation (4-9) for any positive integer *k* may be obtained directly by recursion as follow:

$$\underbrace{\begin{array}{lll} k = 0, & x_1 = \Phi x_0 + \Theta u_0 & \\ k = 1, & x_2 = \Phi x_1 + \Theta u_1 & = \Phi^2 x_0 + \Phi\Theta u_0 + \Theta u_1 \\ k = 2, & x_3 = \Phi x_2 + \Theta u_2 & = \Phi^3 x_0 + \Phi^2\Theta u_0 + \Phi\Theta u_1 + \Theta u_2 \end{array}}_{\Downarrow}$$

$$x_k = \Phi^k x_0 + \sum_{j=0}^{k-1} \Phi^{k-j-1}\,\Theta u_j \tag{4-11}$$

The equation (4-11) is the response equation for the system exposed in (4-9) in discrete-time form. Clearly, x_k consists of two parts, one represents the contribution of initial state x_0 and the other contribution of the input u_j [56].

$$x_k = \underbrace{\Phi^k x_0}_{Part\ one} + \underbrace{\sum_{j=0}^{k-1} \Phi^{k-j-1}\,\Theta u_j}_{Part\ Two}$$

Where $k = 1, 2, 3, \dots.$, and $j = 0, 1, 2, \dots, k-1$. The output y_k is given by:

$$y_k = C\Phi^k x_0 + C\sum_{j=0}^{k-1} \Phi^{k-j-1}\,\Theta u_j + Du_j \tag{4-12}$$

Obtaining the matrices Φ and Θ can be calculated by using Tylor series[1], Where:

$$\Phi = e^{AT} \tag{4-13}$$

$$\Theta = \int_0^{\lambda} e^{A\lambda}\, Bd\lambda \tag{4-14}$$

Where $\lambda = (k+1)T - \tau$.

The formulas (4-13) and (4-14) can be extracted as follow:

$$\Phi = I + \Delta t + \frac{\Delta t^2}{2!}A^2 + \frac{\Delta t^3}{3!}A^3 + \frac{\Delta t^4}{4!}A^4 + \dots + \frac{\Delta t^k}{k!}A^k \tag{4-15}$$

$$\Theta = \Delta tB + \frac{\Delta t^2}{2!}A^2B + \frac{\Delta t^3}{3!}A^3B + \frac{\Delta t^4}{4!}A^4B + \dots + \frac{\Delta t^k}{k!}A^kB \tag{4-16}$$

4.3 Apply ILC Algorithm on Linear Dynamic System

Main Arimoto's algorithm (2-6) and its error equation (2-7) is used in this monograph.

$$u_{k+1} = u_k + \Gamma e_k$$

$$e_{k(t)} = y_{d(t)} - y_{k(t)}$$

Where Γ is the diagonal learning gain matrix, and convergence is certain if and only if $\|I - CB\Gamma\|_i < 1$. The analysis of the dynamic system equation in the discrete-time form when $t = 0, 1, 2, \dots, N_p$ where N_p is the expected iteration length, and at the iteration k takes the following sequence:

[1] In MATLAB, Φ and Θ matrecis could be found by using the command [Φ,Θ]=c2d(A,B,dt)

$x_k = Ax_k(t-1) + Bu_k(t-1) \quad \Rightarrow \quad x_k(t+0) = A^1x_k(t-1) + Bu_k(t-1)$

$x_k(t+1) = Ax_k(t) + Bu_k(t) \quad \Rightarrow \quad x_k(t+1) = A^2x_k(t-1) + ABu_k(t-1) + Bu_k(t)$

$x_k(t+2) = Ax_k(t+1) + Bu_k(t+1) \quad \Rightarrow \quad x_k(t+2) = A^3x_k(t-1) + A^2Bu_k(t-1) + ABu_k(t) + Bu_k(t+1)$

$x_k(t+3) = Ax_k(t+2) + Bu_k(t+2) \quad \Rightarrow \quad x_k(t+3) = A^4x_k(t-1) + A^3Bu_k(t-1) + A^2Bu_k(t) + ABu_k(t+1) + Bu_k(t+2)$

The above sequence can be written as follows:

$$x_k(t+n) = A^{n+1}x_k(t-1) + A^nBu_k(t-1) + A^{n-1}Bu_k(t) + A^{n-2}Bu_k(t+1) + \cdots + Bu_k(t+n-1) \qquad (4\text{-}17)$$

By depending on the equation (4-11) the equation of the state of the system (4-17) can be expanded in matrix form as follows:

$$\begin{bmatrix} x_k(t) \\ x_k(t+1) \\ x_k(t+2) \\ \vdots \\ x_k(t+n) \end{bmatrix} = \begin{bmatrix} \Phi \\ \Phi^2 \\ \Phi^3 \\ \vdots \\ \Phi^{n+1} \end{bmatrix} x_{k-1}(t) + \begin{bmatrix} \Theta & 0 & 0 & \cdots & 0 \\ \Phi\Theta & \Theta & 0 & \cdots & 0 \\ \Phi^2\Theta & \Phi\Theta & \Theta & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots \\ \Phi^n\Theta & \Phi^{n-1}\Theta & \Phi^{n-2}\Theta & \cdots & \Theta \end{bmatrix} \begin{bmatrix} u_k(t-1) \\ u_k(t) \\ u_k(t+1) \\ \vdots \\ u_k(t+n-1) \end{bmatrix} \qquad (4\text{-}18)$$

And from the equation (4-12) the output matrix can be expressed as:

$$\underbrace{\begin{bmatrix} y_k(t) \\ y_k(t+1) \\ y_k(t+2) \\ \vdots \\ y_k(t+n) \end{bmatrix}}_{Y_k} = \underbrace{\begin{bmatrix} C\Phi \\ C\Phi^2 \\ C\Phi^3 \\ \vdots \\ C\Phi^{n+1} \end{bmatrix}}_{d} x_{k-1}(t) + \underbrace{\begin{bmatrix} C\Theta & 0 & 0 & \cdots & 0 \\ C\Phi\Theta & C\Theta & 0 & \cdots & 0 \\ C\Phi^2\Theta & C\Phi\Theta & C\Theta & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C\Phi^n\Theta & C\Phi^{n-1}\Theta & C\Phi^{n-2}\Theta & \cdots & C\Theta \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} u_k(t-1) \\ u_k(t) \\ u_k(t+1) \\ \vdots \\ u_k(t+n-1) \end{bmatrix}}_{U_k} \qquad (4\text{-}19)$$

Which can be referred as the expression in (4-19):

$$Y_k = GU_k + dx_{k-1} \qquad (4\text{-}20)$$

Where G is a lower triangular matrix and its parameters are the Markov parameters of the dynamic system. The input vector to the control system can be founded by:

$$U_{k+1} = U_k + \Gamma e_k \qquad (4\text{-}21)$$

Where Γ represent ILC learning gain matrix, and e_k is the difference between the reference signal R_k and the output of the system Y_k from the equation (4-20):

$$e_k = R_k - Y_k \qquad (4\text{-}22)$$

By substituting equations (4-20) in the equation (4-22), and then plug the output in equation (4-21) we get the overall equation of the input vector to the ILC control system as shown below:

$$U_{k+1} = U_k + \Gamma(R_k - GU_k - dx_{k-1}) = f(U_k) \qquad (4\text{-}23)$$

The learning gain matrix Γ of ILC algorithm is responsible for ensuring the convergence, the equation (4-23) which will be converged when the maximum absolute eigenvalue of the learning gain matrix Γ be less than one $|\lambda|_{max} < 1$.

$$\|I - \Gamma G\| = \max_{1 \le t \le N_p} |\lambda_i| \tag{4-24}$$

When it been converged it will lead to $U_k = U_{k-1} = U_\infty$, so the error have to be zero $e_k = 0$. To ensure that the learning gain matrix will lead the term to approach to zero error.

The main eigenvalue's definition will be used to calculate the learning gain matrix Γ.

$$Av = \lambda v \tag{4-25}$$

Let's the matrix $A = (I - \Gamma G)$ so:

$$(I - \Gamma G)v = \lambda v \tag{4-26}$$

$$v - \Gamma G v = \lambda v \tag{4-27}$$

$$(\mathrm{I} - \lambda)v = \Gamma G v \tag{4-28}$$

$$\Gamma = (v - \lambda v)v^{-1}G^{-1} \tag{4-29}$$

As the eigenvector is an arbitrary vector, so for simplicity let $v = I$, and if the eigenvalue took as matrix Λ, so the equation (4-29) can be written as:

$$\Gamma = (I - \Lambda)G^{-1} \tag{4-30}$$

The equations (4-30) is applicable only if the lower diagonal matrix G is an invertible matrix, to find the inverse of the diagonal matrix G^{-1}, the below method will be followed. Consider G matrix in the system below:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} a & 0 & 0 \\ b & c & 0 \\ d & e & f \end{bmatrix}}_{G} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4-31}$$

Where, $a, b, c, d, e,$ and f are scalars, to inverse the system (4-31), it will be written as:

$$\left.\begin{aligned} X &= ax \\ Y &= bx + cy \\ Z &= dx + ey + fz \end{aligned}\right\} \Rightarrow$$

$$\begin{cases} x = \frac{1}{a}X \\ y = \frac{1}{c}\left(Y - \left(\frac{b}{a}X\right)\right) \\ z = \frac{1}{f}\left(Z - d\left(\frac{1}{a}X\right) - \left(\frac{e}{c}\left(Y - \left(\frac{b}{a}X\right)\right)\right)\right) \end{cases} \tag{4-32}$$

$$\begin{aligned} x &= \left(\frac{1}{a}\right)X \\ y &= -\left(\frac{b}{ac}\right)X + \left(\frac{1}{c}\right)Y \\ z &= \left(-\frac{d}{a} + \frac{eb}{ac}\right)X - \left(\frac{e}{c}\right)Y + \left(\frac{1}{f}\right)Z \end{aligned} \tag{4-33}$$

so the inverted system becomes:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{a} & 0 & 0 \\ \frac{-b}{ac} & \frac{1}{c} & 0 \\ \frac{-d}{a} + \frac{eb}{ac} & \frac{-e}{c} & \frac{1}{f} \end{bmatrix}}_{G^{-1}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4-34}$$

and the inverse of G is:

$$G^{-1} = \begin{bmatrix} \dfrac{1}{a} & 0 & 0 \\ \dfrac{-b}{ac} & \dfrac{1}{c} & 0 \\ \dfrac{-d}{a} + \dfrac{eb}{ac} & \dfrac{-e}{c} & \dfrac{1}{f} \end{bmatrix} \tag{4-35}$$

This method can be used to find the inverse matrices when the dynamic system is (SISO), but in (MIMO) dynamic systems (e.g. two inputs two outputs system) C and B vectors will be $[2 \times 2]$ matrices, then CB will not be a scalar and it will be $[2 \times 2]$ matrix, in this case, finding G matrix by using this method will be not an easy job as follow:

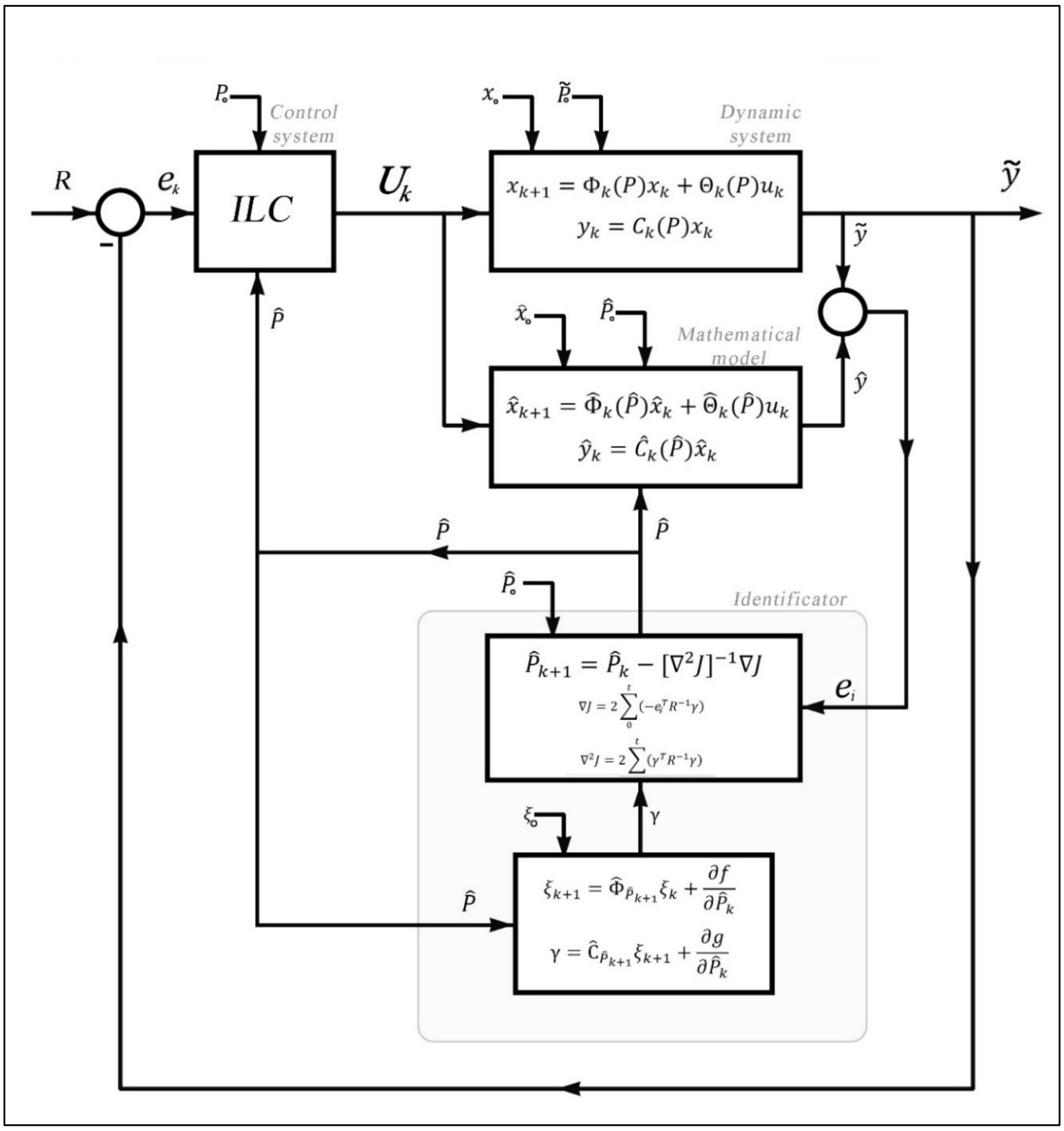
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} A_{22} & 0 & 0 \\ B_{22} & C_{22} & 0 \\ D_{22} & E_{22} & F_{22} \end{bmatrix}}_{G} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4-36}$$

Where, $A_{22}, B_{22}, C_{22}, D_{22}, E_{22}, and\ F_{22}$ are scalars, to inverse the system (4-31), by following the previous steps, the matrix will be:

$$G^{-1} = \begin{bmatrix} A_{22}^{-1} & 0_{22} & 0_{22} \\ -B_{22}A_{22}^{-1}C_{22}^{-1} & C_{22}^{-1} & 0_{22} \\ -D_{22}A_{22}^{-1} + E_{22}B_{22}A_{22}^{-1}C_{22}^{-1} & -E_{22}C_{22}^{-1} & F_{22}^{-1} \end{bmatrix} \tag{4-37}$$

4.4 Adaptive Iterative Learning Control (AILC) Algorithm

Adaptive iterative learning control (AILC) algorithm considered by using Arimoto's ILC standard algorithm (2-6) and the error output method of the least squares approximation (3-76), the adaptive iterative learning control (AILC) algorithm is explained in Figure 4.4.



R
e_k
P_o
Control system
ILC
U_k
x_o
$\tilde{P}_o$
Dynamic system
$x_{k+1} = \Phi_k(P)x_k + \Theta_k(P)u_k$
$y_k = C_k(P)x_k$
$\tilde{y}$
$\tilde{y}$
-
$\hat{P}$
$\hat{x}_o$
$\hat{P}_o$
Mathematical model
$\hat{x}_{k+1} = \hat{\Phi}_k(\hat{P})\hat{x}_k + \hat{\Theta}_k(\hat{P})u_k$
$\hat{y}_k = \hat{C}_k(\hat{P})\hat{x}_k$
$\hat{y}$
$\hat{P}$
$\hat{P}$
$\hat{P}_o$
Identificator
$\hat{P}_{k+1} = \hat{P}_k - [\nabla^2 J]^{-1}\nabla J$
$\nabla J = 2\sum_0^t(-e_i^T R^{-1}\gamma)$
$\nabla^2 J = 2\sum_0^t(\gamma^T R^{-1}\gamma)$
e_i
ξ_o
γ
$\hat{P}$
$\xi_{k+1} = \hat{\Phi}_{\hat{P}_{k+1}}\xi_k + \frac{\partial f}{\partial \hat{P}_k}$
$\gamma = \hat{C}_{\hat{P}_{k+1}}\xi_{k+1} + \frac{\partial g}{\partial \hat{P}_k}$

Figure 4.4: The Developed (AILC) Algorithm.

The main dynamic system with given initial values of unknown parameters P is:

$$x_{k+1} = \Phi_k(P)x_k + \Theta_k(P)u_k \tag{4-38}$$

$$y_k = C_k(P)x_k \tag{4-39}$$

The mathematical model with values of identified parameters $\hat{P}$ is:

$$\hat{x}_{k+1} = \hat{\Phi}_k(\hat{P})\hat{x}_k + \hat{\Theta}_k(\hat{P})u_k \tag{4-40}$$

$$\hat{y}_k = \hat{C}_k(\hat{P})\hat{x}_k \tag{4-41}$$

The input signal u_k for the dynamic systems (4-38) and the mathematical model (4-40) is calculated by the ILC algorithm (4-23), with the identified parameters $\hat{P}$, which is calculated by the method of error output method of least squares approximation. By using (3-23) and (3-24), the necessary and sufficient conditions of the identification process take the forms:

Necessary condition

$$\nabla J = 2\sum_{0}^{t}(-e_i{}^T R^{-1}\gamma) \tag{4-42}$$

Sufficient condition

$$\nabla^2 J = 2\sum_{0}^{t}(\gamma^T R^{-1}\gamma) \tag{4-43}$$

Where ∇J Jacobian matrix and $\nabla^2 J$ is the Hessian matrix, and e is the error between the output of the main dynamic system y and the output mathematical model $\hat{y}$.

$$e_i = y - \hat{y} \tag{4-44}$$

And γ can be calculated from the system below:

$$\xi_{k+1} = \hat{\Phi}_{\hat{P}_{k+1}}\xi_k + \frac{\partial f}{\partial \hat{P}_k} \tag{4-45}$$

$$\gamma = \hat{C}_{\hat{P}_{k+1}}\xi_{k+1} + \frac{\partial g}{\partial \hat{P}_k} \tag{4-46}$$

Where ξ is the sensitivity of the state which can be implemented by $\xi_{ij} = dx_i/d\hat{P}_j$, and γ is the sensitivity of output which can be implemented by $\gamma_{ij} = dy_i/d\hat{P}_j$.

The parameter identification algorithm can be calculated as shown below in equation (4-47), which is found by using Newton-Raphson optimization algorithm and applying it to the necessary conditions ∇J and sufficient conditions $\nabla^2 J$. The equation below is mainly based on the equation (3-76).

$$\hat{P}_{k+1} = \hat{P}_k - [\nabla^2 J]^{-1} \nabla J \tag{4-47}$$

Using Newton-Raphson method has a very good advantage in solving estimation problems because this method is a very fast method, that will make the implementation of the new estimated parameters become very fast and the system will be applicable for fast dynamic systems.

CHAPTER FIVE

SIMULATION AND RESULTS

The application of the adaptive iterative learning control (AILC) algorithm on the LTI dynamic system that expressed in chapter four will be simulated in this chapter by using MATLAB. The system will be taken in simulation is in discrete-time state-space system shown in the expression (4-5), and its output is shown in (4-6), the discrete-time form of the matrix *A* and the vector *B* could be implemented by using the series expressed in (4-15) and (4-16). The discretized matrices Φ and Θ will be used in the discrete-time form of the dynamic system shown in the expressions (4-9) and (4-10). After the determining the discrete-time form of the dynamic system, the control algorithm will be applicable to it.

5.1 Reference Signals

Two kinds of reference signals will be used in this simulation, first one is a combination of sine and cosine waves as illustrated in the expression (5-1). And the second one is a rough square wave which has configured by using Fourier series as expressed in (5-2).

5.1.1 First reference signal RC_1

The first reference signal which is used in this simulation is RC_1, its equation is shown in (5-1), and its shape is illustrated in Figure 5.1.

$$RC_1 = \sin\big(2\pi(t/T)\big) + 0.04(\cos\big(0.4\pi(t/T + 4)\big) + 0.2\sin\big(2\pi(t/T + 5)\big) \qquad (5\text{-}1)$$

where t is time duration of the simulation, the symbol T is a weight factor to control the frequency of signal, which is taken as 0.4 in this simulation. The initial values of the dynamic system's state vector $x_{k-1} = [x_i \quad v_i]^T$ are taken as $x_i = 2$ and $v_i = 2$ to start far away from the starting point of the reference signal $RC1$ which is (-0.4532) in order to see the response clearly in spite of that will make a jump in the input signal.

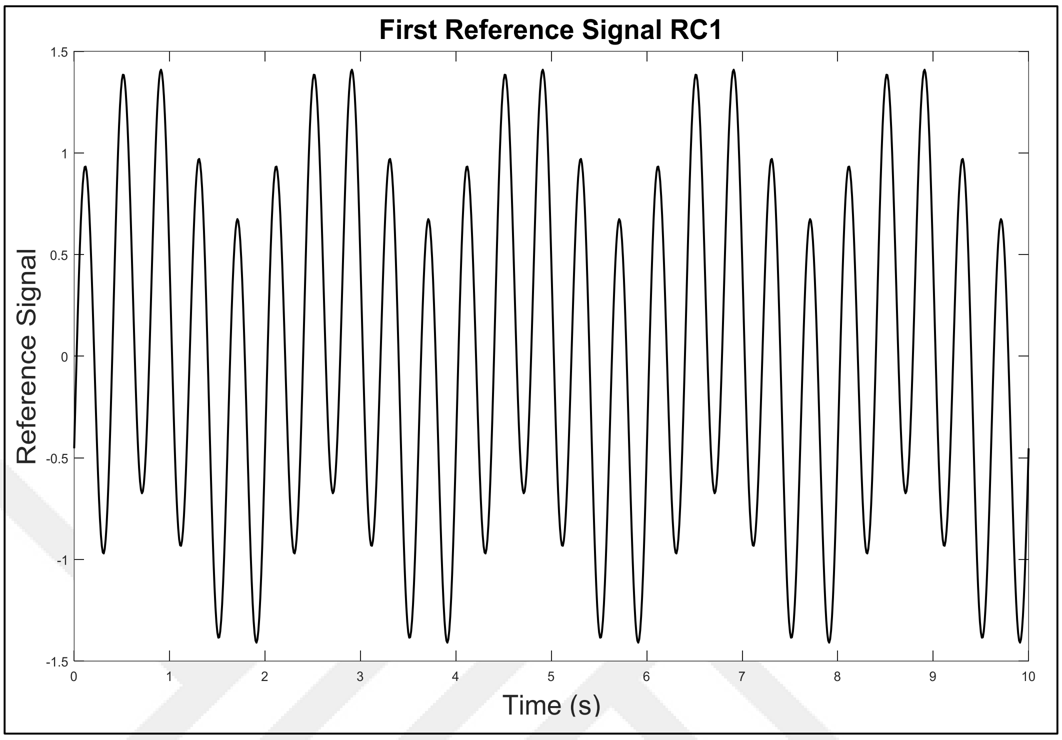


Figure 5.1: First Reference Signal (RC_1).

5.1.2 Second reference signal RC_2

The second reference signal $RC2$ that will be used in this simulation is a rough square wave configured in the equation (5-2), which implemented by applying Fourier series on sine wave to get the shape shown in Figure 5.2.

$$RC_2 = \sum_{i=1}^{j} \frac{\sin(4.1i)t}{(i)!} \qquad (5\text{-}2)$$

Where $j = 1, 3, 5, \ldots, 19$, t is the time $0 < t < N$, and $N = 10\ sec.$, and the initial values of the dynamic system's state vector $x_{k-1} = [x_i \quad v_i]^T$ will be taken as in the first reference signal $x_i = 2$, and $v_i = 2$ to start far away from the starting point of the $RC2$ which is (0).

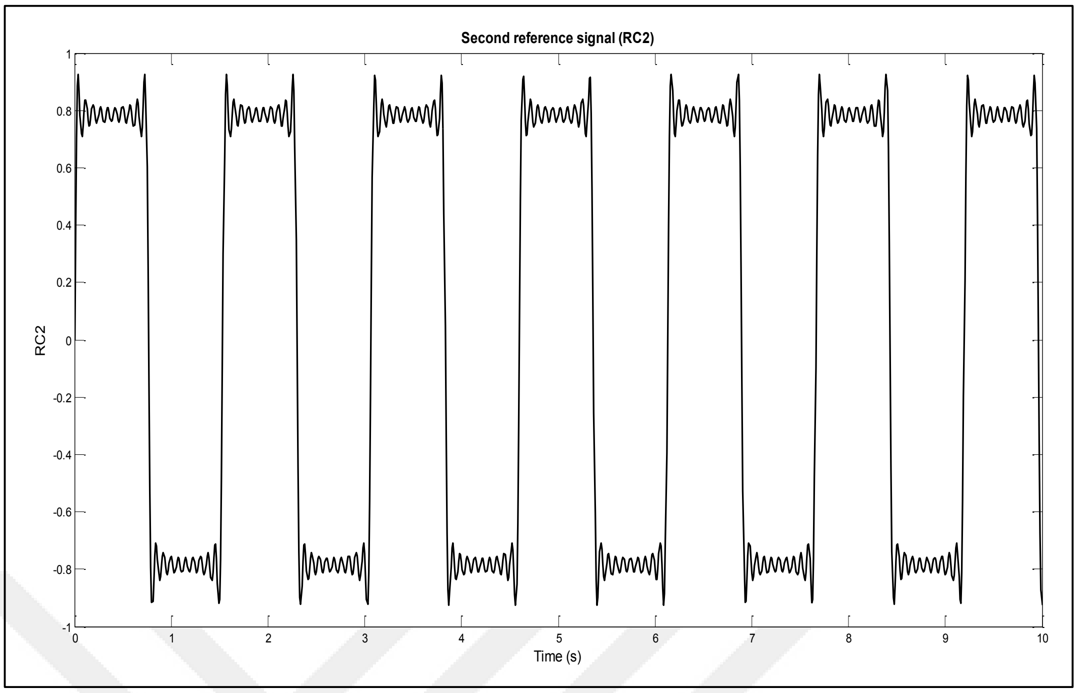


Figure 5.2: Second Reference Signal (RC_2).

5.2 Simulation of ILC Algorithm

In this part, Arimoto's ILC algorithm (4-23) will be applied to the illustrated dynamic system and the output vector Y_k expressed in equation (4-20) will be taken into consideration. The learning gain matrix Γ will be implemented by using the equation (4-30), and it will be surely exist if $CB > 0$, eigenvalues λ of the learning gain matrix will be used to control the speed of the ILC control system.

The actual parameters of the dynamic system will be taken as $\zeta = 0.1$, and $\omega_n = 0.2$, in this part, the actual parameters P and the identified parameters $\hat{P}$ will be the same values $P = \hat{P} = [\zeta \quad \omega_n]^T = [0.1 \quad 0.2]^T$, that means the dynamic system has no unknown parameters and the control system has no parameter identificator.

The goal of this part of the simulation is showing the behavior of the pure ILC controller with the LTI dynamic system and how to track the reference signals RC_1 and RC_2 before applying the unknown parameters on the dynamic system. The supposed results of this part is, ILC should track the reference signals at several situations and shows different results for that, and also it should fail to track the same signals at the same situations when applying the unknown parameters on the dynamic system. In the next part, AILC algorithm should be successful to achieve what ILC failed to do, and also gives close results to that results of pure ILC when no unknown parameters.

Figure 5.3 shows the results when ILC tracking the reference signal RC_1 at eigenvalue $\lambda = 0.99$ and iteration number $k = 5$, and Figure 5.4 shows the results when ILC tracking the reference signal RC_2 at the same values of eigenvalue and iteration number.

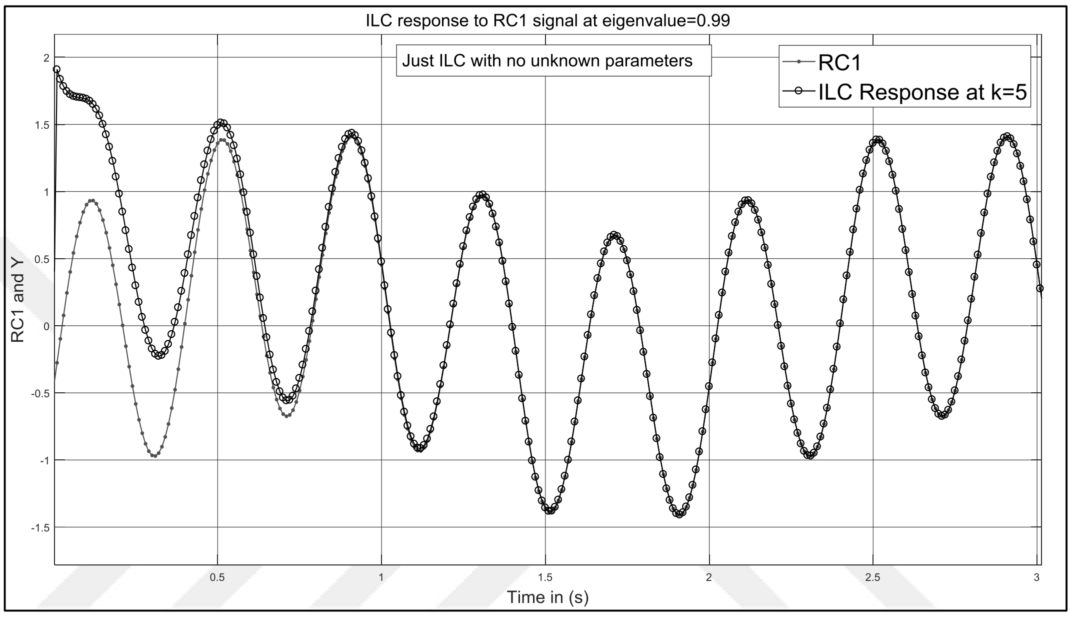


Figure 5.3: ILC Response to RC_1at $\lambda = 0.99$ and $k = 5$.

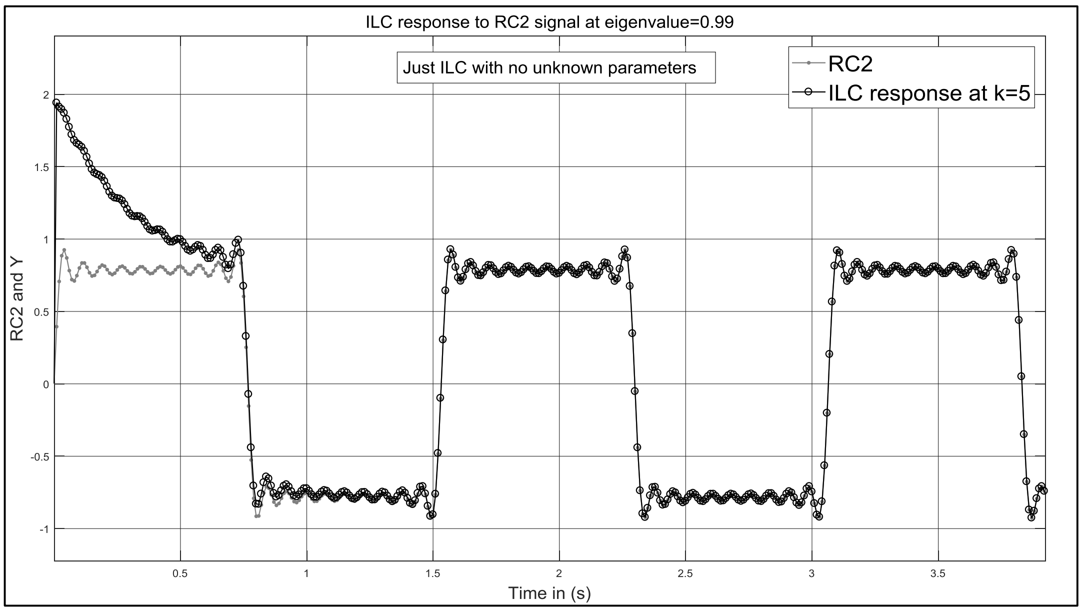


Figure 5.4: ILC Response to RC_2 at $\lambda = 0.99$ and $k = 5$.

Figure 5.5 and Figure 5.6 show the converging of ILC algorithm to RC_1 and RC_2 when eigenvalue of the learning gain matrix $\lambda = 0.99$, and also show tracking progress for many iteration numbers k. The convergence speed of the control system also affect the response of the dynamic system which can be changed by with change of eigenvalue λ of the learning gain matrix Γ, as the eigenvalue λ approach to 1, as ILC controller needs more iterations k to reach zero error.

To show the effect of the control system's convergence speed on the dynamic system's reference, ILC algorithm have applied on the dynamic system with constant iteration number $k = 5$, with different eigenvalues $\lambda = 0.99$, 0.95 and 0.90, the results of ILC controller to track RC_1 and RC_2 at the mentioned situations are illustrated on Figure 5.7 and Figure 5.8.

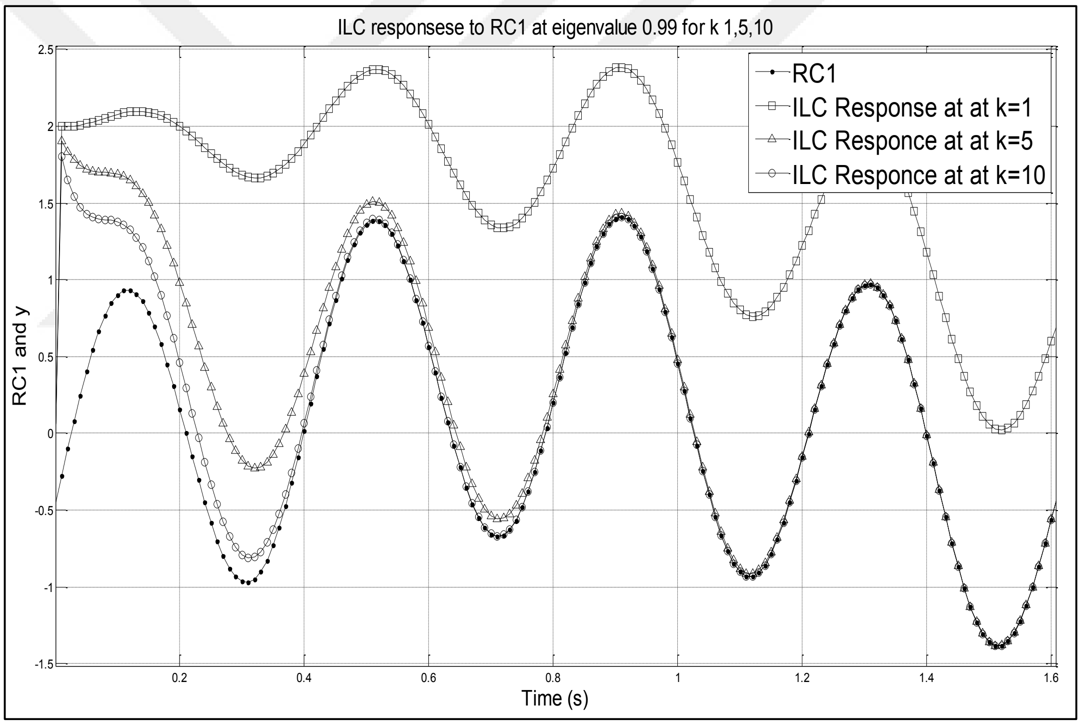


Figure 5.5: ILC Response to RC_1 Signal at $\lambda = 0.99$ (Iteration Convergence).

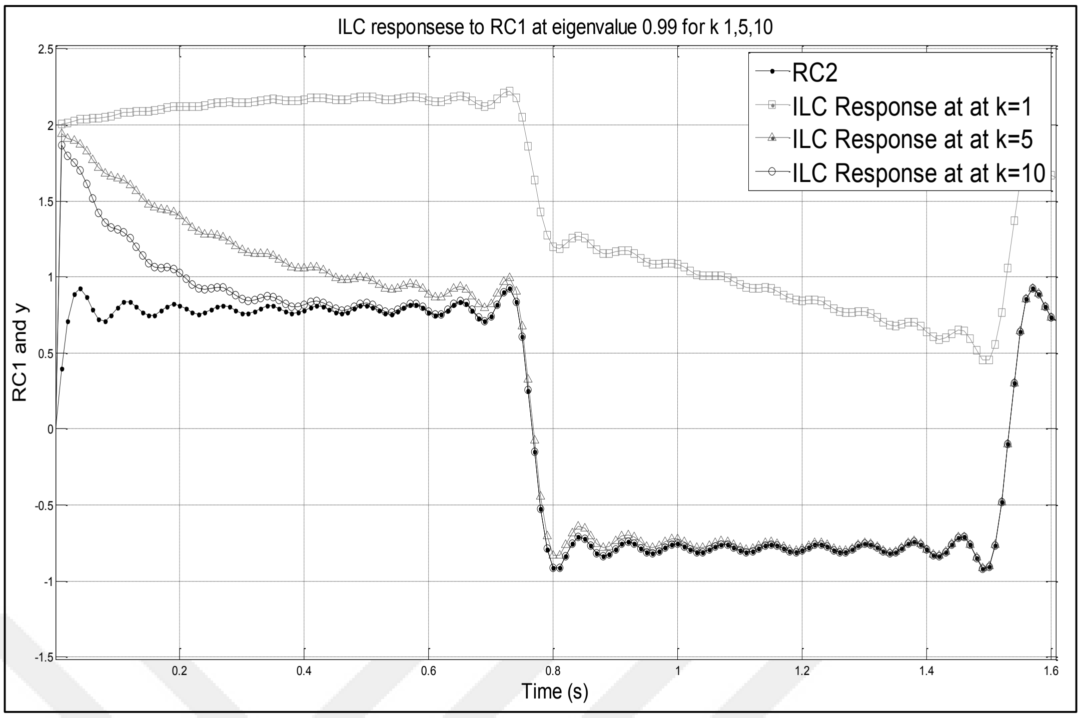


Figure 5.6: ILC Response to RC_2 Signal at $\lambda = 0.99$ (Iteration Convergence).

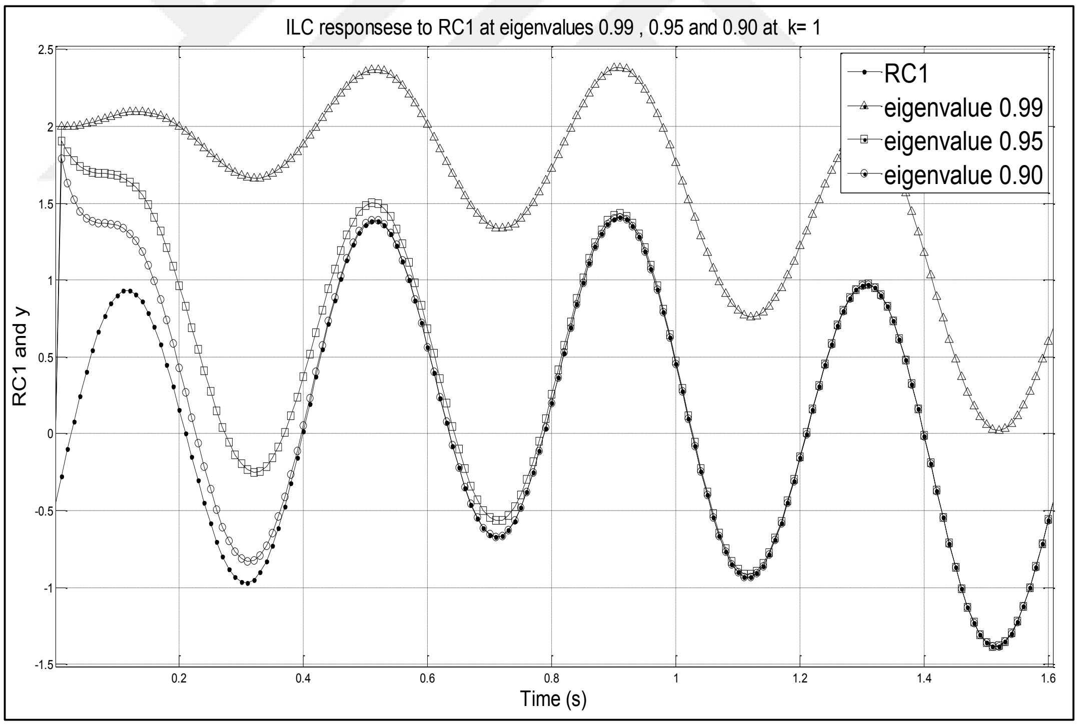


Figure 5.7: ILC Response Sensitivity to RC_1 Signal at $\lambda = 0.99, 0.95, 0.90$, k=1.

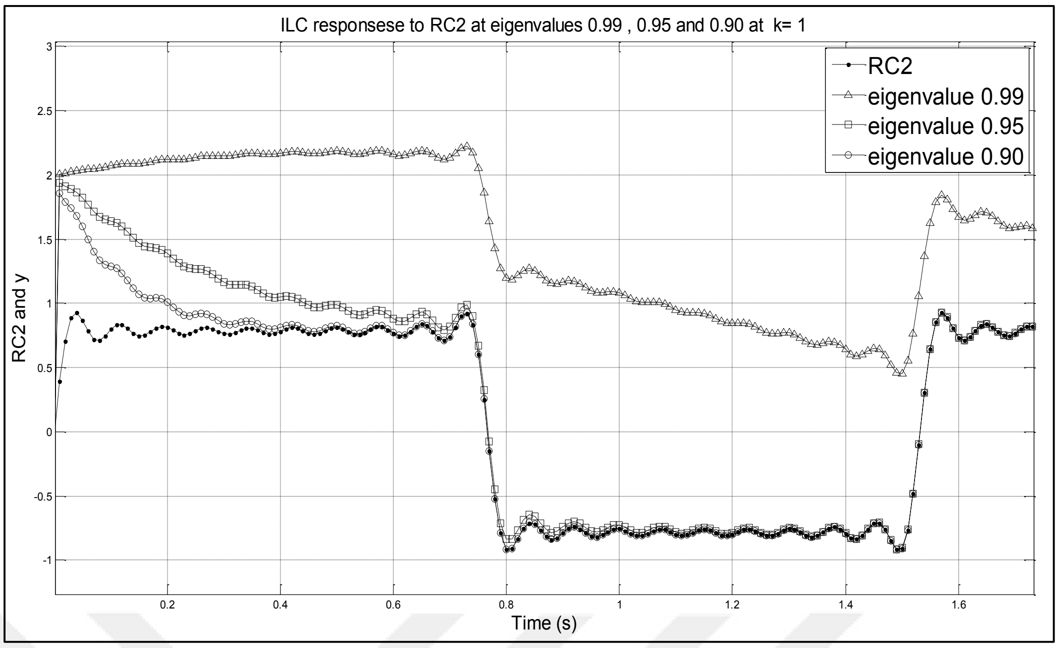


Figure 5.8: ILC Response Sensitivity to RC_2 Signal at $\lambda = 0.99, 0.95, 0.90$, k=1.

As ILC controller will fail to control the dynamic system to track the reference in case of any change in one or more of its parameters, so it needs to an identificator to calculate the optimum parameters to adapt its output with the new changes, and that what it took into consideration in AILC algorithm, the difference between ILC and AILC has illustrated in Figure 5.9 and Figure 5.10 under high initial parameters to show big error and to show the robustness of AILC.

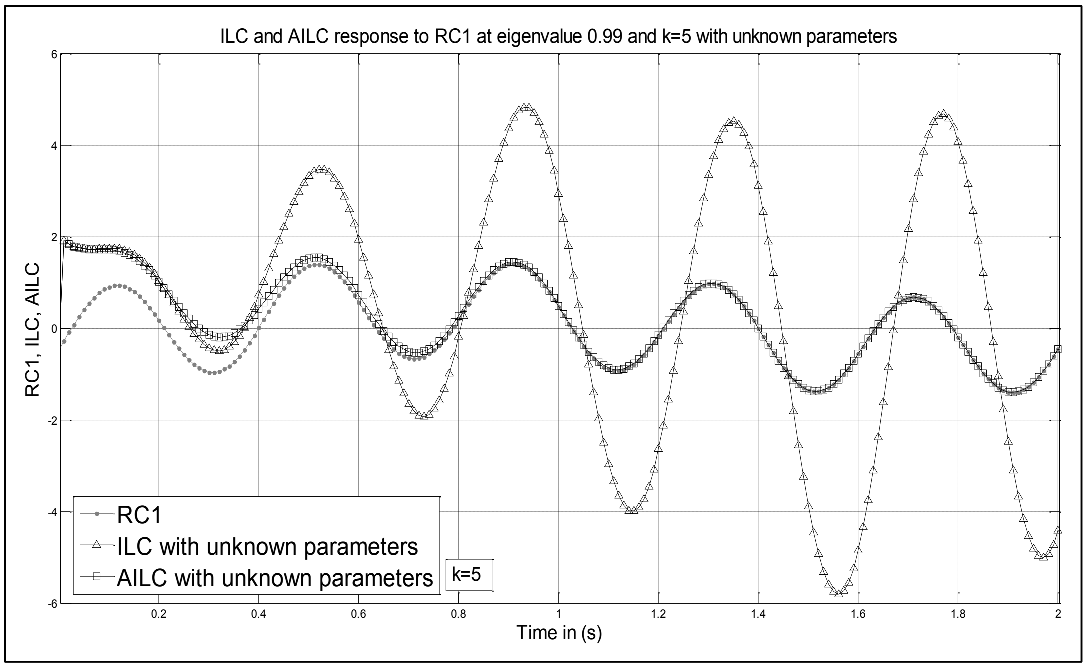


Figure 5.9: ILC vs AILC for RC_1at $\lambda = 0.99$, k=5 and $\hat{P} = [5; 5]^T$.

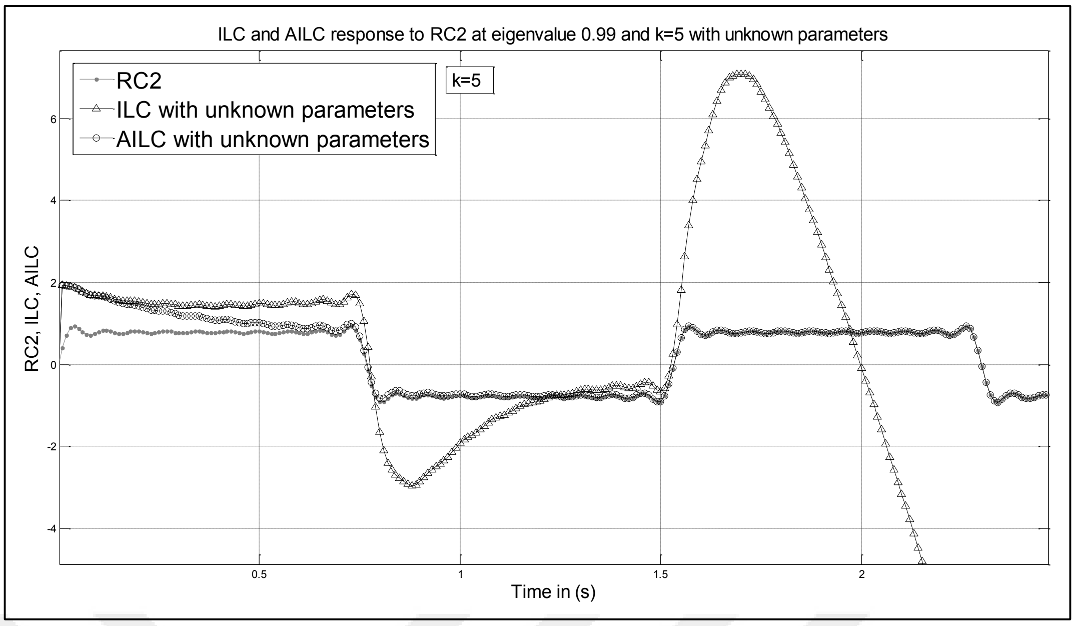


Figure 5.10: ILC vs AILC for RC_2at $\lambda = 0.99$, k=5 and $\hat{P} = [5; 5]^T$

After these results, ILC behavior to reference signals RC_1 and RC_2 becomes clear, and the comparison between ILC and AILC becomes possible now. After adding the unknown parameters to the dynamic system, AILC should behave close to the ILC's behavior when control the same dynamic system but without unknown parameters.

5.3 Simulation of Adaptive Iterative Learning Control (AILC) Algorithm

The same algorithm of Arimoto (4-23) will be applied on this part with taking ζ and ω_n as unknown parameters, where P_i denotes the initial values of original parameters $P_i = [\zeta \quad \omega_n]^T = [0.1 \quad 0.2]^T$, and $\hat{P}_i$ denotes the values of the identified parameters $[\hat{\zeta} \quad \hat{\omega}_n]^T$, lets the initial values of the identified parameters be $\hat{P}_i = [0.08 \quad 0.22]^T$. The initial values of the system's state will be taken as in (ILC) simulation $x_i = 2$ and $v_i = 2$. By considering Figure 4.4, and the adaptive iterative learning control (AILC) algorithm expressed in the part (4.4 Adaptive Iterative Learning Control (AILC) Algorithm). The expressed initial values of the actual parameters P will be applied on the discrete-time model of the dynamic system expressed in (4-38), and the initial identified parameters $\hat{P}$ will be applied on the mathematical model expressed in (4-40). By considering Figure 4.4, the error e_i will be calculated by taking the difference between the output of the dynamic system y and the output of the mathematical model $\hat{y}$ of the parameter identificator as expressed in equation (4-44), this error value is used to calculate the next identified parameters $\hat{P}$ which will represent the new parameters entering to the ILC controller and the mathematical model after been calculated by the equation (4-47), the necessary and sufficient conditions $\nabla^2 J$ and ∇J respectively will be implemented by equations (4-42) and (4-43), and sensitivity of output γ will be calculated by equation (4-45) after finding the sensitivity of state ξ from equation (4-46).

The sensitivity of state: $\xi_{k+1} = \widehat{\Phi}_{\hat{P}_{k+1}} \xi_k + \frac{\partial f}{\partial \hat{P}_k}$

The sensitivity of output: $\gamma = \widehat{C}_{\hat{P}_{k+1}} \xi_{k+1} + \frac{\partial g}{\partial \hat{P}_k}$

The discretized dynamic system (4-40) consists of functions naturally, so it can be described as $f(\hat{P}) = \hat{x}_{k+1} = \widehat{\Phi}_k(\hat{P})\hat{x}_k + \widehat{\Theta}_k(\hat{P})u_k$, and because we have two states and two unknown parameters, the partial derivative of the functions with respect to unknown parameters will be $[4 \times 4]$ matrix.

For the two states in the dynamic system, $f(\hat{P}) = \begin{bmatrix} f_1(\hat{P}) \\ f_2(\hat{P}) \end{bmatrix}$ explained in details below:

$$\begin{bmatrix} f_1(\hat{P}) \\ f_2(\hat{P}) \end{bmatrix} = \begin{bmatrix} x_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k \qquad (5\text{-}3)$$

The partial derivative of f with respect to unknown parameters is $\frac{\partial f}{\partial \hat{P}} = \begin{bmatrix} \frac{\partial f_1}{\partial \hat{P}} & \frac{\partial f_2}{\partial \hat{P}} \end{bmatrix}^T$, since there are two unknown parameters $\hat{P} = [\hat{\zeta} \quad \widehat{\omega}_n]^T$, so the complete matrix will be:

$$\begin{bmatrix} \frac{\partial f_1}{\partial \hat{P}} \\ \frac{\partial f_2}{\partial \hat{P}} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial \hat{\zeta}} & \frac{\partial f_1}{\partial \widehat{\omega}_n} \\ \frac{\partial f_2}{\partial \hat{\zeta}} & \frac{\partial f_2}{\partial \widehat{\omega}_n} \end{bmatrix} \qquad (5\text{-}4)$$

The size of equations inside the matrix (5-4) directly depends on the number of terms that have applied in discretization process of the matrices Φ and Θ in (4-15) and (4-16), the higher the number of discretization terms, the larger the size of the functions. Increasing of discretization terms has an effective benefit, where it increases the accuracy of the results and reduce the errors, but taking high number of terms will complicate the code as well as consume more time in simulation, because of that, just two terms have taken in this simulation, and just the result of one term have written in (5-5):

$$f_1 = x + vdt$$
$$f_2 = -v\left(2\widehat{\omega}_n\hat{\zeta}dt - 1\right) - \widehat{\omega}_n^2 xdt \tag{5-5}$$

The partial derivative of each function with respect to each of the unknown parameters in $\hat{P}$ is shown as follow:

$$\begin{bmatrix} \frac{\partial f_1}{\partial \hat{P}} \\ \frac{\partial f_2}{\partial \hat{P}} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -2v\widehat{\omega}_n dt & -2\widehat{\omega}_n xdt - 2v\hat{\zeta}dt \end{bmatrix} \tag{5-6}$$

The next system shows the formulas of the matrix above in case of more discretization terms:

$$\begin{bmatrix} \frac{\partial f_1}{\partial \hat{P}} \\ \frac{\partial f_2}{\partial \hat{P}} \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{k_i} \frac{\Delta t^{k_i}}{k_i!}\left(\frac{\partial f_1}{\partial \hat{\zeta}}\right)^{k_i} & \sum_{i=0}^{k_i} \frac{\Delta t^{k_i}}{k_i!}\left(\frac{\partial f_1}{\partial \widehat{\omega}_n}\right)^{k_i} \\ \sum_{i=0}^{k_i} \frac{\Delta t^{k_i}}{k_i!}\left(\frac{\partial f_2}{\partial \hat{\zeta}}\right)^{k_i} & \sum_{i=0}^{k_i} \frac{\Delta t^{k_i}}{k_i!}\left(\frac{\partial f_2}{\partial \widehat{\omega}_n}\right)^{k_i} \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} \tag{5-7}$$

Where k_i is discretization term, and k refers to ILC iteration number.

After implementing the new update of the unknown parameters $\hat{P}_{k+1}$, it will be sent to the ILC controller in order to adapt the input U_{k+1} according to the new identified parameters, which will affect the matrix G due to the change that happened in matrices Φ and Θ, the discretized forms of A and B matrices of the continuous-time form of the system. The new identified parameters $\hat{P}$ will reduce the error e_i that calculated by equation (4-44).

5.3.1 AILC simulation for RC_1 at $\lambda = 0.99$

In this part, AILC algorithm behavior with reference signal RC_1 will be shown when eigenvalue $\lambda = 0.99$, at this eigenvalue the convergence speed will be slow so the control system will need more iterations to reach zero error. The initial identified parameters $\hat{P}_i$ in this part of simulation and the next parts will be taken as $\hat{P}_i = [0.08 \quad 0.22]^T$. The input signal graph of this part is shown in Figure 5.11, and the response of the dynamic system is shown in Figure 5.12, the convergence steps of response shown in Figure 5.13. The graph of distance error e_d is illustrated in Figure 5.14, the identified parameters $\hat{P}$ is shown in Figure 5.15, the error of the unknown parameter identificator is shown in Figure 5.16, the velocity of the dynamic system to the velocity of the reference signal RC_1 is shown in Figure 5.17, and the error of the two velocities e_v is shown in Figure 5.18.

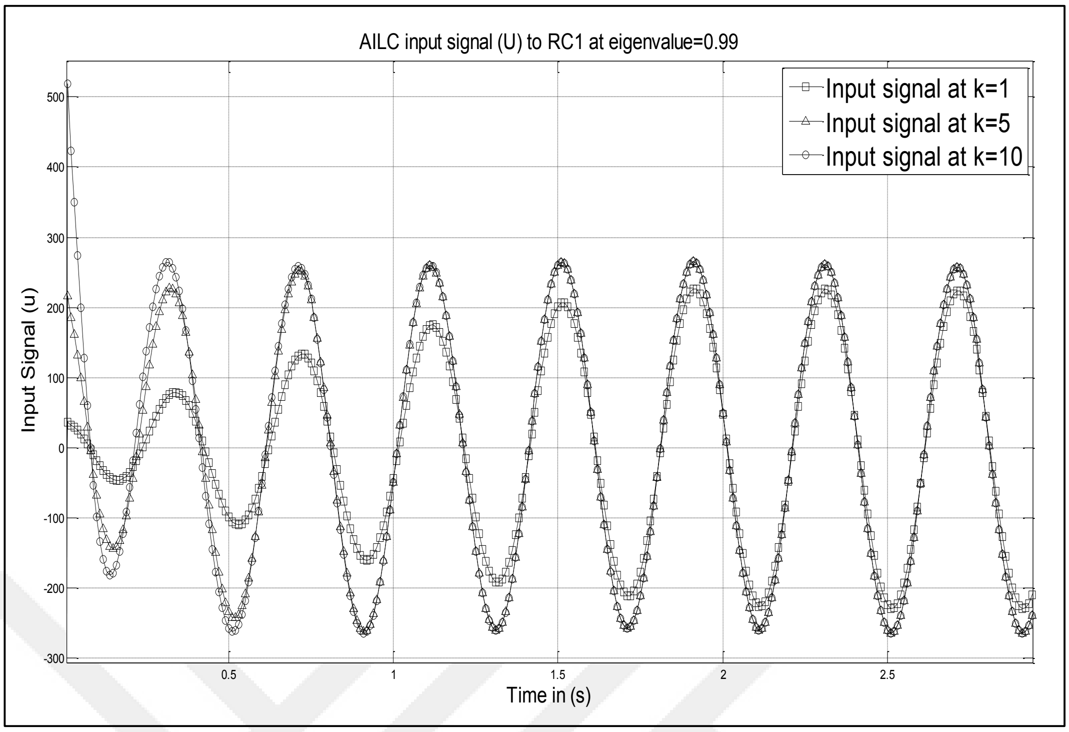


Figure 5.11: AILC Input Signal (U) for RC_1 Signal at $\lambda = 0.99$.

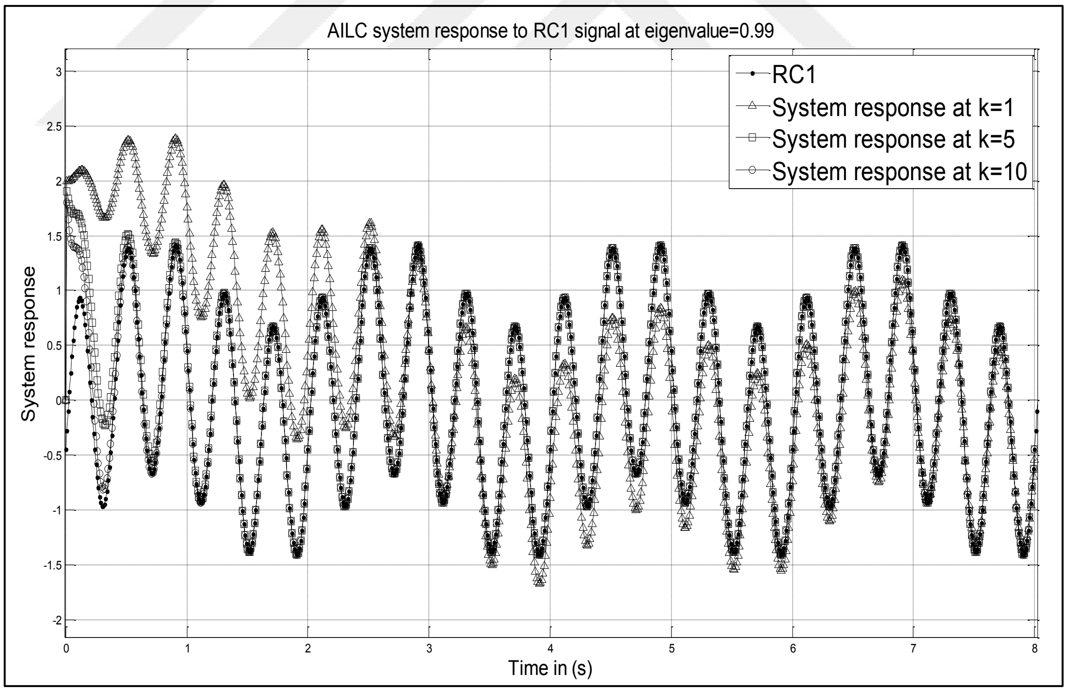


Figure 5.12: AILC Response Sensitivity to RC_1 Signal at $\lambda = 0.99$.

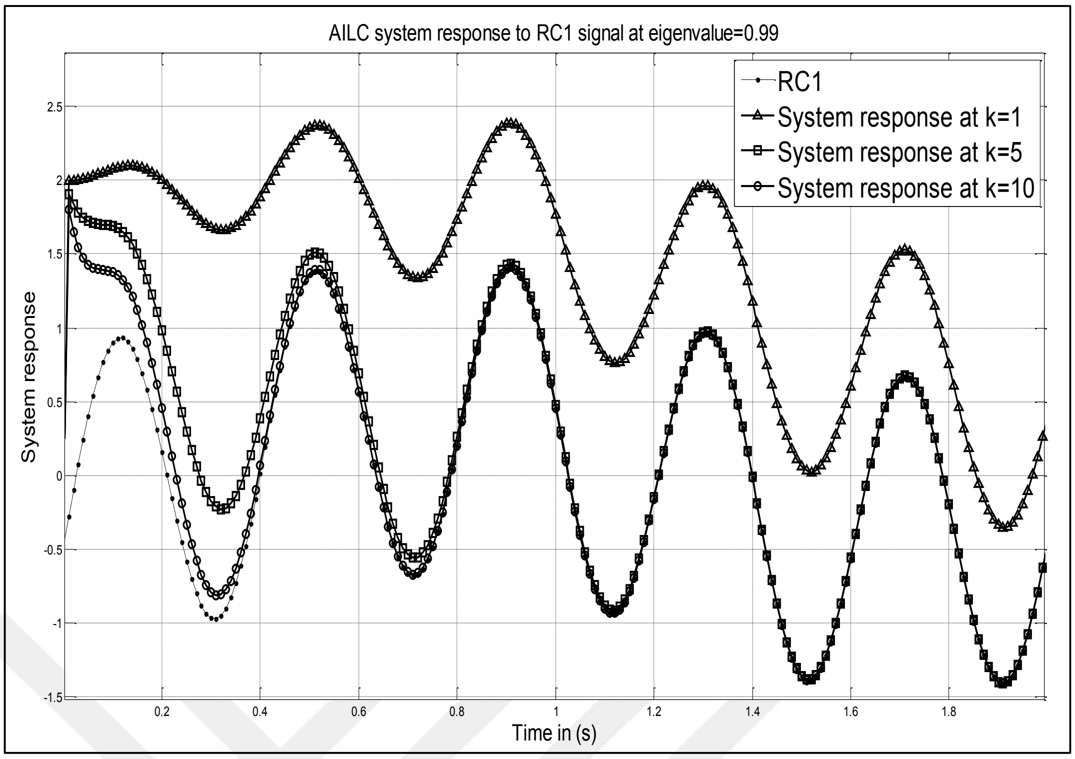


Figure 5.13: AILC Response to RC_1 Signal at $\lambda = 0.99$ (zoomed).

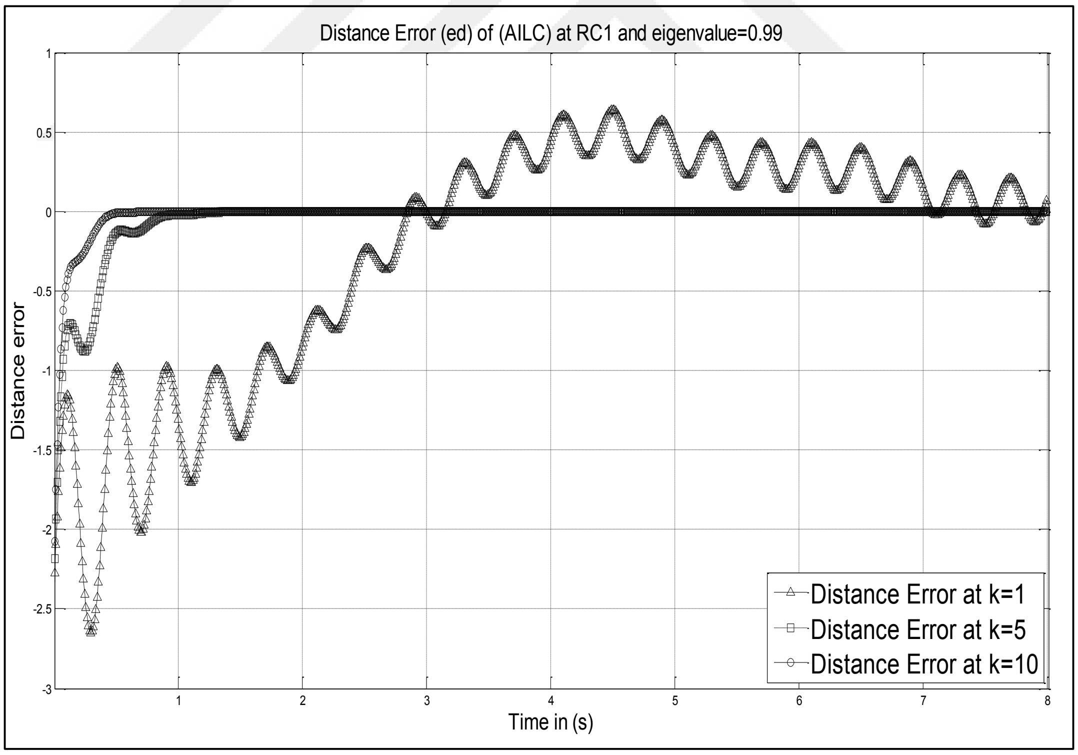


Figure 5.14: Distance Error e_d for RC_1 at $\lambda = 0.99$.

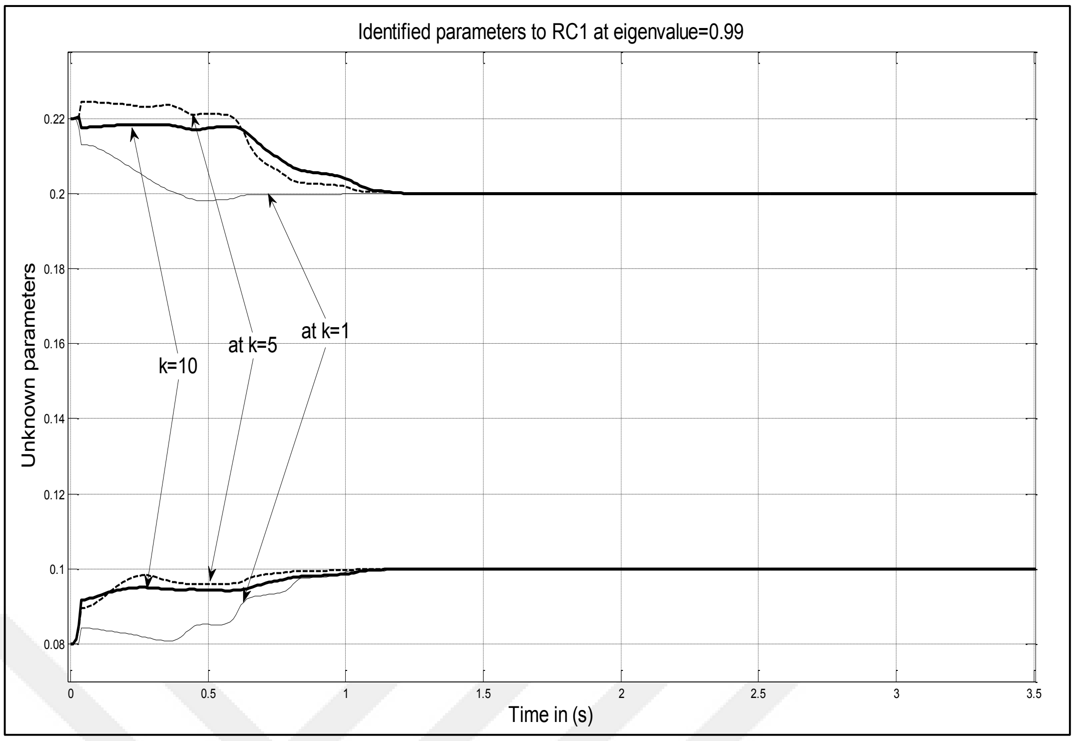


Figure 5.15: Identified Parameters $\hat{P}$ to RC_1 at $\lambda = 0.99$.

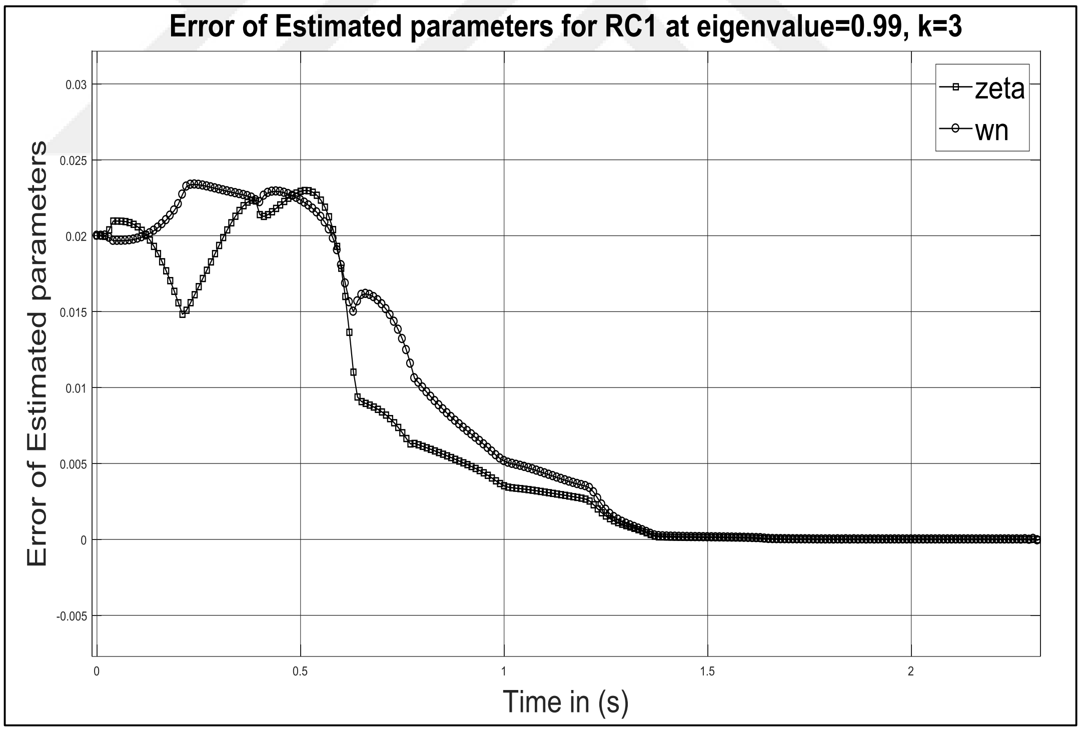


Figure 5.16: The Error e_i Between y and $\hat{y}$ to RC_1 at $\lambda = 0.99$.

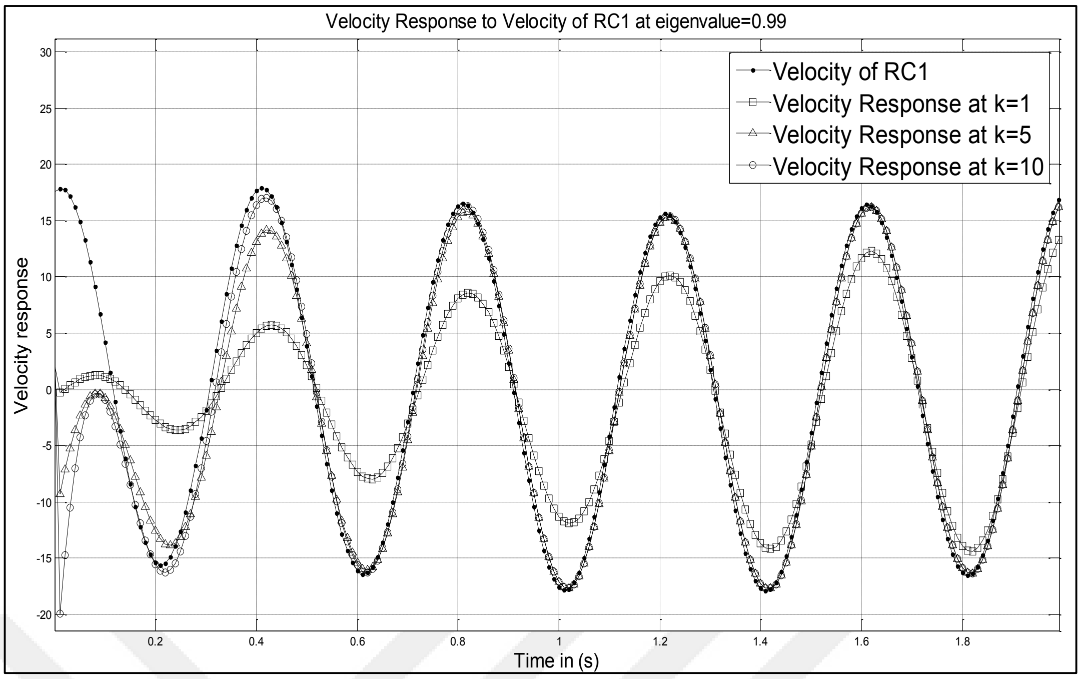


Figure 5.17: The Velocity (v) Response to Velocity of RC_1 at $\lambda = 0.99$.

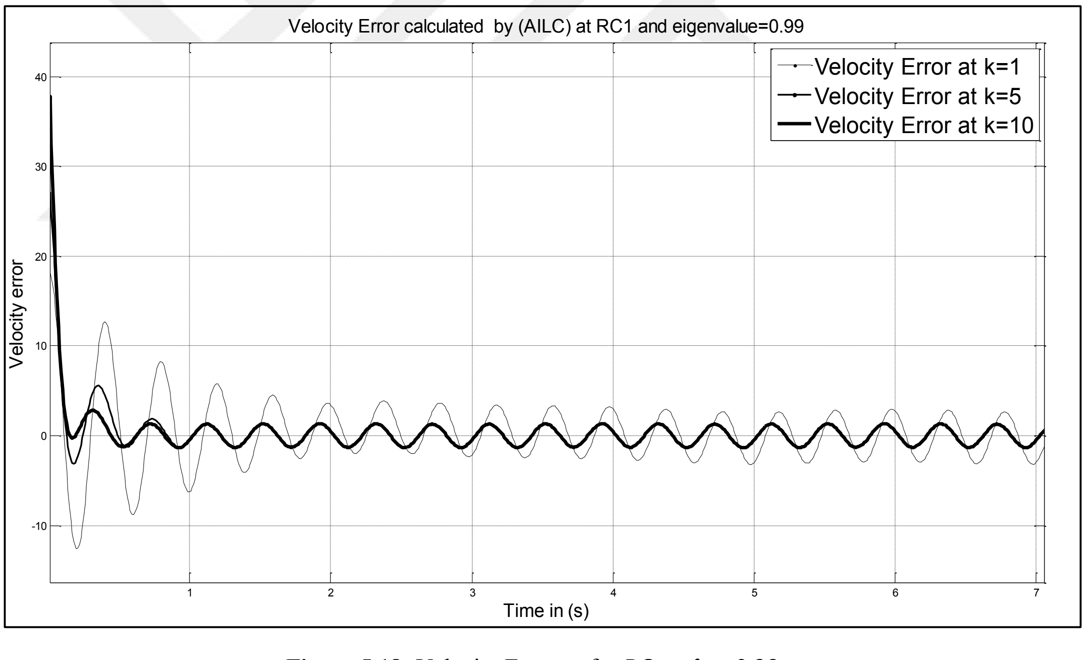


Figure 5.18: Velocity Error e_v for RC_1 at $\lambda = 0.99$.

5.3.2 AILC simulation for RC_1 at $\lambda = 0.95$

In this part, AILC will be applied on the reference signal RC_1 also, but the eigenvalue of the learning gain matrix will be taken as $\lambda = 0.95$ in order to increase the speed of convergence, therefore, will not need iteration number as large as when $\lambda = 0.99$ The input signal graph of this part is shown in Figure 5.19, and the response shown in Figure 5.20 and Figure 5.21, the distance error e_d is illustrated in

Figure 5.22, the identified parameters $\hat{P}$ is shown in Figure 5.23, the error of the unknown parameter identificator is shown in Figure 5.24, the velocity of the dynamic system to the velocity of the reference signal $RC1$ is Figure 5.25, and the error of the two velocities e_v is shown in Figure 5.26.

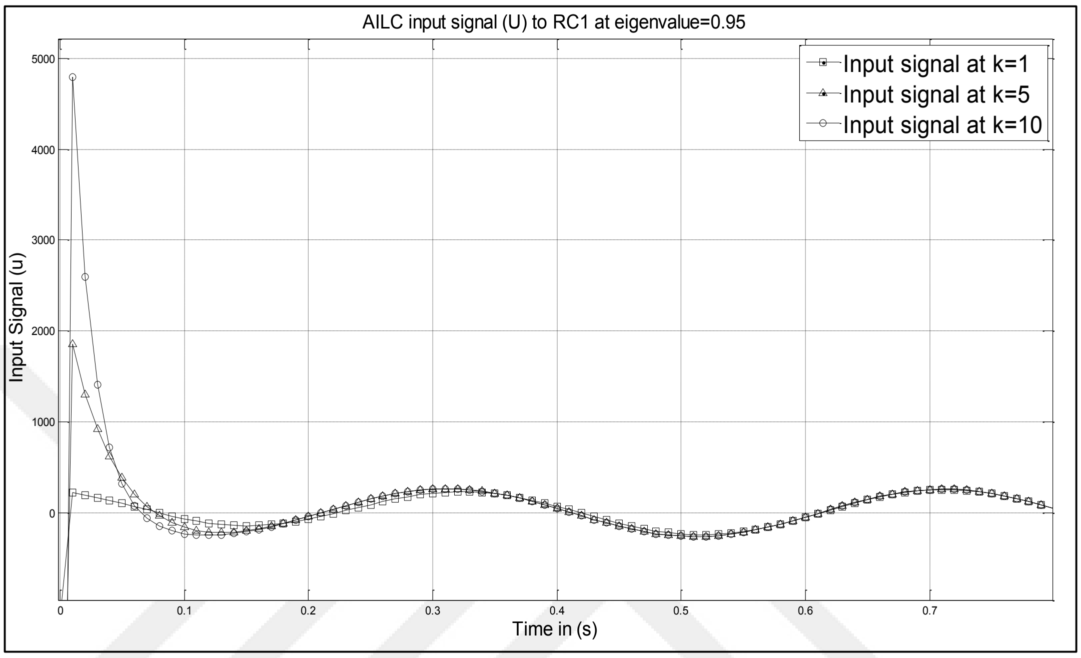


Figure 5.19: AILC Input Signal (U) for RC_1 Signal at $\lambda = 0.95$.

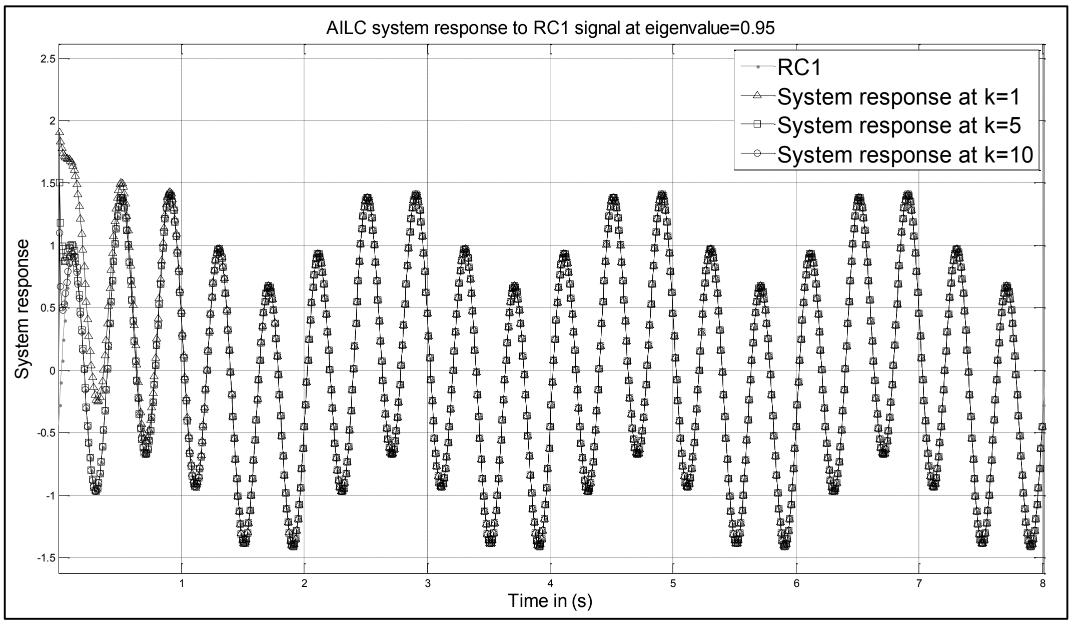


Figure 5.20: AILC Response to RC_1 Signal at $\lambda = 0.95$.

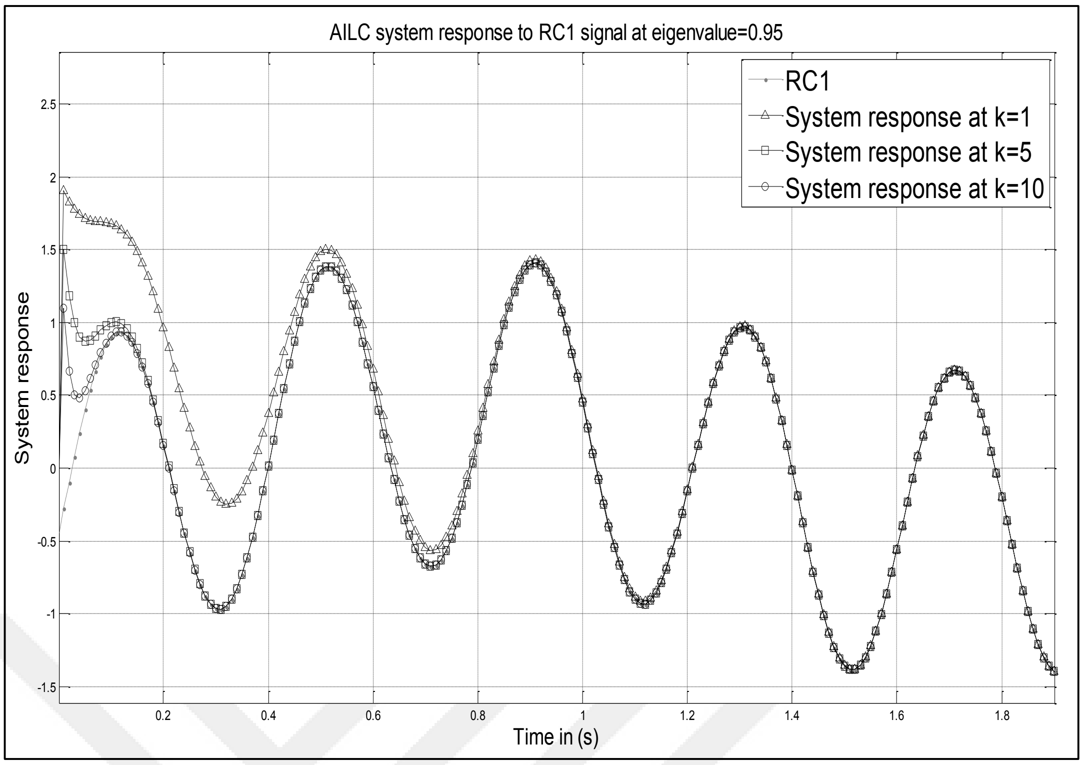


Figure 5.21: AILC Response to RC_1 Signal at $\lambda = 0.95$ (zoomed).

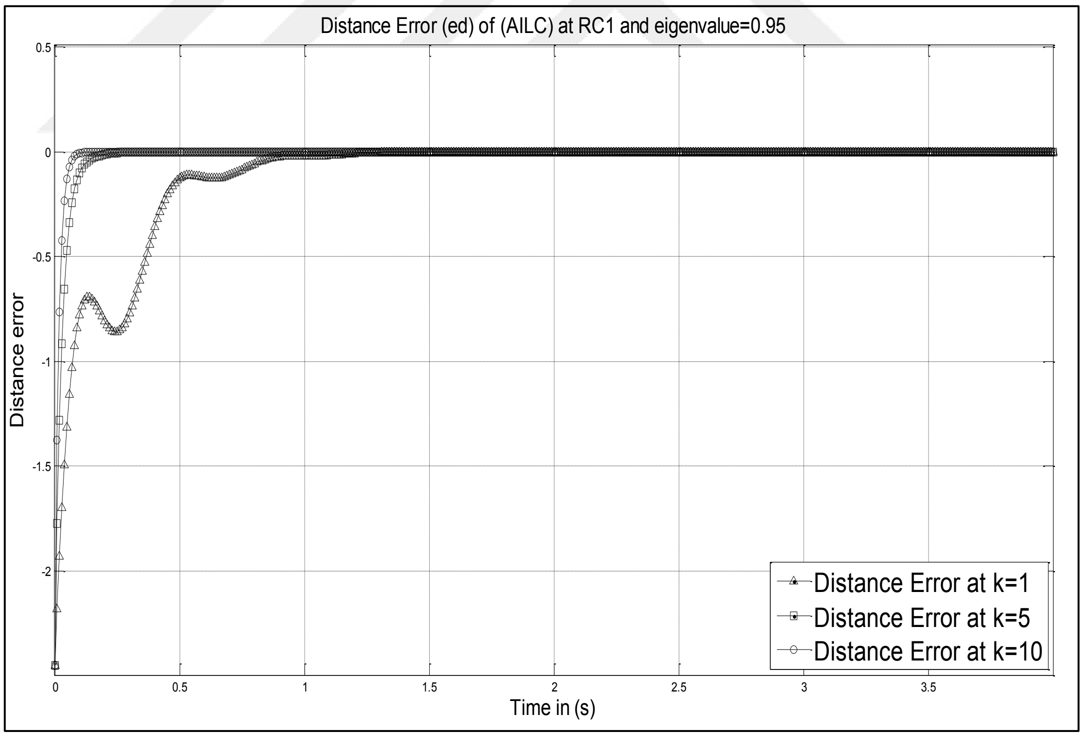


Figure 5.22: Distance Error e_d for RC_1 at $\lambda = 0.95$.

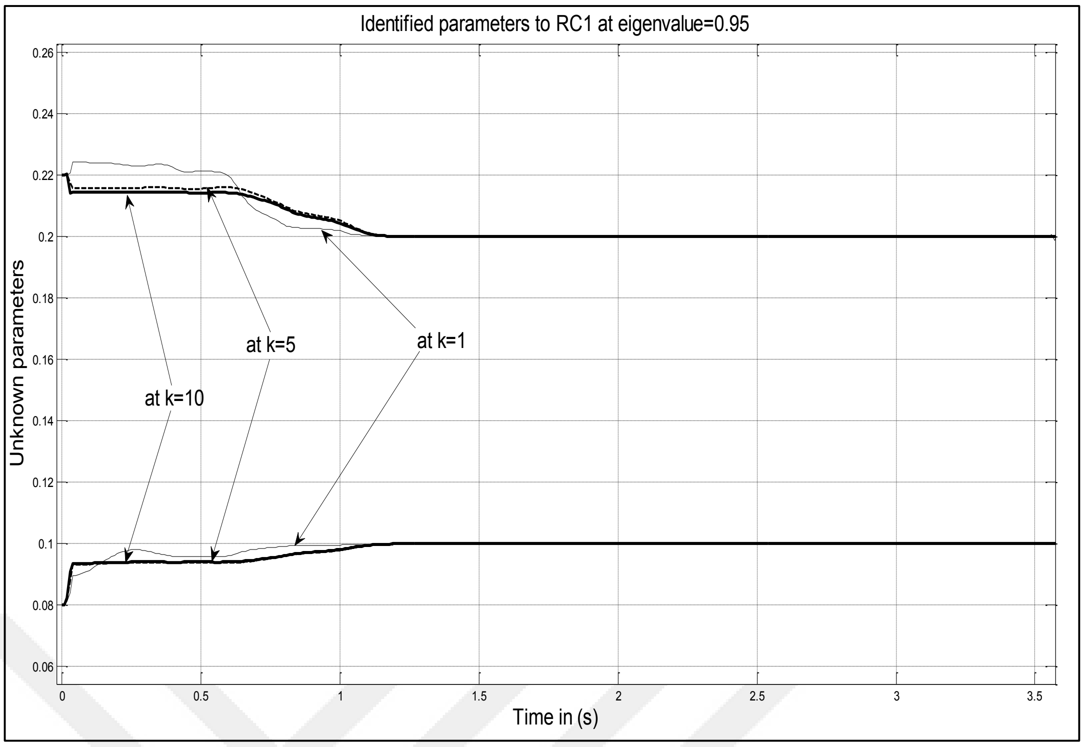


Figure 5.23: Identified Parameters $\hat{P}$ to RC_1 at $\lambda = 0.95$.

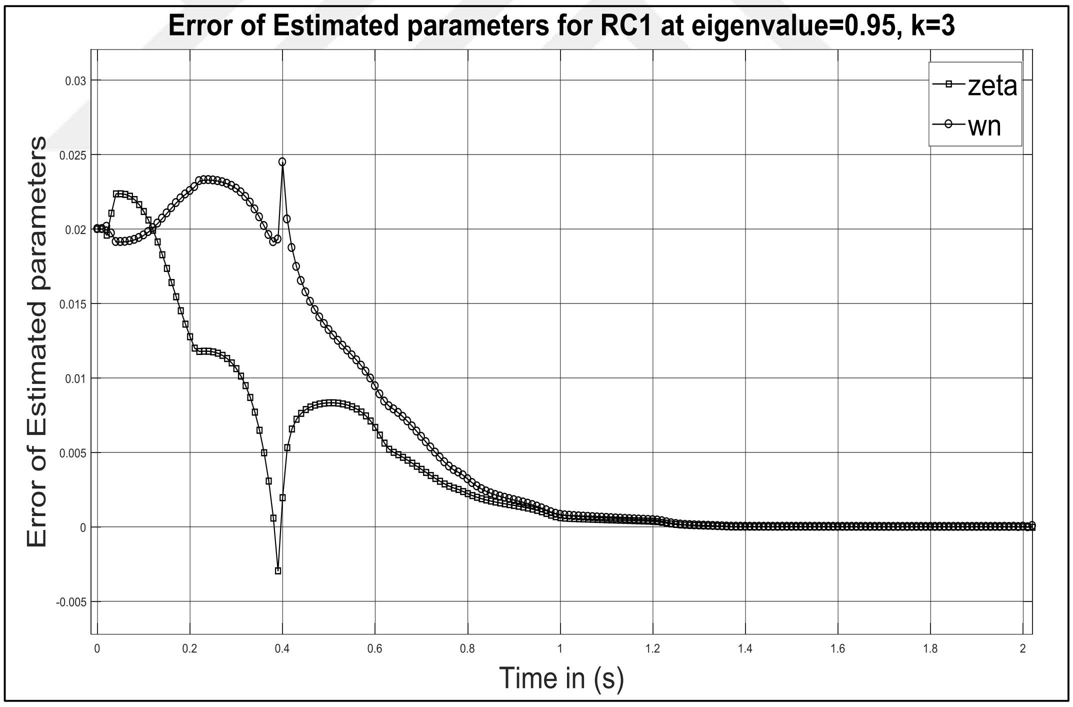


Figure 5.24: The Error e_i Between y and $\hat{y}$ to RC_1 at $\lambda = 0.95$.

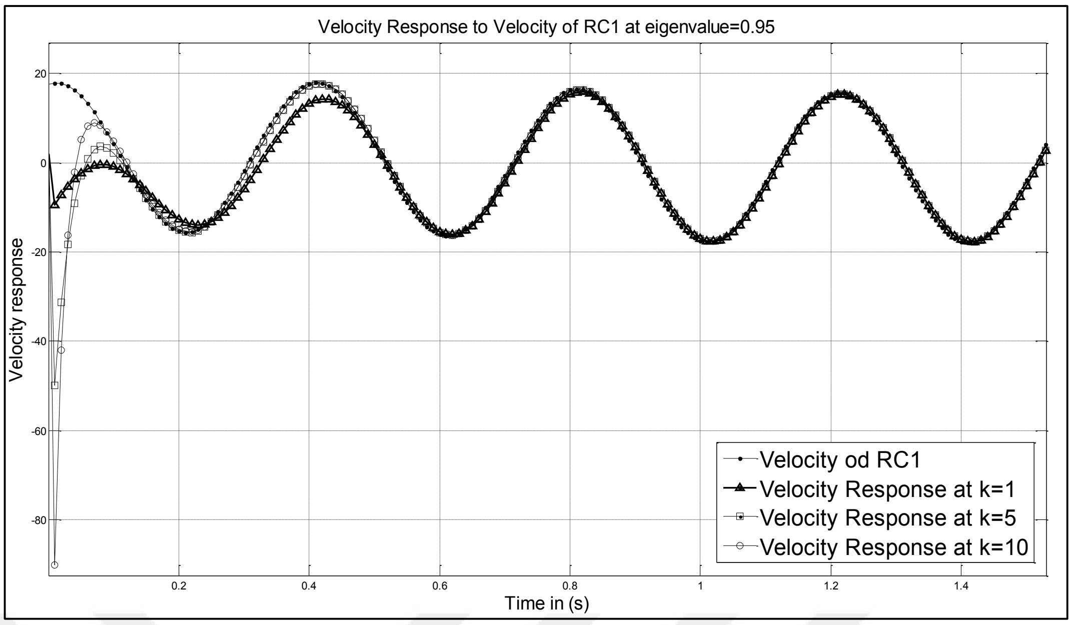


Figure 5.25: The Velocity (v) Response to Velocity of RC_1 at $\lambda = 0.95$.

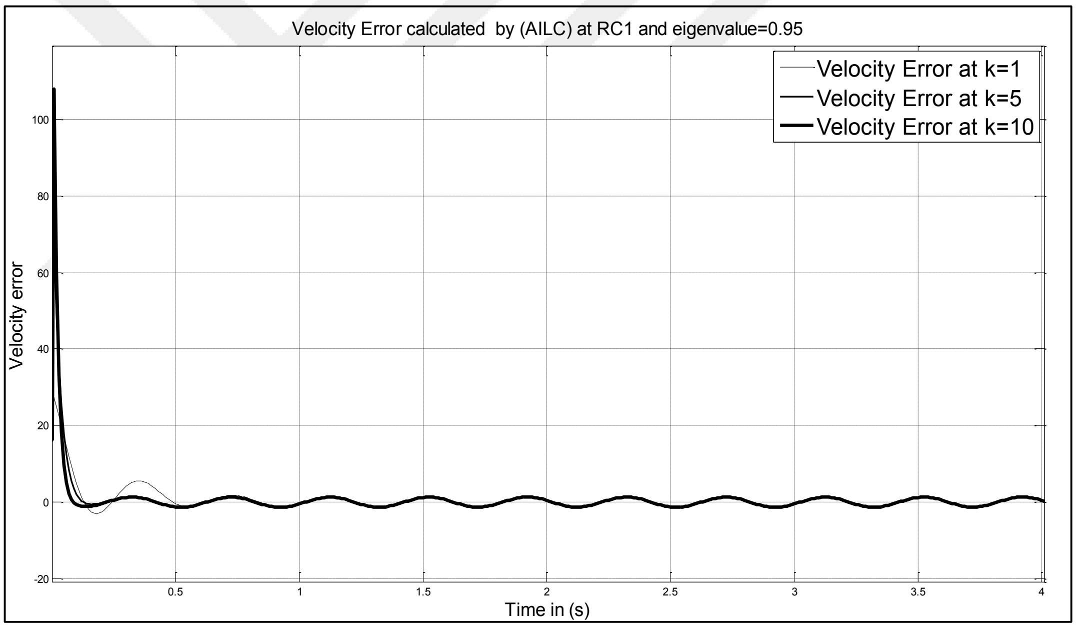


Figure 5.26: Velocity Error e_v for RC_1 at $\lambda = 0.95$.

5.3.3 AILC simulation for RC_1at $\lambda = 0.90$

In this part, AILC algorithm behavior with reference signal RC_1 will be shown when eigenvalue $\lambda = 0.90$, in this part, the convergence speed is faster than before due to the low number of eigenvalue, the lower the eigenvalue, the faster the convergence. The results will be arranged as before, where the input signal graph is shown in Figure 5.27, and the response in Figure 5.28 and Figure 5.29. The error e_d of distance in Figure 5.30, the identified parameters $\hat{P}$ in Figure 5.31, the error of the unknown parameter identificator in Figure 5.32, the velocity of the dynamic system to the velocity of the reference signal RC_1 is Figure 5.33, and the error of the two velocities e_v is shown in Figure 5.34.

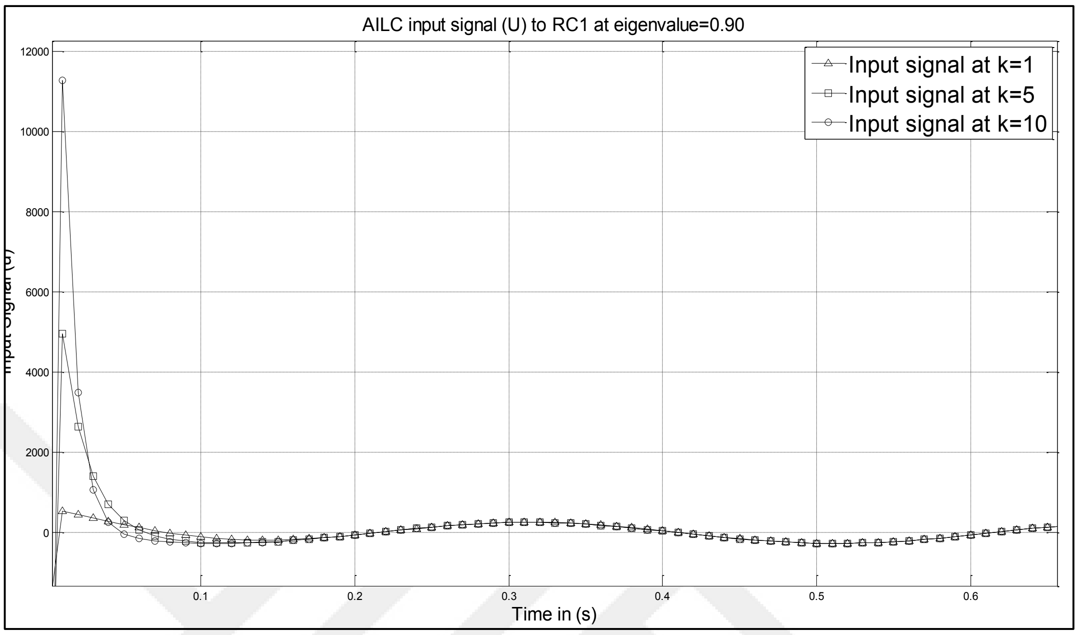


Figure 5.27: AILC Input Signal (U) for RC_1 Signal at $\lambda = 0.90$.

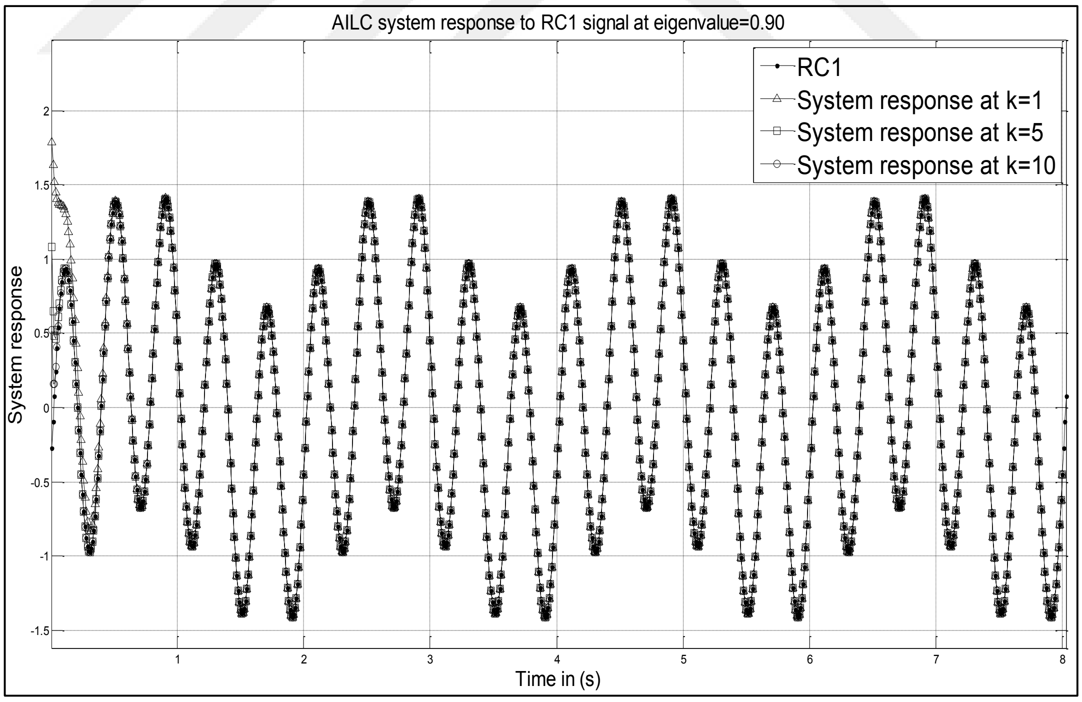


Figure 5.28: AILC Response to RC_1 Signal at $\lambda = 0.90$.

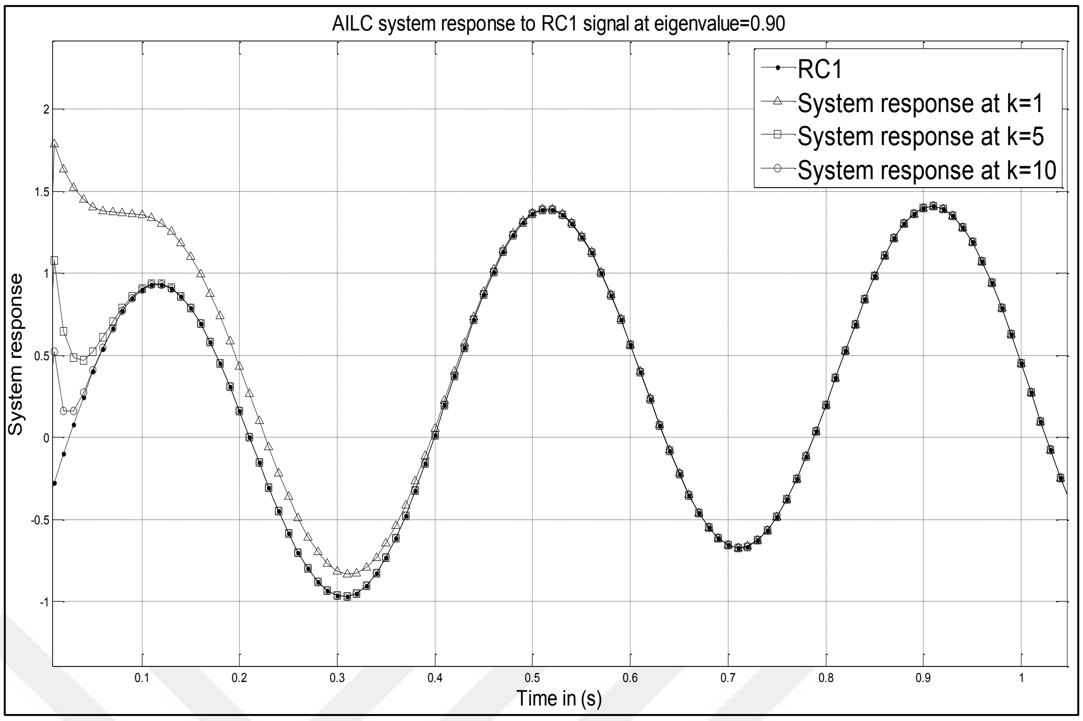


Figure 5.29: AILC Response to RC_1 Signal at $\lambda = 0.90$ (zoomed).

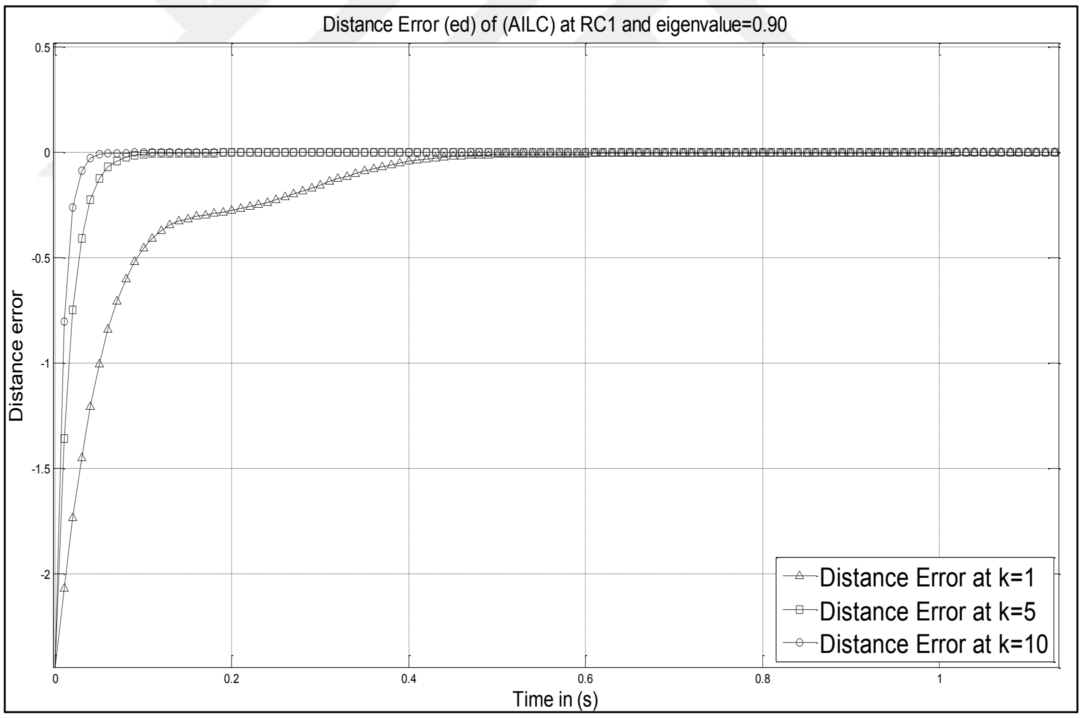


Figure 5.30: Distance Error e_d for RC_1 at $\lambda = 0.90$.

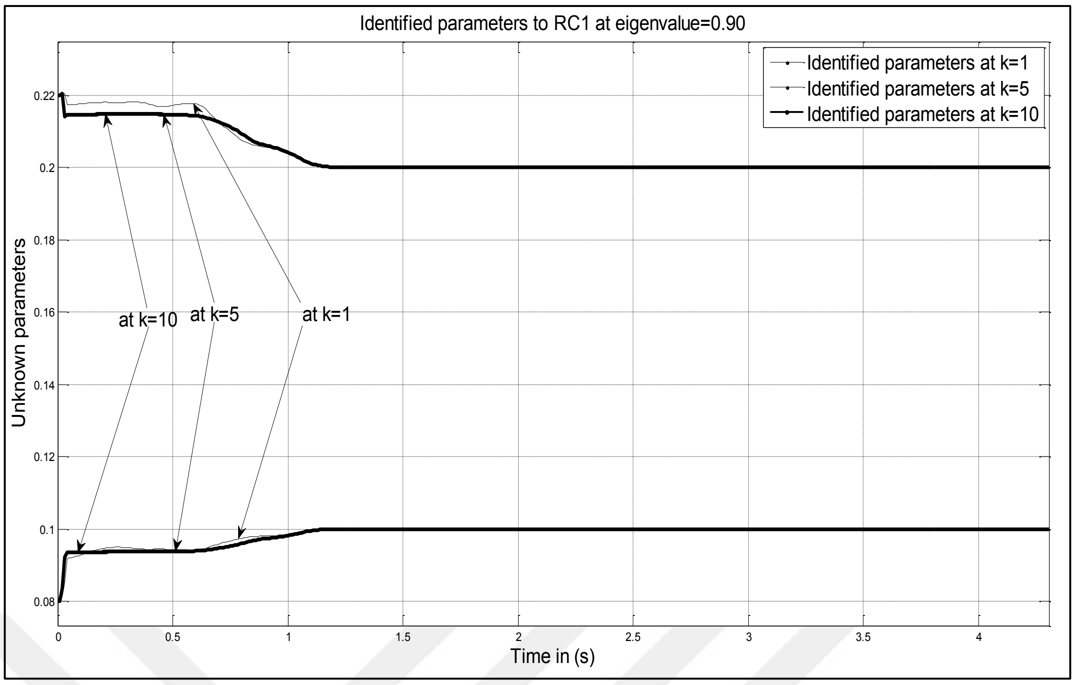


Figure 5.31: Identified Parameters $\hat{P}$ to RC_1 at $\lambda = 0.90$.

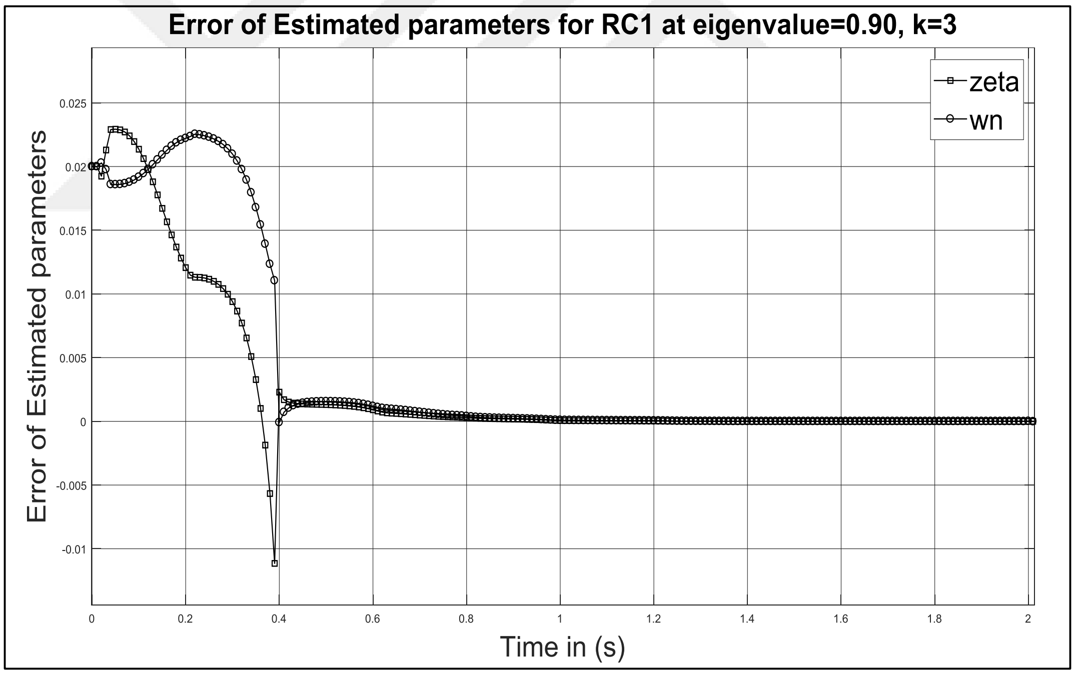


Figure 5.32: The Error e_i Between y and $\hat{y}$ to RC_1 at $\lambda = 0.90$.

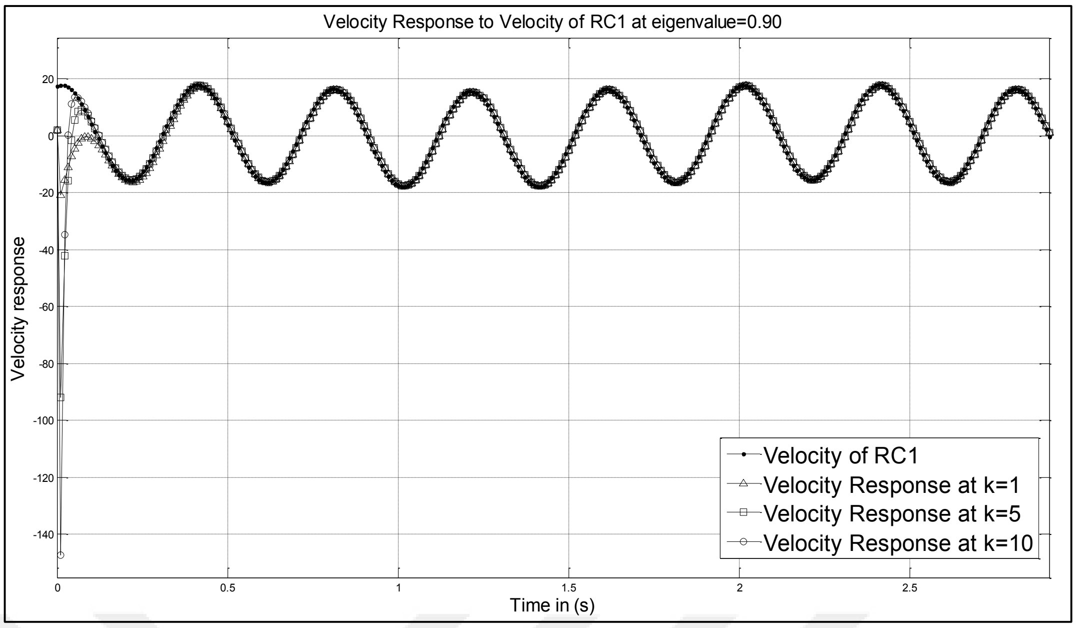


Figure 5.33: The Velocity (v) Response to Velocity of RC_1 at $\lambda = 0.90$.

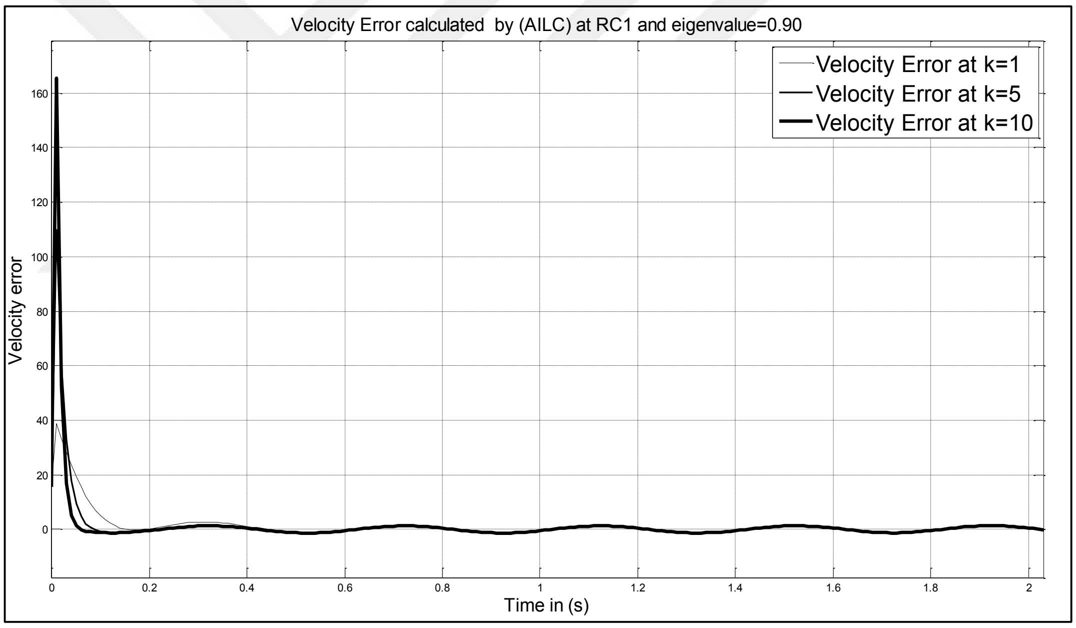


Figure 5.34: Velocity Error e_v for RC_1 at $\lambda = 0.90$.

5.3.4 AILC simulation for RC_2 at $\lambda = 0.99$

In this part and the next two parts, the same three previous simulations will be repeated but on the second reference signal RC_2 , this reference signal has rough and sudden movements which makes its tracking harder than RC_2 . In this part of simulation, AILC algorithm application on RC_2 at eigenvalue $\lambda = 0.99$ will be shown at the same initial identified parameters $\widehat{P}_i = [0.08 \quad 0.22]^T$, where the input signal graph is shown in Figure 5.35, and the response of the dynamic system is shown in

Figure 5.36 and Figure 5.37. The distance error e_d is illustrated in Figure 5.38, the identified parameters $\hat{P}$ is shown in Figure 5.39, the error of the unknown parameter identificator is shown in Figure 5.40, the velocity of the dynamic system to the velocity of the reference signal RC_2 is Figure 5.41, and the error of the two velocities e_v is shown in Figure 5.42.

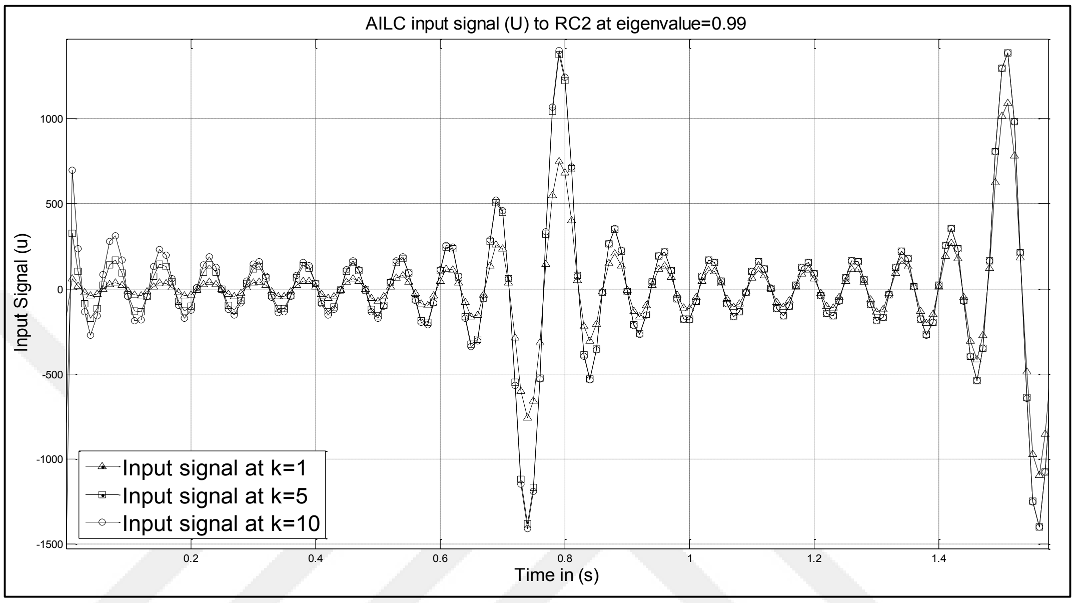


Figure 5.35: AILC Input Signal (U) for RC_2 Signal at $\lambda = 0.99$.

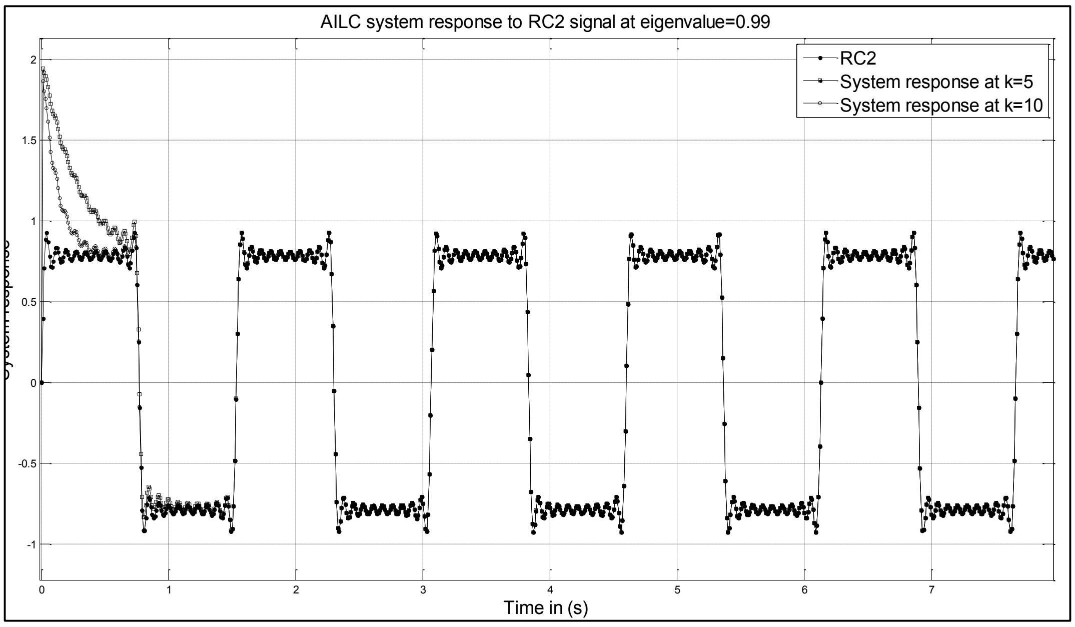


Figure 5.36: AILC Response to RC_2 Signal at $\lambda = 0.99$.

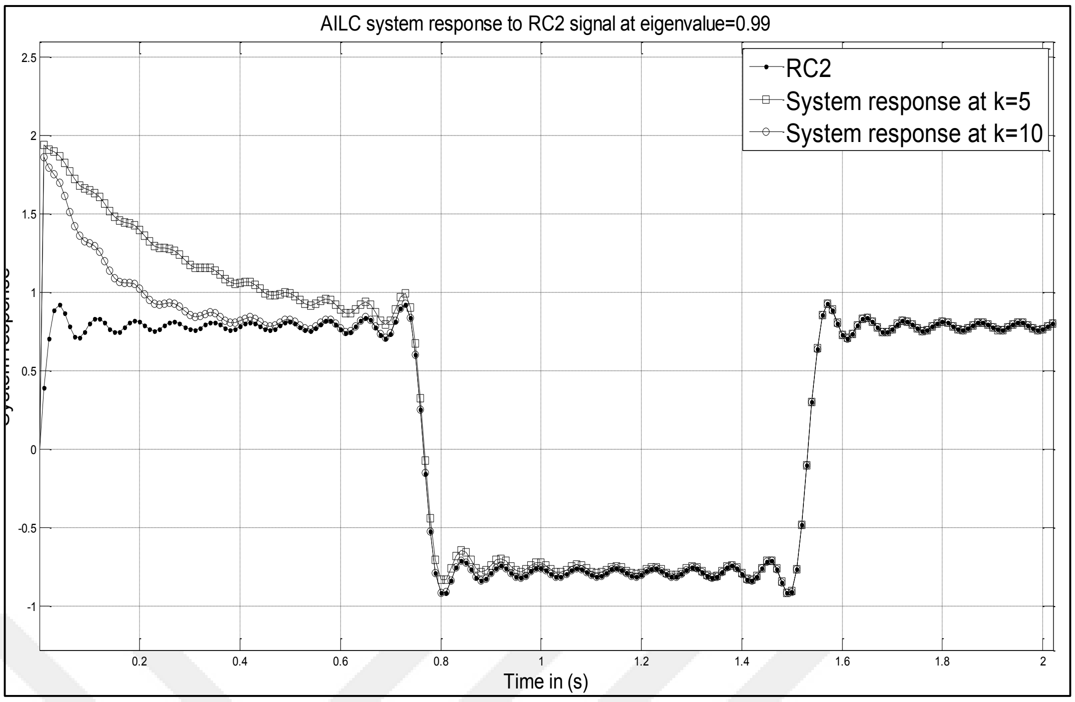


Figure 5.37: AILC Response to RC_2 Signal at $\lambda = 0.99$ (zoomed).

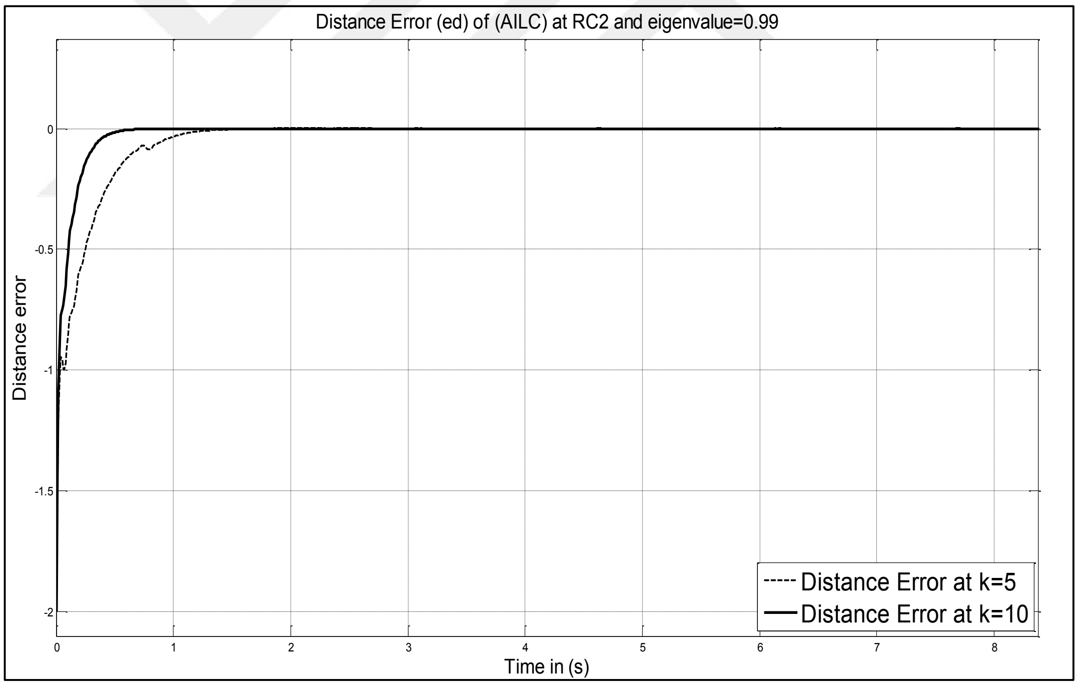


Figure 5.38: Distance Error e_d for RC_2 at $\lambda = 0.99$.

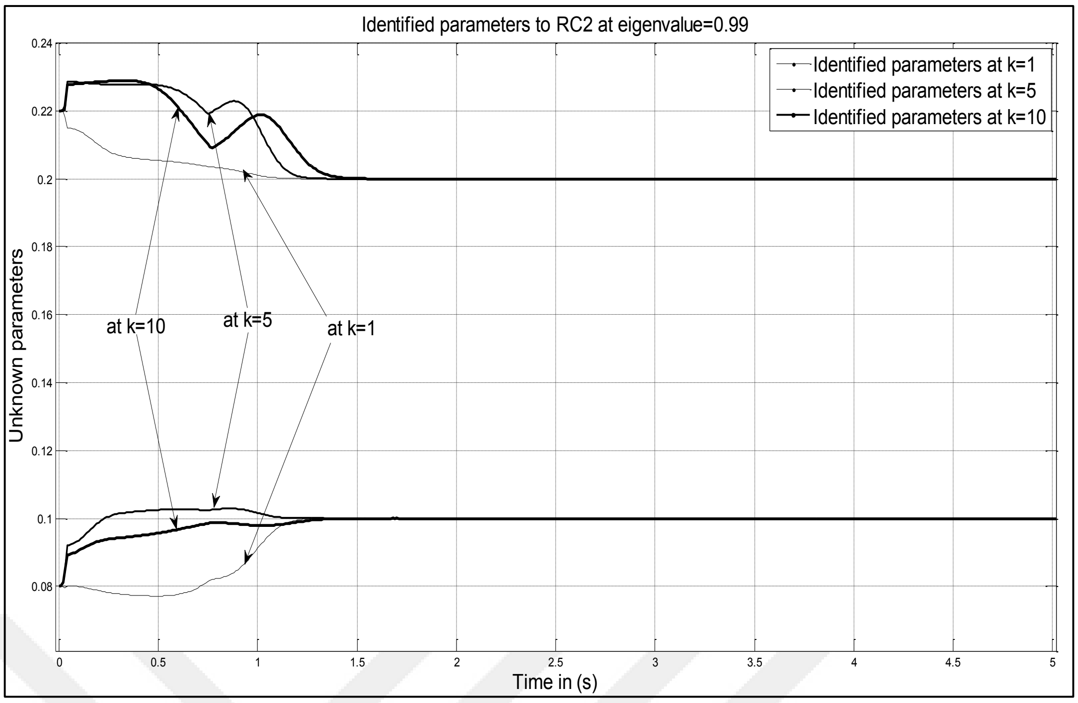


Figure 5.39: Identified Parameters $\hat{P}$ to RC_2 at $\lambda = 0.99$.

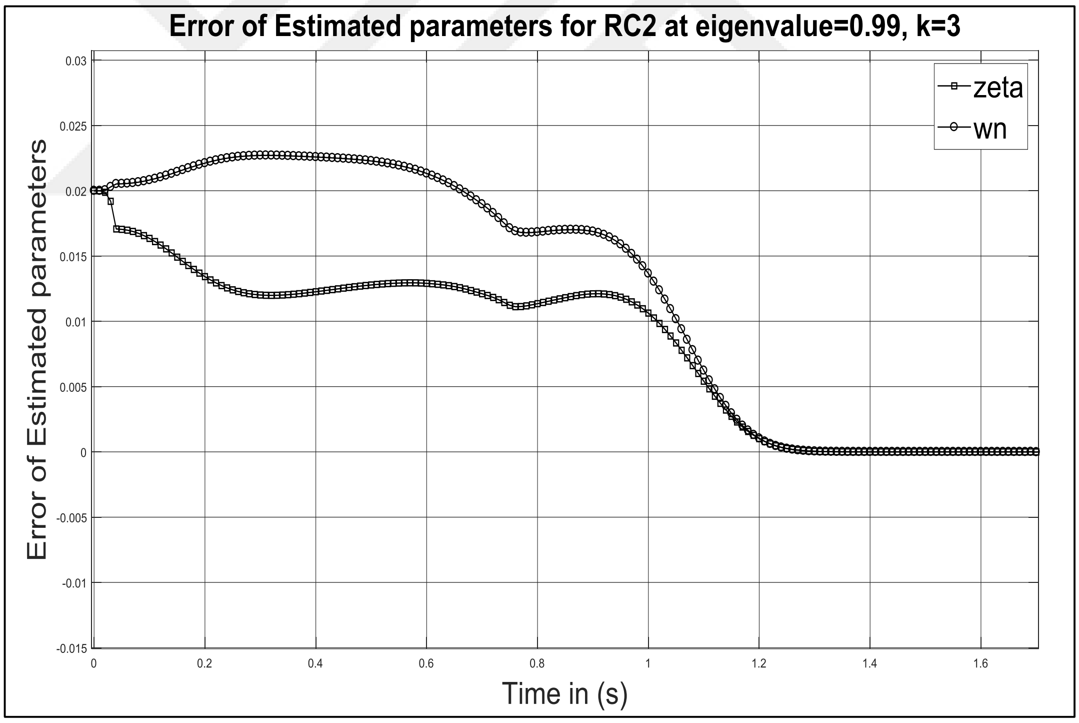


Figure 5.40: The Error e_i Between y and $\hat{y}$ to RC_2 at $\lambda = 0.99$.

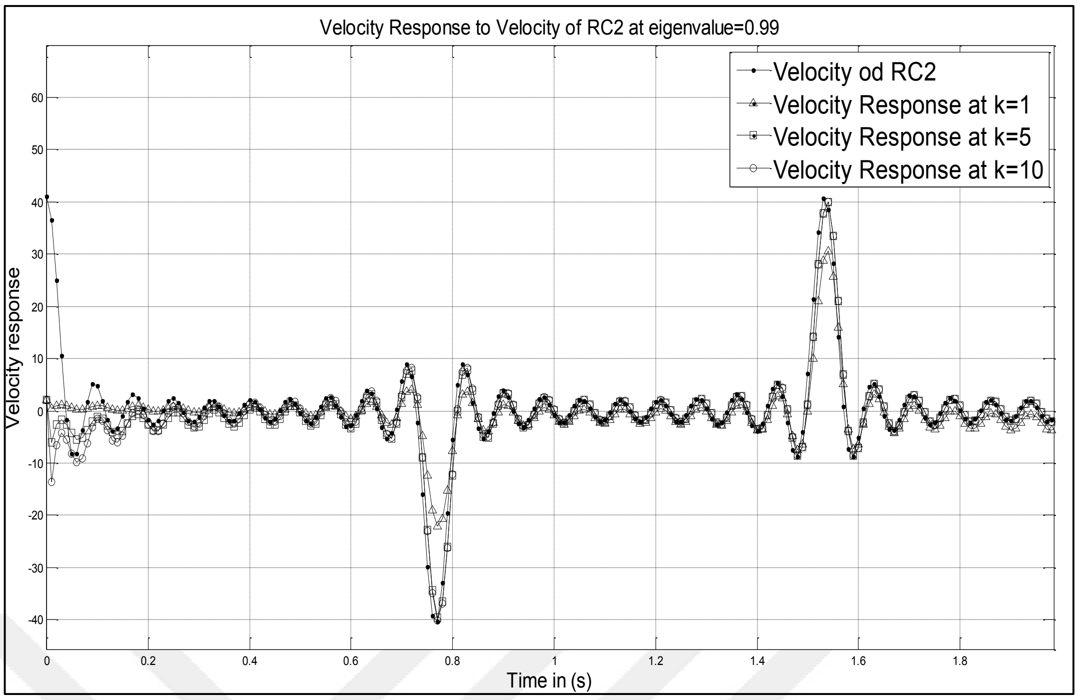


Figure 5.41: The Velocity (v) Controlled by (AILC) at RC_2 and $\lambda = 0.99$.

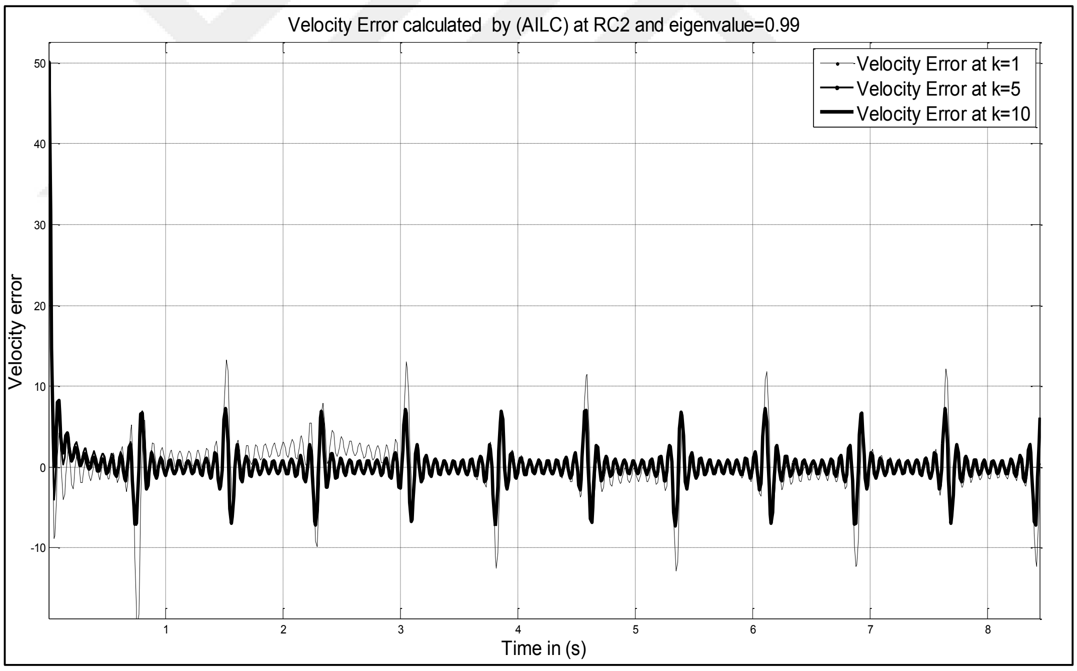


Figure 5.42: Velocity Error e_v for RC_2 at $\lambda = 0.99$.

5.3.5 AILC simulation for RC_2 at $\lambda = 0.95$

The same previous part will be repeated in this part except the eigenvalue will reduce to $\lambda = 0.95$, where the input signal graph is shown in Figure 5.43, the response is shown in Figure 5.44 and Figure 5.45, the error e_d of distance is illustrated in Figure 5.46, the identified parameters $\hat{P}$ is shown in Figure 5.47, error of the unknown parameter identificator is shown in Figure 5.48, the velocity of the dynamic system to the velocity of the reference signal RC_2 is Figure 5.49, and the error of the two velocities e_v is shown in Figure 5.50.

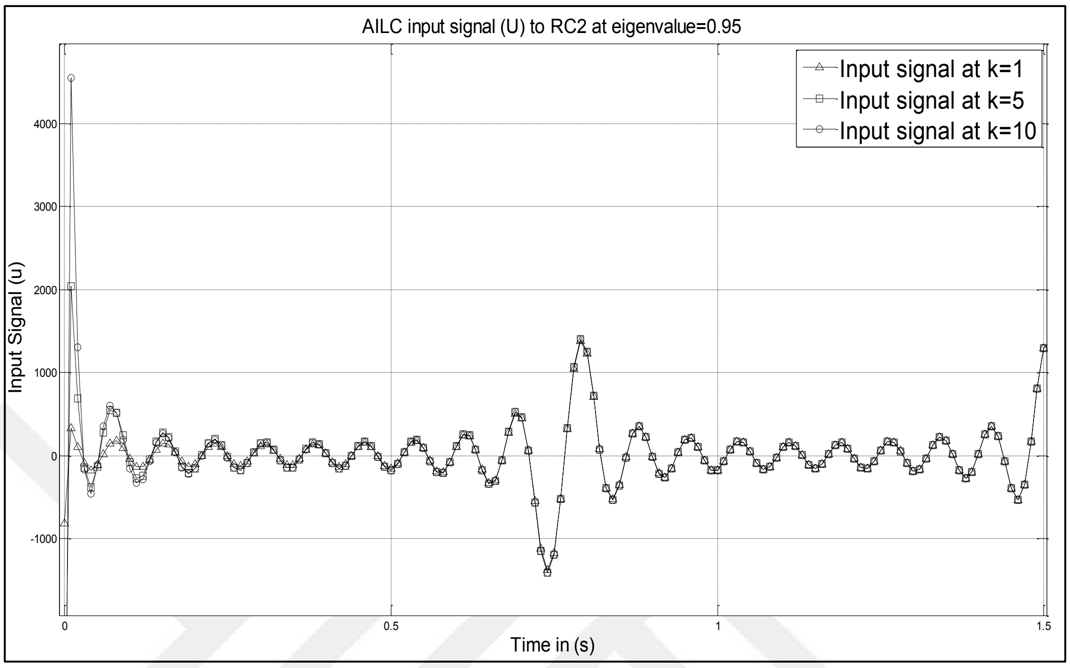


Figure 5.43: AILC Input Signal (U) for RC_2 Signal at $\lambda = 0.95$.

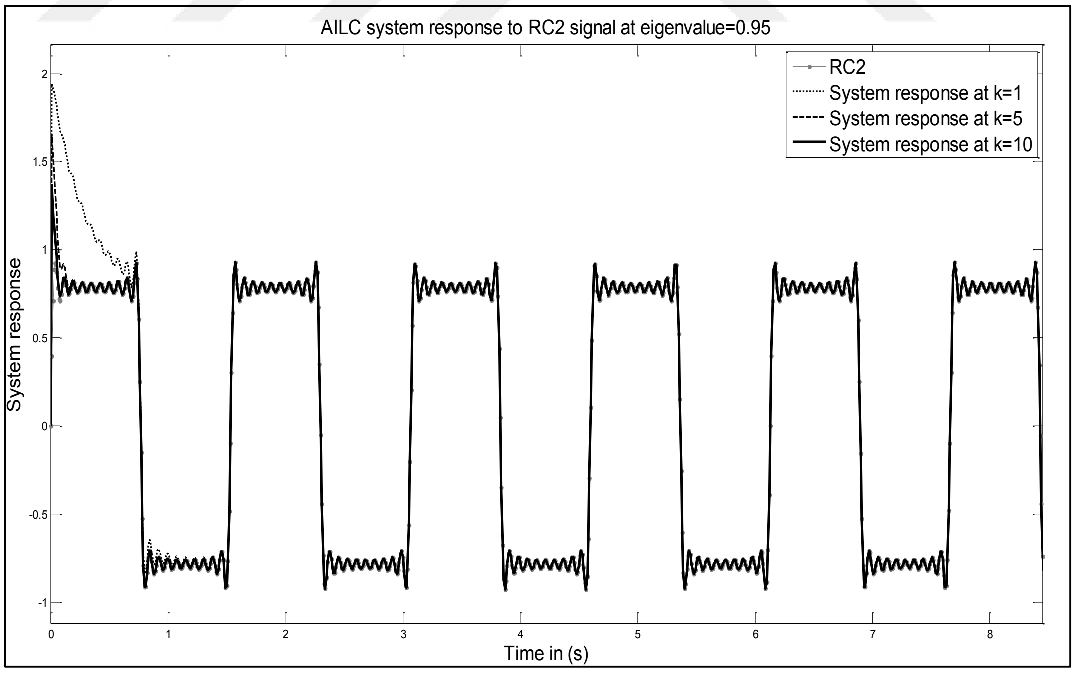


Figure 5.44: AILC Response to RC_2 Signal at $\lambda = 0.95$.

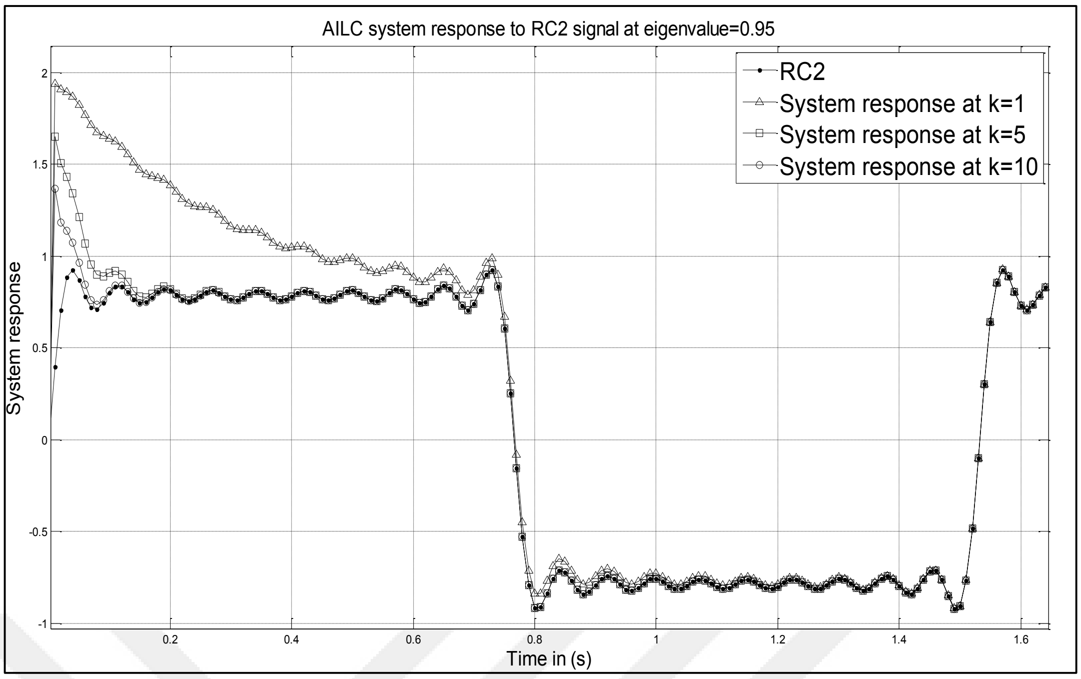


Figure 5.45: AILC Response to RC_2 Signal at $\lambda = 0.95$ (zoomed).

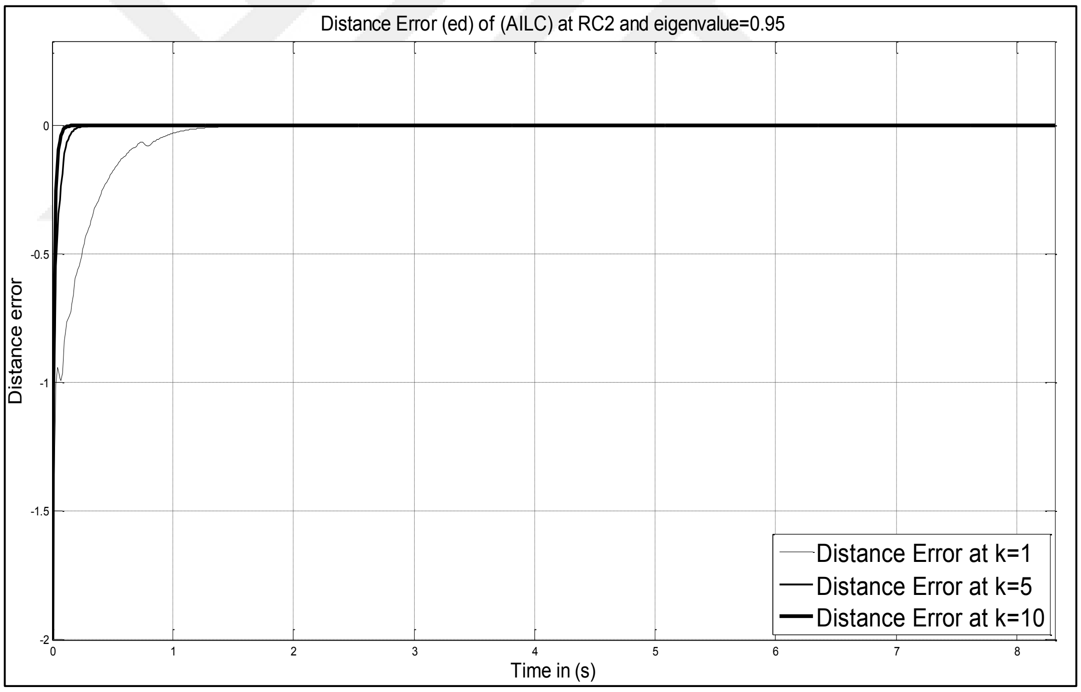


Figure 5.46: Distance Error e_d for RC_2 at $\lambda = 0.95$.

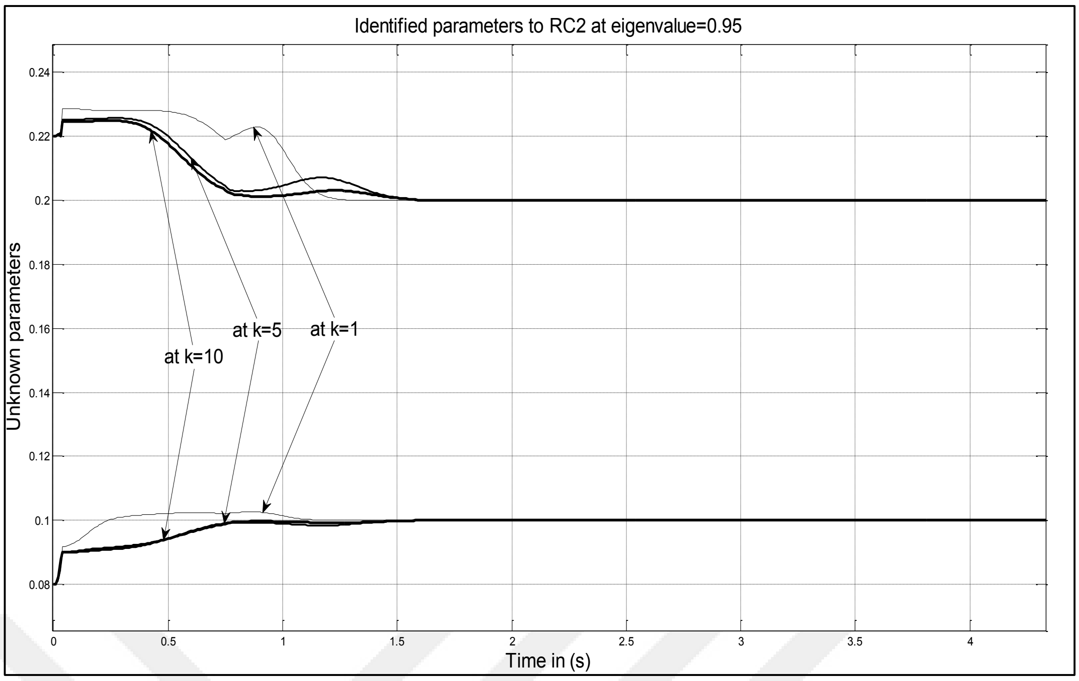


Figure 5.47: Identified Parameters $\hat{P}$ to RC_2 at $\lambda = 0.95$.

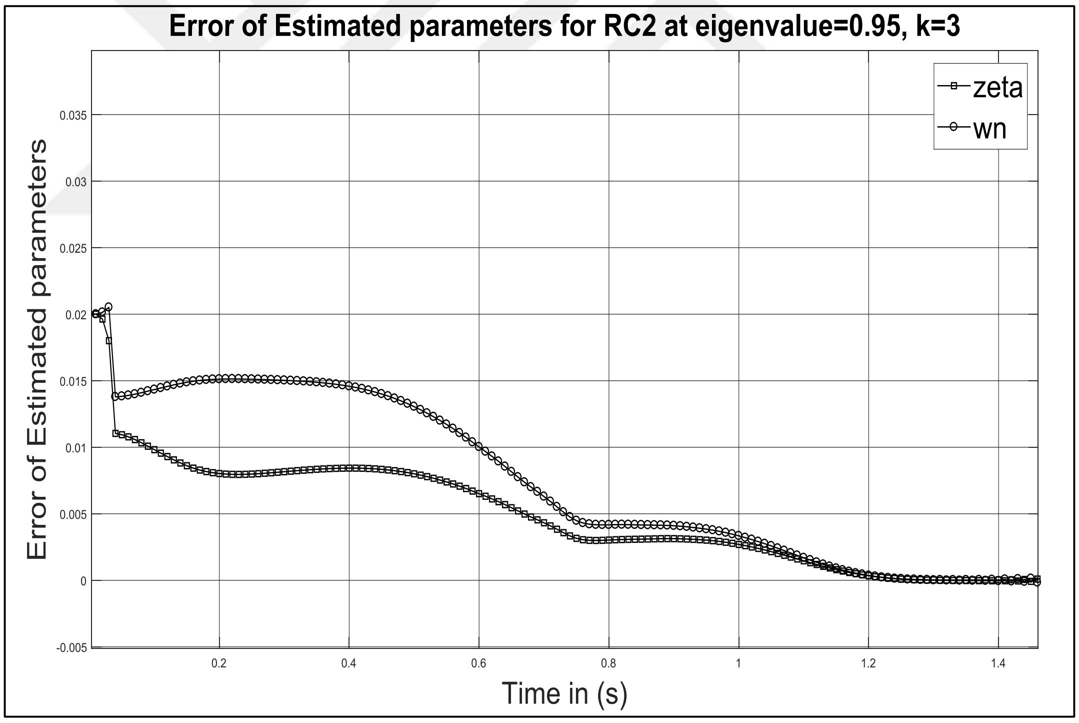


Figure 5.48: The Error e_i Between y and $\hat{y}$ to RC_2 at $\lambda = 0.95$.

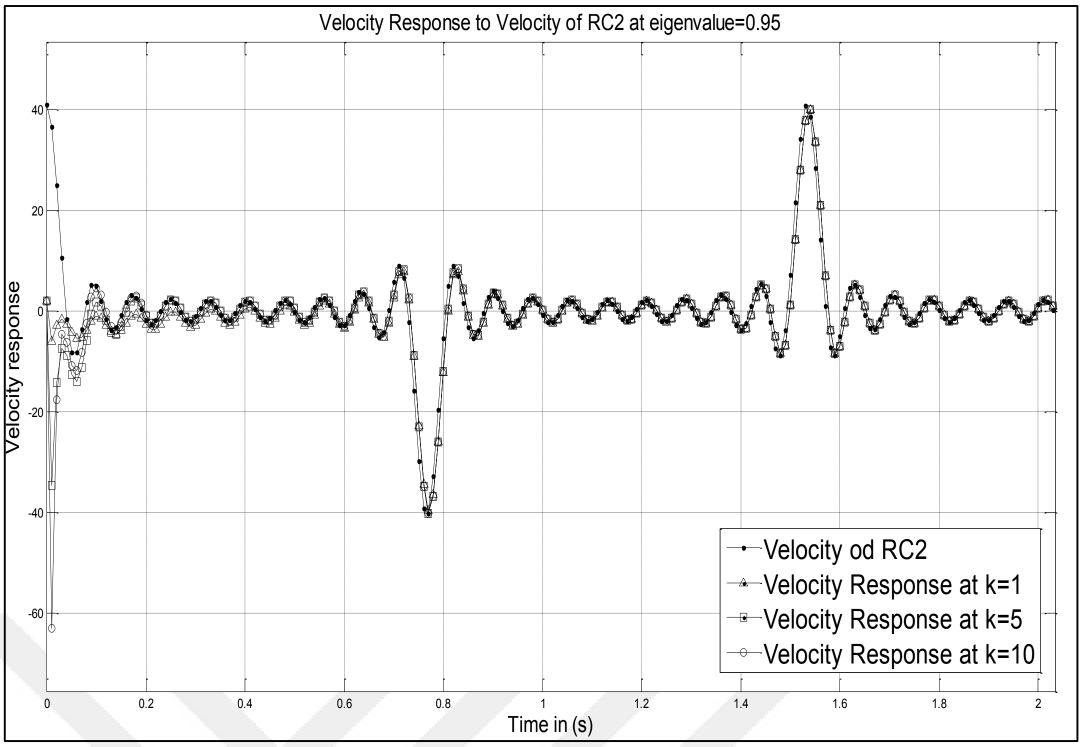


Figure 5.49: The Velocity (v) Controlled by (AILC) at RC_2 and $\lambda = 0.95$.

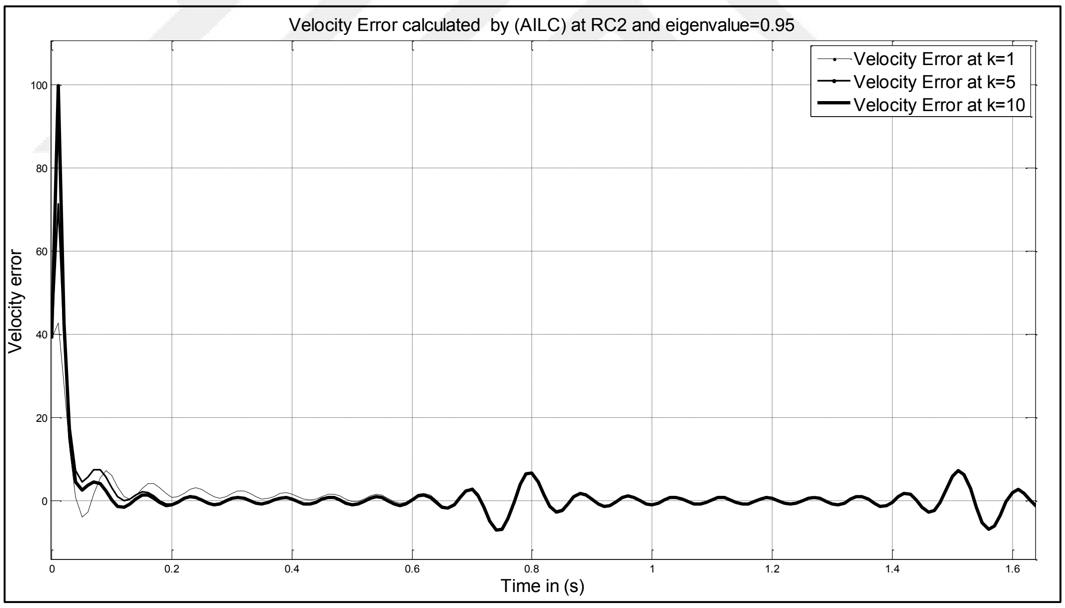


Figure 5.50: Velocity Error e_v for RC_2 at $\lambda = 0.95$.

5.3.6 AILC simulation for RC_2 at $\lambda = 0.90$

In this part, eigenvalue is taken as $\lambda = 0.90$ as the lowest value in this monograph, which will show the fastest convergence and the lowest number of AILC algorithm's iterations, where the input signal graph is shown in Figure 5.51, the response is shown in Figure 5.52 and Figure 5.53, the distance error e_d is illustrated in

Figure 5.54, the identified parameters $\hat{P}$ is shown in Figure 5.55, error of the unknown parameter identificator is shown in Figure 5.56, the velocity of the dynamic system to the velocity of the reference signal RC_2 is Figure 5.57, and the error of the two velocities e_v is shown in Figure 5.58.

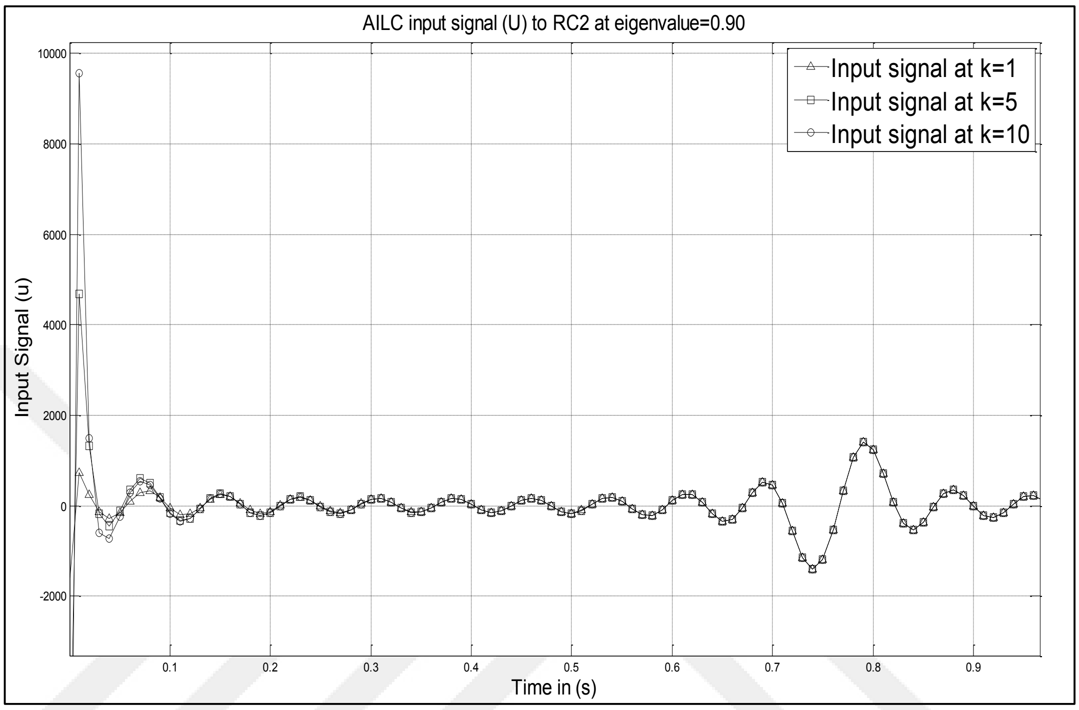


Figure 5.51: AILC Input Signal (U) for RC_2 Signal at $\lambda = 0.90$.

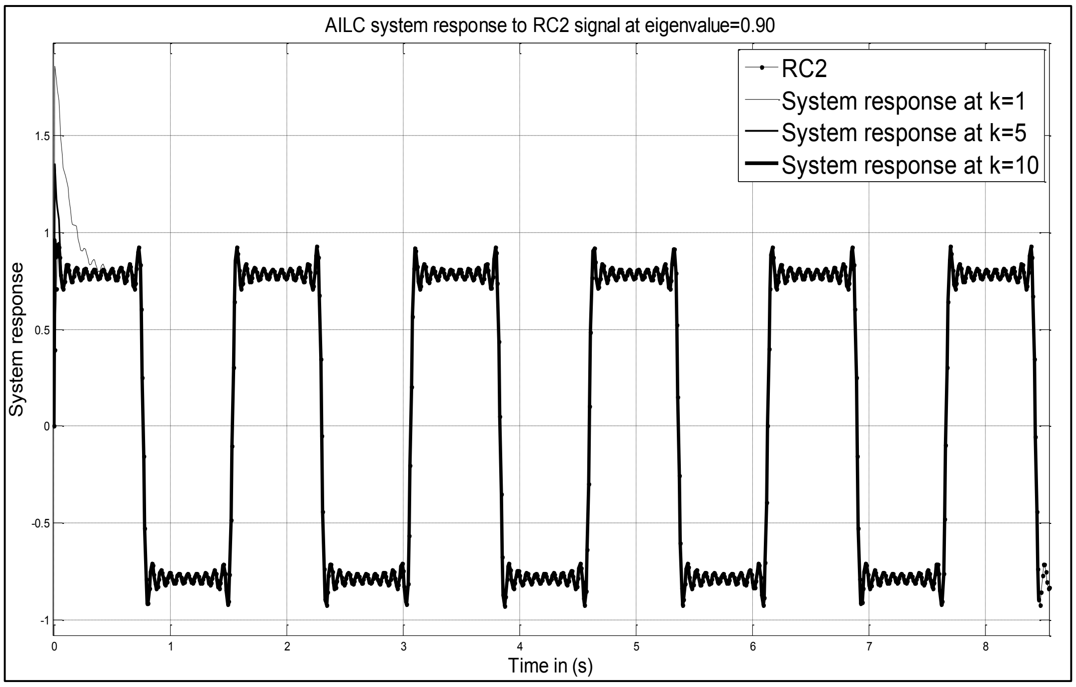


Figure 5.52: AILC Response to RC_2 Signal at $\lambda = 0.90$.

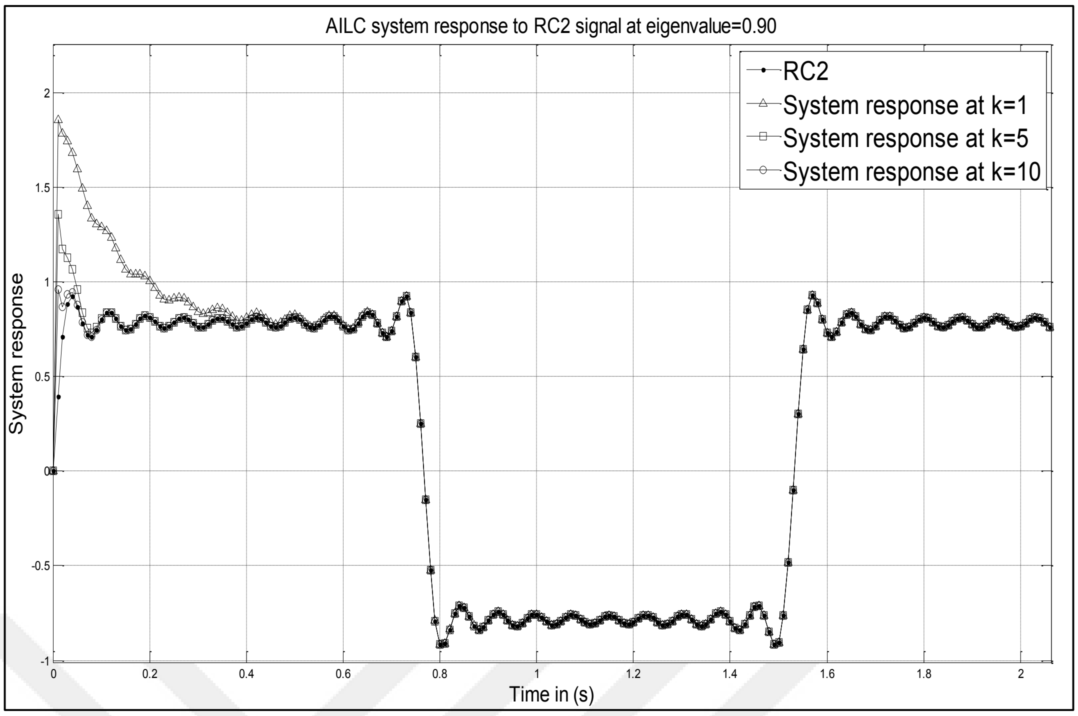


Figure 5.53: AILC Response to RC_2 Signal at $\lambda = 0.90$ (zoomed).

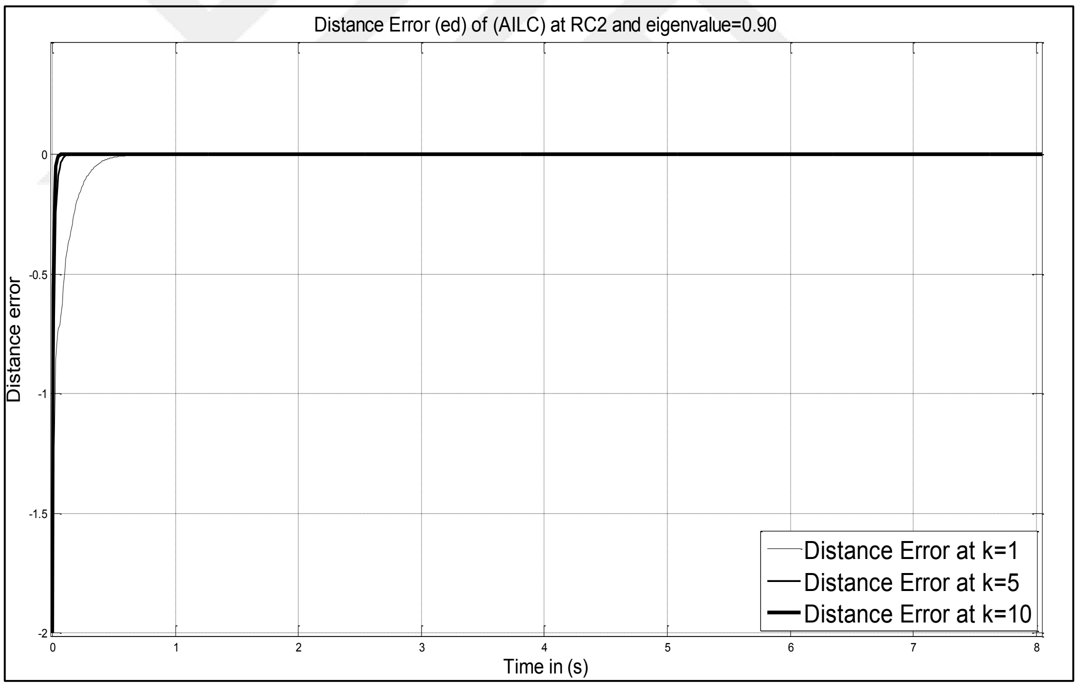


Figure 5.54: Distance Error e_d for RC_2 at $\lambda = 0.90$.

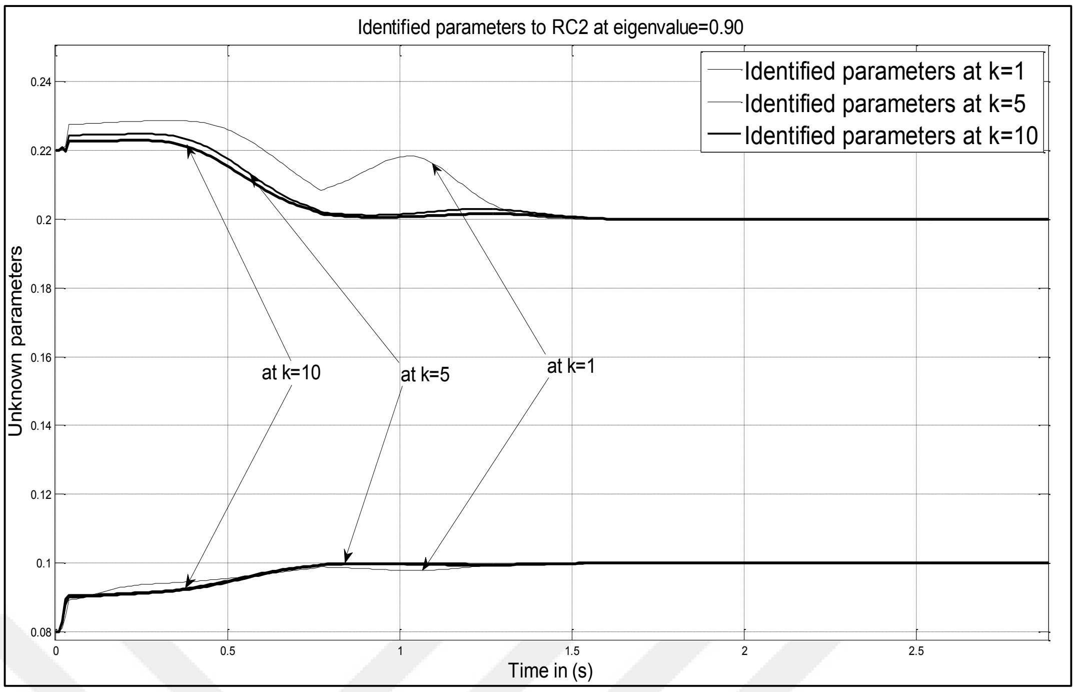


Figure 5.55: Identified Parameters $\hat{P}$ to RC_2 at $\lambda = 0.90$.

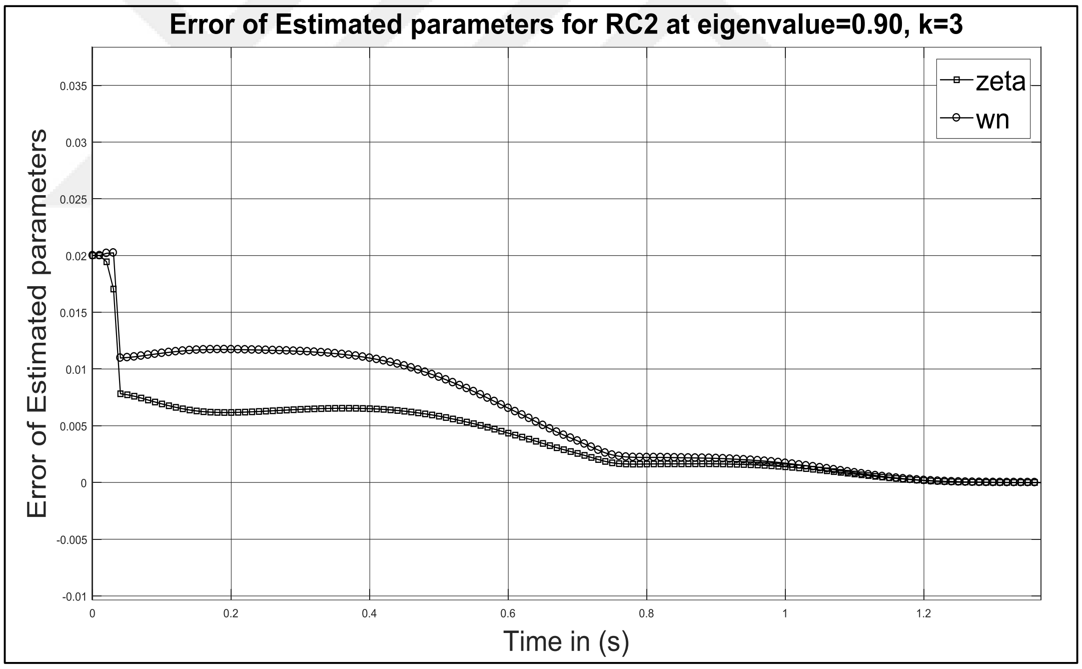


Figure 5.56: The Error e_i Between y and $\hat{y}$ to RC_2 at $\lambda = 0.90$.

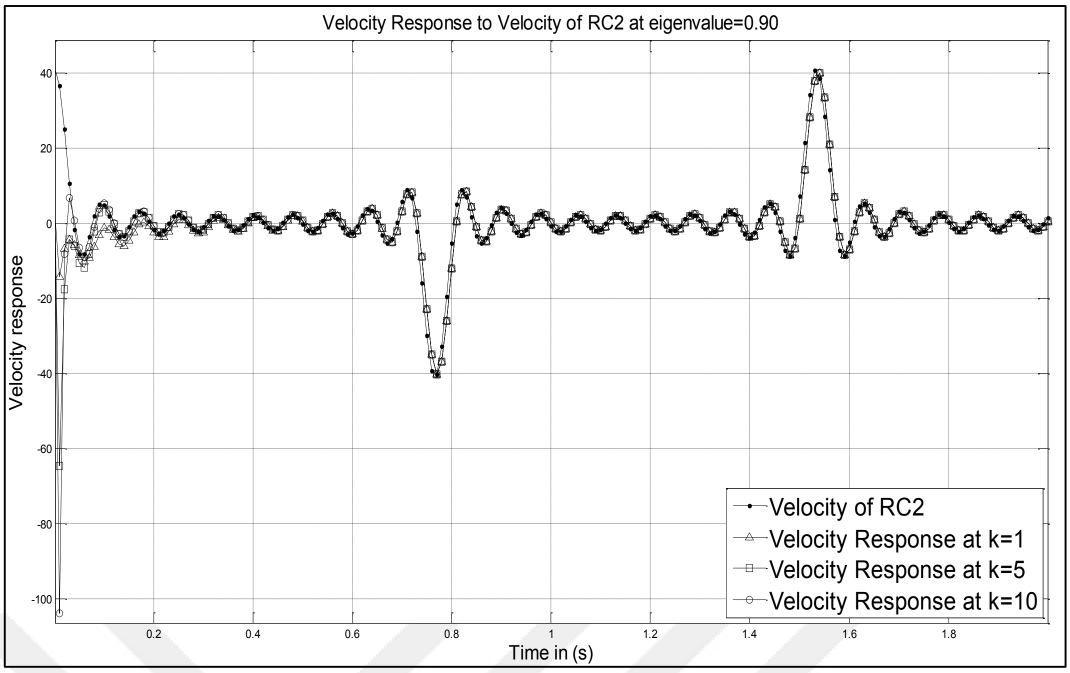


Figure 5.57: The Velocity (v) Controlled by (AILC) at RC_2 and $\lambda = 0.90$.

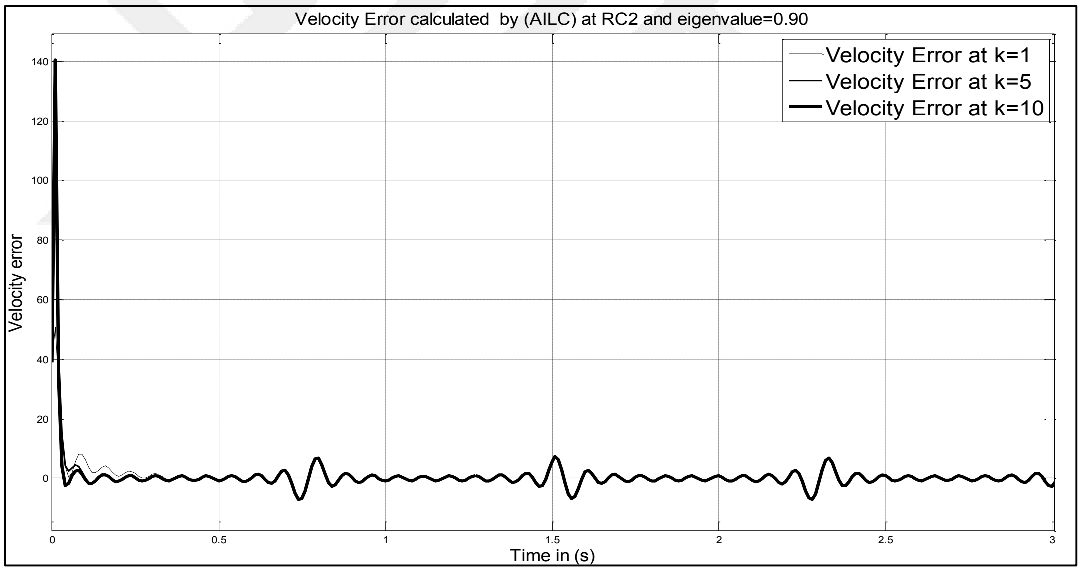


Figure 5.58: Velocity Error e_v for RC_2 at $\lambda = 0.90$.

5.3.7 AILC sensitivity for different initial identified parameters

All of the previous results were taken when the initial identified parameters values as expressed in the vector $\widehat{P}_i = [0.08 \quad 0.22]^T$. In this part, several initial identified parameters $\widehat{P}_i$ will be taken to show how the AILC will be affected by each of them. The reference signals RC_1 and RC_2 will be taken in this part at eigenvalues $\lambda = 0.99$, $\lambda = 0.95$ and $\lambda = 0.90$, the common iteration number $k = 3$ to show observable difference between the results. Several sets of initial points will be taken in this part of the simulation, these values will be farther from the main value that it depended on most parts of this simulation in order to show the effectiveness of AILC algorithm.

The identified parameters of the signal RC_1 at $\lambda = 0.99$ with the three sets are shown in Figure 5.59, when eigenvalue $\lambda = 0.95$ the identified parameters graph will be as shown in Figure 5.60, and the Figure 5.61 shows the identified parameters when eigenvalue $\lambda = 0.90$.

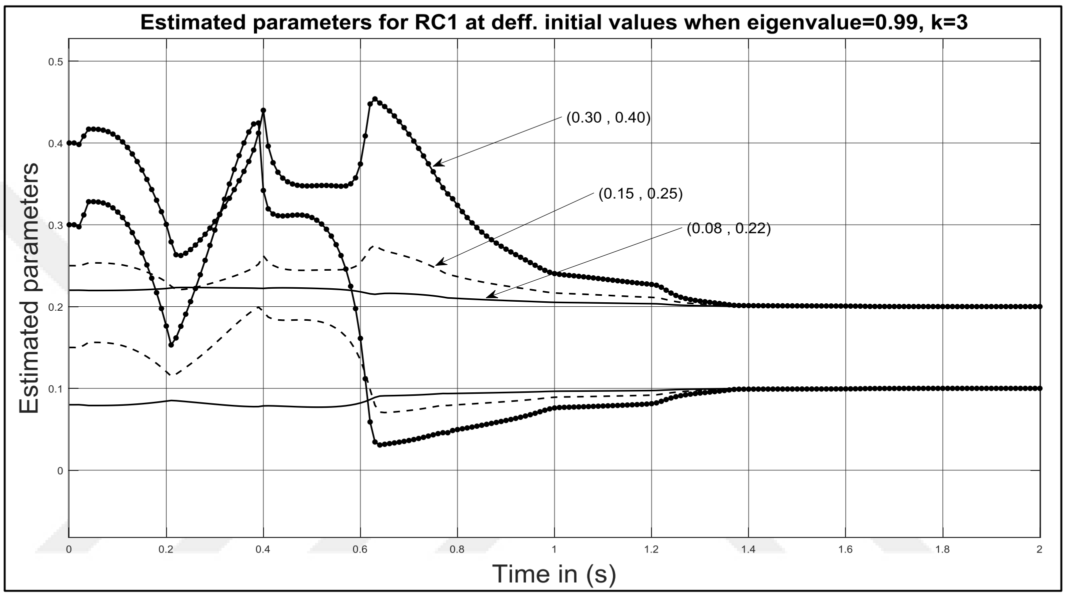


Figure 5.59: Identified Parameters for RC_1 Signal at $\lambda = 0.99$ and k=3.

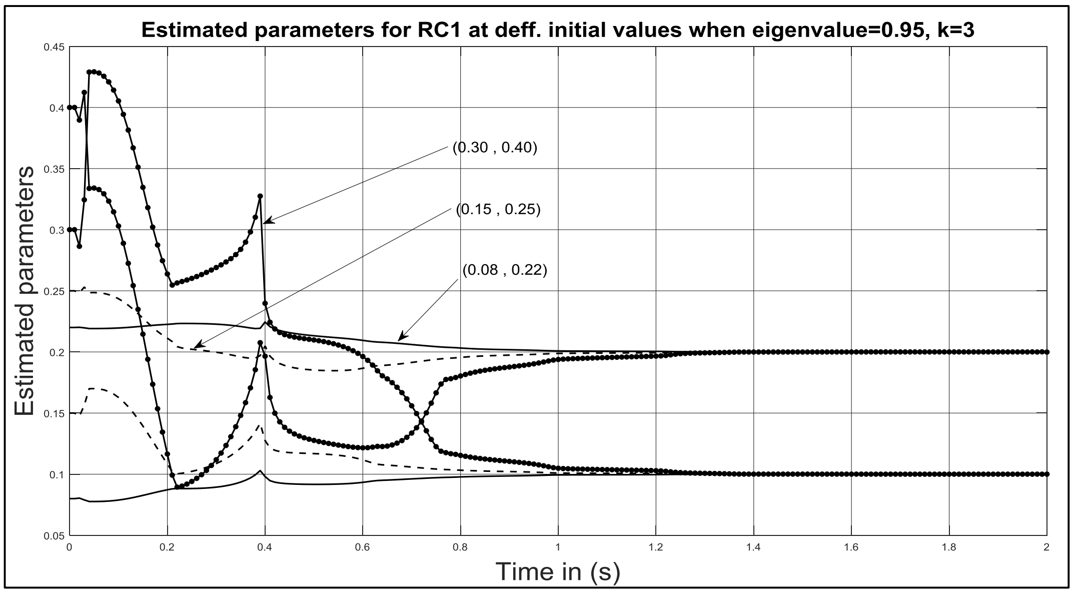


Figure 5.60: Identified Parameters for RC_1 Signal at $\lambda = 0.95$ and k=3.

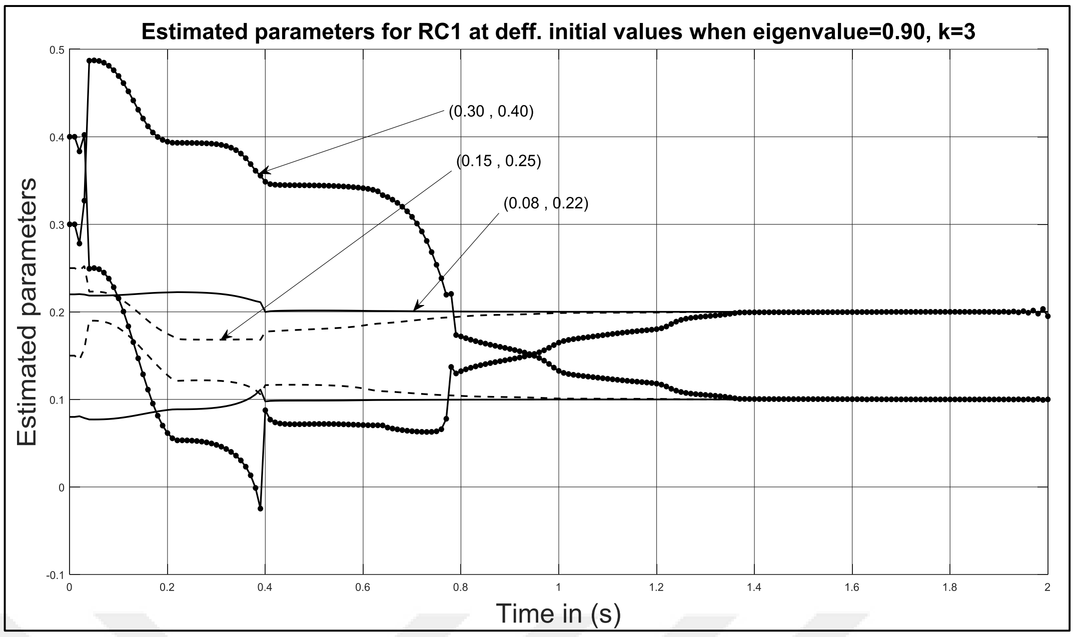


Figure 5.61: Identified Parameters for RC_1 Signal at $\lambda = 0.90$ and k=3.

The same previous process will be taken for the second reference signal RC_2 . The identified parameters when the eigenvalue $\lambda = 0.99$, the three sets' results are shown in Figure 5.62, when eigenvalue $\lambda = 0.95$ the identified parameters graph will be as shown in Figure 5.63, and the Figure 5.64 shows the identified parameters when eigenvalue $\lambda = 0.90$.

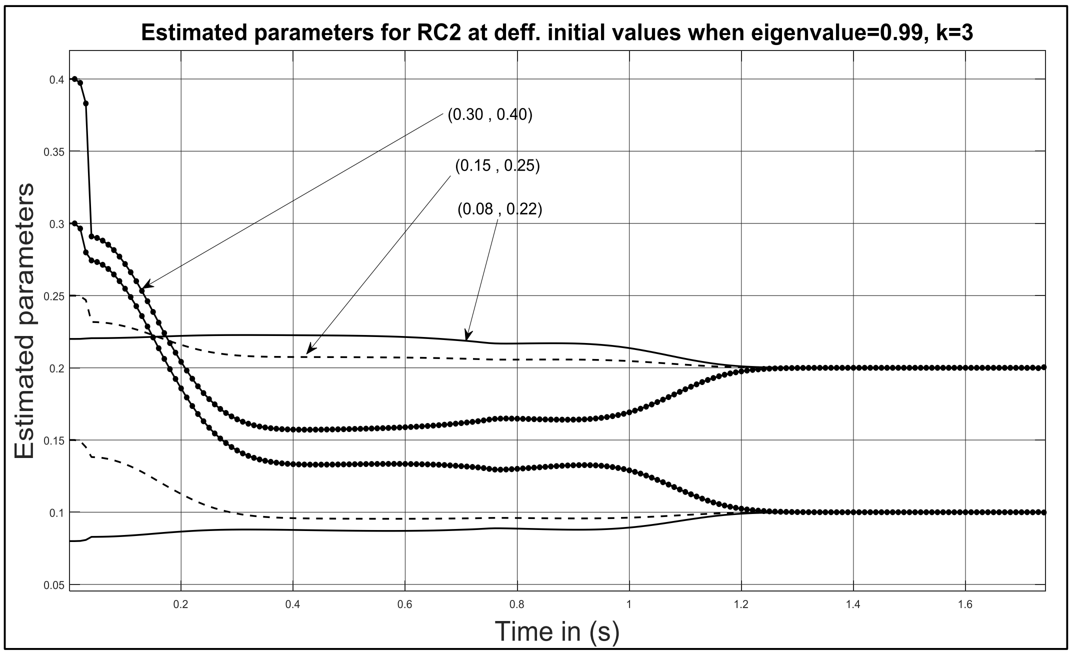


Figure 5.62: Identified Parameters for RC_2 Signal at $\lambda = 0.99$ and k=3.

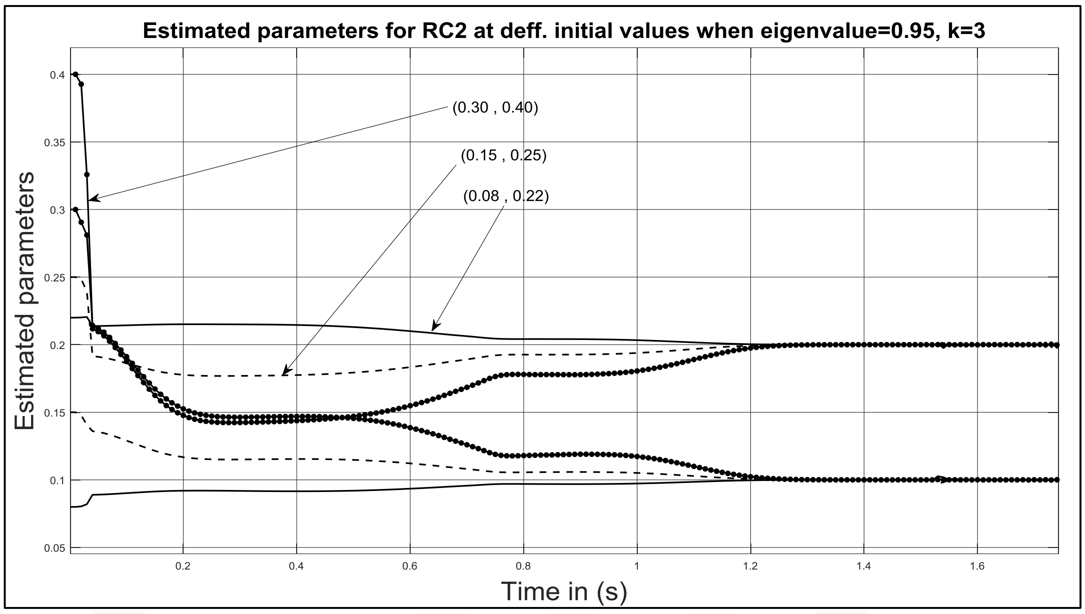


Figure 5.63: Identified Parameters for RC_2 Signal at $\lambda = 0.95$ and k=3.

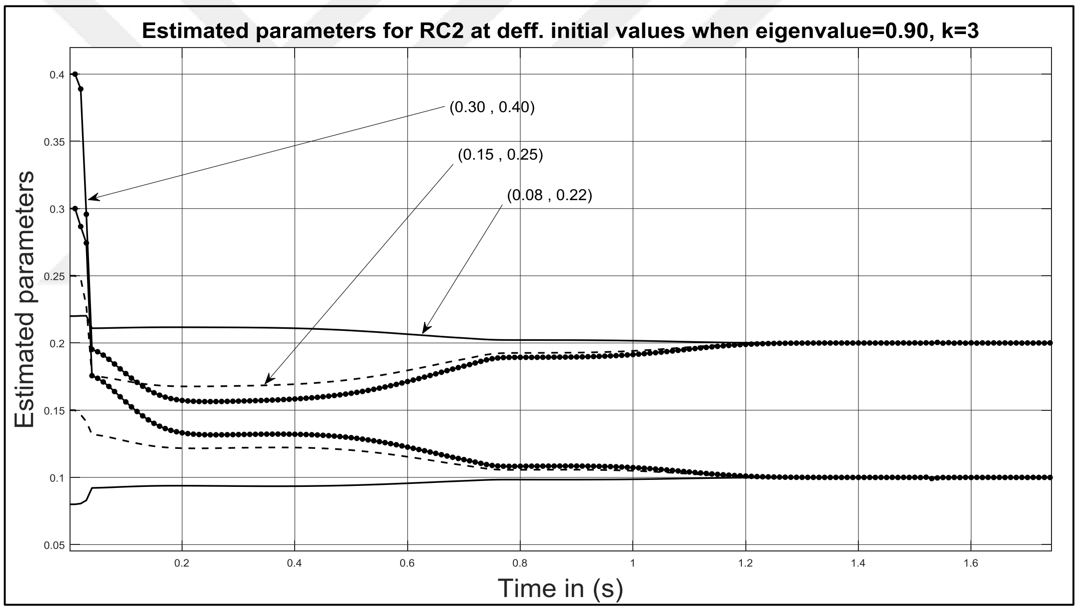


Figure 5.64: Identified Parameters for RC_2 Signal at $\lambda = 0.90$ and k=3.

5.4 Result Discussion

The reference signals RC_1 and RC_2 have tracked by ILC algorithm when the dynamic system has no unknown parameters as shown in Figure 5.3 and Figure 5.4, the progress of tracking with the iteration number for both reference signals are illustrated in Figure 5.5 and Figure 5.6.

The comparison between ILC and AILC has shown in Figure 5.9 and Figure 5.10, where they show the weak point of ILC and how it failed to control the dynamic system when has unknown parameters. From the results of AILC simulation it is obvious that the algorithm has succeeded to track what ILC failed on, and has adapted itself by identifying the new parameters to track the same reference signals and the same dynamic system with unknown parameters and at difference eigenvalues as seen in Figure 5.12, Figure 5.20 and Figure 5.28 for RC_1 , and in Figure 5.36, Figure 5.44 and Figure 5.52 for RC_2 .

In some conditions AILC algorithm shows more efficient results than other ones, in some conditions the algorithm does not need many iterations to reach to zero error, in the graphs shown in Figure 5.13, Figure 5.21 and Figure 5.29, AILC examined under RC_1 signal for three numbers of iterations k (1, 5 and 10), and for different eigenvalues (0.99, 0.95 and 0.90), the results were, as the eigenvalue of the learning gain matrix Γ is low as the controller AILC needs less iteration number to reach zero error. The same results have implemented when RC_2 is used, see Figure 5.37, Figure 5.45 and Figure 5.53.

The porous of the control system is to track the position of the dynamic system, so it is important to get the zero error in distance x tracking. The results of the distance error e_d are illustrated under the same circumstances and for the two reference signals RC_1 and RC_2 as well, see Figure 5.14, Figure 5.22, Figure 5.30, for RC_1, and Figure 5.38, Figure 5.46 and Figure 5.54 for RC_2 , where the error will reach to zero at the small eigenvalues faster than the big ones.

By considering the results of velocity error e_v for RC_1 shown in Figure 5.18, Figure 5.26 and Figure 5.34, and for RC_2 shown in Figure 5.42, Figure 5.50 and Figure 5.58, it is obvious that the velocity error doesn't reach to zero, this error is due to several reasons, one of them is the discretization process which has its own errors especially when a small number of terms have taken in discretization, in this simulation just two terms have taken in calculation for simplicity and faster calculation. The second reason is the derivation. Since the velocity comes from the derivation of the distance, the distance errors will be doubled and appear as we have seen in the last figures, and also the errors in the identified parameters are another reason for that velocity errors.

The identificator has the ability to adapt the unknown parameters even when they have different initial values. In the simulation, the AILC have simulated at initial parameters $\hat{P}_0 = [0.08 \quad 0.22]^T$, then the ability of the identificator have been examined by another pairs $[0.30 \quad 0.40]^T$ and $[0.15 \quad 0.25]^T$ and the results as seen in Figure 5.59, Figure 5.60, Figure 5.61, Figure 5.62, Figure 5.63, and Figure 5.64.

Simulation results comparison is summarized in Table 5-1 below:

Table 5-1: Simulation Percentages of AILC Algorithm.

RC	λ	Iterations (k)	$\frac{max(e_p)}{max(RC)}\%$	$\frac{max(e_v)}{max(\dot{RC})}\%$	$\frac{max(e_i)}{max(\hat{\zeta})}\%$	$\frac{max(e_i)}{max(\hat{\omega}_n)}\%$
RC_1	0.99	1	45.4026	70.6933	0.00001	0.00001
		5	0.0395	31.2582	0.0407	0.0159
		10	0.0072	15.8328	0.0125	0.0050
	0.95	1	0.0198	30.7313	0.0459	0.0195
		5	0.000534	7.4695	0.00001	0.00001
		10	0.0001625	7.3999	0.00001	0.00001
	0.90	1	0.0093	15.0273	0.0040	0.0016
		5	0.00015918	7.3999	0.00001	0.00001
		10	0.00010292	7.3999	2.2115	0.000001
RC_2	0.99	1	220.4697	32.2570	0.00001	0.00001
		5	0.0694	71.8004	0.00001	0.00001
		10	0.0043	17.7502	0.00001	0.00001
	0.95	1	0.0573	17.7933	0.00001	0.00001
		5	0.0090	17.7502	0.3302	0.1268
		10	0.000218	17.7502	0.00001	0.00001
	0.90	1	0.0134	17.7502	0.0278	0.0161
		5	0.000206	17.7502	0.00001	0.00001
		10	0.000655	17.7502	0.6002	0.2457

Where: e_p: Position error, the difference between the position of the dynamic system and the reference signal $X - RC$.

e_v: Velocity error, the difference between the velocity of the dynamic system and the velocity of the reference signal $V - \dot{RC}$.

e_i: The difference between the output of the dynamic system and the identificator.

CHAPTER SIX

CONCLUSION AND RECOMMENDATIONS

In this chapter, the conclusion of the thesis is illustrated, and what kind of ideas could be helpful for future researchers to be followed them.

6.1 Conclusion

From the illustrated results in this thesis, it has been considered that AILC algorithm succeeded to control the SISO time-invariant dynamic system within the illustrated operating conditions. AILC algorithm succeeded in overcoming the obstacles that ILC failed in, where AILC has controlled the dynamic system with unknown parameters and get the same results that ILC algorithm got on the same dynamic system and in the same conditions before its parameters have changed. The explained learning gain matrix was able to adapt the new parameters and change these values in the system in order to correct the errors in signal tracking. It has been clarified that convergence speed of the controller is directly proportional to the speed of response, and inversely proportional to the iteration number, the faster the controller convergence speed, the faster response and fewer iterations number. Strength and weakness points of AILC algorithm have described and classified in the result discussion in terms of controller convergence speed, iteration number, and the shape of reference signals.

6.2 Recommendations

1. Iterative learning control algorithm (ILC) cannot obtain the stability of unstable systems without merging with one of the conventional control theories, so, this algorithm can be edited to be applicable for unstable

repetitive dynamic systems have unknown parameters by merging it with one of the conventional control theories to it.

2. Developing AILC to be able to control time-varying systems.
3. AILC algorithm could be developed to be applicable to nonlinear continuous systems by using integral equations.
4. Checking the convergence ability of the parameters estimation problem and controlling it with the Lyapunov method.
5. Developing the algorithm to control dynamic systems that have robustness problem and stochastic noise.

REFERENCES

[1] K. Ogata, Modern Control Engineering (Fifth Edition), 2010.

[2] H.-S. Ahn, Y. Q. Chen and K. L. Moore, "Iterative Learning Control: Brief Survey and Categorization," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS,* pp. VOL. 37, NO. 6, November 2007.

[3] J. Andersson, *A framework for evaluation of iterative learning control,* Linköping: Linköpings universitet, 2014.

[4] K. L. Moore, "An Introduction to Iterative Learning Control Theory," in *Kevin L. Moore, EGES 504/604A Seminar, Colorado School of Mines, January 24, 2006*, Colorado , 2006.

[5] S. Ashraf, E. Muhammad and A. Al-Habaibeh, *Self-learning control systems using identification-based adaptive iterative learning controller.,* Proceedings of the I MECH E Part C: Journal of Mechanical Engineering Science, 222, 1177–1187., 2008.

[6] Z. B. a. K. M. Huh, "“Higher-order iterative control algorithm,”," *IEEE Proc. Part D, Control Theory Appl.,,* vol. vol. 136, pp. no. 3, 1227, May 1989,.

[7] M. Uchiyama, "(Formation of High-Speed Motion Pattern of Mechanical Arm by Trail)," *Transactions of the Society of Instrument and Control Engineers,,* pp. vol. 14, issue 6, pp., 1978.

[8] S. Arimoto, S. Kawamura and F. Miyazaki, "Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems," in *23rd Conf. Decision Control*, Las Vegas, Dec.1984.

[9] Z. Bien and J.-X. Xu, Iterative Learning Control Analysis, Design, Integration and Applications, New York: Kluwer Academic Publishers, 1998.

[10] J.-X. Xu and Y. Tan, Linear and Nonlinear Iterative Learning Control, Springer, March 3, 2003.

[11] D. H. Owens, Iterative Learning Control An Optimization Paradigm, London : Springer, 2016.

[12] H. Stearns, B. Fine and M. Tomizuka, *Iterative Identification of Feedforward Controllers for Iterative Learning Control,* California, USA: IFAC Proceedings Volumes (IFAC-PapersOnline), pp. 203-208, 2009.

[13] Y. Chen and C. Wen, *Lecture Notes in Control and Information Sciences,* Springer, UK, 1999.

[14] D. A. Bristow, M. Tharayil and A. G. Alleyne, *A survey of iterative learning control,* Control Systems Magazine, IEEE, 2006.

[15] W. Messner, R. Horowitz, W. W. Kao and M. Boals, *A new adaptive learning rule,* Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on, 1522–1527 vol.3, 1990.

[16] M. French and E. Rogers, *Nonlinear iterative learning by an adaptive Lyapunov technique,* Proc. 37th IEEE Conf. Decision Control, Tampa, FL, pp. 175–180, Dec.1998.

[17] Z. Bien and J. X. Xu, *Iterative learning control analysis, design, integration and applications,* Kluwer Academic Publishers,Norwell, MA, USA, 1998.

[18] D. H. Owens and G. S. Munde, *Universal Adaptive Iterative Learning Control,* Tampa, Florida USA: Proceedings of the 37th IEEE Conference on Decision & Control, Dec 1998.

[19] M. French, G. Munde, G. Rogers and D. H. Owens, *Recent developments in adaptive iterative learning control,* Phoenix, Arizona USA: Proceedings of the 38* Conference on Decision & Control, December 1999.

[20] M. Q. Phan and J. A. Frueh, *Model Reference Adaptive Learning Control with basis functions,* Phoenix, AZ: Proc. 38th IEEE Conf. Decision Control, Dec. 1999, pp. 251–257, Dec. 1999.

[21] J. Y. Choi and J. S. Lee, *Adaptive iterative learning control of uncertain robotic systems,* 2000.

[22] D. DeRoover and O. H. Bosgra, *Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system,* International Journal of Control, vol. 73, pp. 968-979, 2000.

[23] S. Gunnarsson and M. Norrlöf, *On the design of ILC algorithms using optimization,* Automatica, 37(12), 2011-2016, 2001.

[24] S. C. Lee, R. W. Longman and M. Q. Phan, *Direct model reference learning and repetitive control,* Intell. Autom. Soft Comput. Vol. 8, no. 2, pp. 143-161, 2002.

[25] H. Chen and P. Jiang, *Adaptive iterative feedback control for nonlinear system with unknown high-frequency gain,* Proc. 4th World Congr. Intell. Control Autom. pp. 847–851, Jun. 2002.

[26] S. Yang, X. Fan and A. Luo, *Adaptive robust iterative learning control for uncertain robotic systems,* Proc. 4thWorld Congr. Intell. Control Autom, pp. 964-968, 2002.

[27] D. H. Owens, K. Feng and , *Parameter Optimization in Iterative Learning Control,* International Journal of Control, 2003.

[28] Y. Miyasato, *Iterative learning control of robotic manipulators by hybrid adaptation schemes,* Japan: Proc. 42nd IEEE Conf.Decision Control, pp. 4428–4433, Dec. 2003.

[29] J. Shou, Z. Zhang and D. Pi, *On the convergence of open-closed-loop D-type iterative learning control for nonlinear systems,* Houston, TX: Proc. IEEE Int. Symp. Intell. Control, pp. 963-967, Oct, 2003.

[30] C.-J. Chien and C.-Y. Yao, *Iterative learning of model reference adaptive controller for uncertain nonlinear systems with only output measurement,* Automatica 40, page 855–864, 2004.

[31] C.-J. Chien and C.-Y. Yao, *An output-based adaptive iterative learning controller for high relative degree uncertain linear systems,* Automatica vol.40, pp. 145-153, 2004.

[32] S. Dong and K. M. James, *High-accuracy trajectory tracking of industrial robot manipulator using adaptive-learning scheme,* Canada: American Control Conference, pages 1935-1939, June, 2009.

[33] X. Bu, Z. Hou and R. Chi, *Model Free Adaptive Iterative Learning Control for Farm Vehicle Path Tracking,* IFAC Proceedings Volumes, Vol. 46, pp. 153-158., 2013.

[34] S.-K. Oh and J. M. Lee, *Stochastic iterative learning control for discrete linear time-invariant system with batch-varying reference trajectories,* Journal of Process Control, Vol. 36, pp. 64-78., 2015.

[35] J. Li and . J. Li, *Coordination control of multi-agent systems with second-order nonlinear dynamics using fully distributed adaptive iterative learning,* Journal of the Franklin Institute, vol. 352, pp. 2441-2463, 2015.

[36] X. Jin, *Adaptive iterative learning control for high-order nonlinear multi-agent systems consensus tracking,* Systems & Control Letters 89 16–23, 2016.

[37] Q. Yu, Z. Hou and R. Chi, *Adaptive Iterative Learning Control for Nonlinear Uncertain Systems with Both State and Input Constraints,* Journal of the Franklin Institute, vol. 353, Issue 15, pp. 3920-3943, 2016.

[38] J. Wei, Y. Zhang, M. Sun and B. Geng, *Adaptive iterative learning control of a class of nonlinear time-delay systems with unknown backlash-like hysteresis input and control direction,* ISATransactions, 2017.

[39] K. L. Moore, "Iterative learning control: A survey and new results," *Journal of Robotic Systems,* vol. 9, no. 5, pp. 563-594, 1992.

[40] K. L. Moore, Iterative Learning Control for Deterministic Systems, New York: Springer-Verlag, 1993.

[41] K. L. Moore, "History of Learning Control and Introduction of Professor Arimoto," in *Symposium on Learning Control at the IEEE CDC 2009*, Shanghai, China,, 14-15 December 2009.

[42] D. H. Owens, *The benefits of prediction in learning control algorithms,* London: Inst. Elect. Eng., 1999.

[43] P. B. Goldsmith, "On the equivalence of causal LTI iterative learning control and feedback control," *Automatica, vol. 38, no. 4,* pp. pp. 703–708,, 2002.

[44] M. Togai and O. Yamano, *Analysis And Design of an Optimal Learning Control Scheme for Industrial Robots: A Discrete System Approach.,* 1985.

[45] N. Amann and D. H. Owens, "Iterative learning control for discrete-time system using optimal feedback and feedforward actions," in *Proceedings of the 34th Conference on Decision and Control*, New Orleans, July 1995.

[46] N. Amann, H. Owens and E. Rogers, "Iterative learning control for discrete-time system with exponential rate of convergence," in *IEEE Proceeding-Control Theory and Applications*, March 1996.

[47] T. a. K. E. Mita, "ITERATIVE CONTROL AND ITS APPLICATION TO MOTION CONTROL OF ROBOT ARM - A DIRECT APPROACH TO SERVO-PROBLEMS," in *Proceedings of 24th Conference on Decision and Contro, pages 1393-1398*, December 1985.

[48] H.-s. L. S.-h. a. K. D.-h. Ahn, "Frequency-domain design of iterative learning controllers for feedback systems," in *Industrial Electronics, 1995. ISIE '95., Proceedings of the IEEE International Symposium on*, Athens, Greece, 10-14 July 1995.

[49] A. P. G. a. U. G. De Luca, "A frequency-domain approach to learning control: implementation for a robot manipulator," *Industrial Electronics, IEEE Transactions on,* vol. 39, no. 1, pp. 1-10, 1992.

[50] L. Hideg and R. Judd, "Frequency domain analysis of learning systems," in *Proceedings of the 27th Conference on Decision and Control, pages 586-591,*, Austin, Texas, December, 1988.

[51] R. P. Judd, R. P. Van Til and L. Hideg, "Equivalent Lyapunov and frequency domain stability conditions for iterative learning control systems," in *Proceedings of 8th IEEE International Symposium on Intelligent Control, pages 487-492*, 1993.

[52] R. J. Harris, *A Primer of Multivariate Statistics,* 3rd ed., 2017.

[53] J. L. Crassidis and J. L. Junkins, OPTIMAL ESTIMATION of DYNAMIC SYSTEMS, CHAPMAN & HALL/CRC Press Company, 2004.

[54] K. W. Iliff, *Parameter Estimation of Flight Vehicles,* NASA Ames Research Center, Edwards, California: Journal of Guidance, Vol. 12, Issue 5, 1989.

[55] R. C. Dorf and R. H. Bishop, Modern Control Systems 12th edition, PEARSON, 2011.

[56] K. Ogata, Discrete-Time Systems 2nd edition, 1995.

[57] A. Madady, "PID Type Iterative Learning Control with Optimal Gains," *International Journal of Control, Automation, and Systems,* Vols. vol. 6, no. 2, pp. pp. 194-203, April 2008.

[58] M. Norrlöf, *Iterative Learning Control: Analysis , Design , and Experiments,* Linköping, Sweden, 2000.

[59] J. E. Kurek and M. B. Zaremba, *Iterative learning control synthesis based on 2-D system theory,* IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 38, NO. 1, , JANUARY 1993.

[60] A. Latif, "Banach Contraction Principle and Its Generalizations," in *Topics in Fixed Point Theory*, S. Almezel, Q. H. Ansari and M. A. Khamsi, Eds., New York, Springer, 2014, pp. 33-64.

[61] C. R. J. Roger A. Horn, Topics in Matrix Analysis, Cambridge University Press, 1991.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Alzaidi, Ibrahim
Nationality: Iraqi (IQ)
Date and Place of Birth: 27 April 1980
Phone: +90 534 720 53 31
Email: Ibrahim.f.taha@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
MSc	University of Turkish Aeronautical Association	2017
BSc	University of Technology\ Baghdad	2005
High School	Al-Aadhamia High School	2000

WORK EXPERIENCE

Year	Place	Enrollment
2007-Present	Mosul Gas Turbine Power Station	Chief Engineer
2006	General Directory of Electric Power Transmission Project	Asst. Engineer

FOREIGN LANGUAGES

Arabic	:	Fluent
English	:	Very good
Turkish	:	Good