2004

Experimental evaluation of some classical and adaptive iterative learning control schemes on a 5DOF robot manipulator

Islam, Shafiqul

Experimental Evaluation of Some Classical and Adaptive Iterative Learning Control Schemes on a 5DOF Robot Manipulator

Shafiqul Islam

Under the Supervision of

Dr. Abdelhamid Tayebi

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF MASTERS OF SCIENCE DEGREE IN CONTROL ENGINEERING

FACULTY OF ENGINEERING
LAKEHEAD UNIVERSITY
THUNDER BAY, ONTARIO, CANADA

July 29, 2004

Your file Votre référence
ISBN: 0-494-00495-9
Our file Notre référence
ISBN: 0-494-00495-9

Contents

List of Figures

Chapter 1

Introduction and Literature overview

1.1 Introduction

In many process industries (*e.g.*, VLSI production lines, Automotive industries, IC welding process, inspections, manipulations), robot manipulators are used to perform the same tasks repeatedly over a finite time interval. The ultimate goal of robotic research is to design intelligent and autonomous robot control systems to perform repetitive tasks that are dull, hazardous, or require skill beyond the capability of humans. The nonlinear nature of the robot dynamics has made this problem a challenging one in robotics research. This highly demanding control problem of driving an industrial robot to follow a desired trajectory perfectly under constrained or unconstrained environment has led to the application of sophisticated control techniques. From the classical or modern control view point, it is a very difficult task to design an intelligent robot control system that can achieve perfect tracking over a finite time interval due to the effect of highly coupled robot dynamics and the presence of the unmodeled dynamics such as friction and backlash that are usually exhibited in the robot system during actual operation.

As robot manipulators are generally used to perform repetitive tasks in many industrial processes, researchers have been motivated to investigate new types of control techniques that can utilize the repetitive nature of the robot dynamics and improve the tracking performance from iteration to iteration. This new technique is known as iterative learning control (ILC). Specifically, ILC aims to enhance the feedback control

performance by modifying the current control input using the previous control information. Compared with other existing control methods, ILC is simpler in structure and easier to implement in the real world operation as it requires less computational efforts and less *a priori* knowledge of system dynamics. Since early works Arimoto *et al.* (1984), Casalino and Bartolini (1984) and Craig (1984), various ILC algorithms for robot manipulators have been studied in the past two decades that can be used for improving the tracking performance by compensating the unknown parametric uncertainties or unmodeled dynamics from operation to operation through memory based learning techniques.

1.2 Literature review

In the last two decades, several control strategies for uncertain robot manipulators have been presented in the literature, see for instance (Takegaki and Arimoto 1981; Arimoto and Miyazaki 1984; Ortega and Spong 1988; Khorrami and Ozguner 1988; Sadegh and Horowitz 1990; Tomei 1991; De Luca and Siciliano 1992; Wen and Murphy 1990; Slotline and Li 1987 and 1991; Ge 1998). Among these designs, PD and PID controllers have attracted the attention of the control community for their simplicity and performance. In fact, a PD controller with gravity compensation is able to asymptotically stabilize rigid robot manipulators around a given joint configuration. In practice, however, the exact knowledge of gravitational forces is a very difficult task because the payloads manipulated by the robot during the execution of a given task are generally varying. As a result, a steady state error is always present with this control technique. An alternative solution to this problem is to use high-gain feedback that dominates over the gradient of the gravity forces in the whole robot work space. In fact, the high-gain feedback solution reduces the error but does not eliminate the error completely. The major drawback of using high-gain feedback is that it may saturate the actuators and excite the unmodeled dynamics. The existing steady state error in PD controller can be eliminated by adding an integral term. But several problems may arise with the design of a PID controller due to the nonlinear nature of the robot manipulators. Moreover, from a theoretical point of view, a PID controller is able to locally asymptotically stabilize the joint positions of rigid robot manipulators around the desired configuration under some complex inequalities

among the proportional, derivative and integral gains (Arimoto *et al.* 1996). In practice, the tuning of the PID gains is a very difficult task for nonlinear systems such as robot manipulators. In industrial applications (Wen 1990), these drawbacks can be compensated by using the integral action at the desired final configuration. In this framework, gross motion is performed with a PD control action and fine positioning can be achieved with PID action. This is a common strategy for industrial robot controllers, but there is no formal proof of the convergence of this method. Another interesting method (Slotine and Li 1987 and 1991) has been developed for trajectory tracking of robot manipulators instead of set-point regulation. This approach consists of a PD action plus an appropriate adaptation rule in order to compensate the gravity forces and the time-derivatives of the reference trajectory. However, this technique is valid only when the robot parameters are time-invariant.

As a matter of fact, the performance of the above mentioned controllers are considerably affected when the robot parameters are time-varying and unstructured uncertainties such as frictions and disturbances are involved. As robot tasks are repetitive in nature, researchers have been motivated to investigate a new type of control techniques that can exploit this property in order to enhance the tracking performance from iteration to iteration. The key feature of this control technique is the ability to compensate for the unmodeled dynamics and parametric uncertainties, that are mainly exhibited during the actual system operation, without using a high-gain feedback. In the past two decades, many ILC schemes for robot manipulators were reported in the literature (Mita and Kato 1985; Gu and Loh 1987; Bondi *et al.*1988; Kawamura *et al.* 1988; Moore *et al.* 1989 and 1990; Arimoto *et al.* 1990; Kuc *et al.* 1991; Horowitz *et al.* 1991; De Luca *et al.* 1992; Kavli 1992; Horowitz 1993; Moon *et al.* 1997; Norrlof and Gunnarsson 1997; Gunnarsson and Norrlof 1997; Moore 1998; Norrlof 2000; Wang 2000; Ye and Wang 2002). These ILC schemes are generally based upon the use of the contraction mapping approach and require a certain *a priori* knowledge of the system dynamics. On the other hand, in some ILC schemes the sup-norm or ∞-norm of the output tracking error can grow quite large before it converges to the desired level even if the convergence and stability conditions are satisfied. This is due to the fact that the learning convergence is based upon the use of the exponentially weighted sup-norm analysis. This transient behavior, which is a serious concern in practical applications of ILC schemes, can be improved by using the

exponentially decaying learning gain (Lee and Bien 1997). The convergence nature of classical ILC schemes for robot manipulators has been analyzed by Longman (1998) from a frequency domain perspective. In order to achieve a monotonic convergence of the sup-norm of the tracking error, one can use a high-gain feedback as discussed by Owen (1992). However, this is not a practical solution because high-gain feedback may saturate the actuator dynamics.

In the search of a robust approach to ILC design, Moon *et al.* (1998) developed a frequency-domain design method for uncertain linear systems. They showed that the ILC design problem can be reformulated as a general robust control problem and thus can be solved by well-established robust control design procedures such as μ-synthesis. Based on the same idea, Tayebi and Zaremba (2000, 2001 and 2003) proposed a robust ILC technique for uncertain systems based upon the use of the robust performance condition. They have shown that if the feedback controller can be designed such that it satisfies the well known robust performance condition, then an iterative updating rule can be generated by using the performance weighting function. However, the robust controller design is based on the nominal model of the plant that has to be controlled, and the performance will be seriously affected by parametric inaccuracies. Due to some difficulties in applying traditional ILC algorithms for certain systems, recently a new type of ILC algorithms that can handle broader class of nonlinear systems with parametric or non-parametric uncertainties have been widely studied. One of the most interesting and significant development in the recent years in ILC research is the so-called adaptive iterative learning control (AILC). This method is based upon the use of standard Lyapunov and Lyapunov-like energy functions. In the past decade, several adaptive ILC schemes have been reported in the literature (Park *et al.* 1996 and 1998; Kaneko and Horowtiz 1997; French and Rogers 1998; Seo *et al.* 1999; Kuc *et al.* 2000; Kuc and Han 2000; Choi and Lee 2000; Ham *et al.* 2000 and 2001; Xu *et al.* 2000; Xu and Tan 2001; Xu 2002; Hsu *et al.* 2003). The main idea behind the adaptive ILC approach is to estimate the uncertain system parameters and disturbances in order to generate a current control input that can improve the output tracking performance from trial to trial. Most adaptation laws for uncertain parameter estimations are designed in the iteration-domain because of the iteration based control problem (Park *et al.* 1996 and 1998; Kuc *et al.* 2000; Xu and Wiswanathan *et al.* 2000). In this case, the projection or dead zone mechanism

is required to guarantee the stability and convergence of the learning process. Among the AILC techniques mentioned above, Park *et al.* (1996) proposed an adaptive iterative learning controller for uncertain robot manipulators based upon the use of a Lyapunov function. Xu and Qu (1998) utilize a Lyapunov-based energy function to illustrate how an ILC can be combined with a variable structure controller in order to handle a broader class of nonlinear systems. It was shown that a nonlinear feedback controller can be combined with a learning control scheme to achieve asymptotic convergence for a class of nonlinear systems. Xu *et al.* (2000) have shown that this nonlinear scheme can be extended to a broader class of nonlinear systems with both periodic and non-periodic uncertainties by introducing a sliding mode into the learning control techniques. In (Ham et al. 2000), a Lyapunov-based technique is utilized to develop an ILC scheme that is combined with a robust control design. This design was extended to a broader class of non-linear systems in Ham *et al.* (2001). In (Choi and Lee 2000), an AILC scheme for robot manipulators with both time-domain and iteration-domain adaptations was proposed. However, this design requires the unknown system parameters to be constant.

Xu and Tan (2001) proposed an AILC algorithm based upon the use of a positive definite Lyapunov-like energy function which is made monotonically decreasing along the iteration axis via a suitable choice of the control input. In this framework, the control scheme can learn iteration-independent time-varying uncertainties. In (Hsu *et al.* 2003), a new adaptive ILC scheme was developed for uncertain robot manipulators. The proof of convergence of this AILC algorithm is based on a Lyapunov energy function. As a matter of fact, the above mentioned adaptive ILC algorithms for robot manipulators require a certain *a priori* knowledge of system dynamics.

Most recently, some interesting AILC schemes have been proposed for the position tracking control of rigid robot manipulators without any *a priori* knowledge of the system dynamics in Tayebi (2004). This new generation of adaptive schemes can handle a broader class of parametric or nonparametric uncertainties, which opens a new avenue for the learning control design. The control schemes are build around a classical PD feedback control structure, for which an iterative term is added in order to ensure asymptotic convergence along the iteration axis. The proof of convergence is based upon the use of a composite energy function, which is made monotonically decreasing through an adequate choice of the control law and the iterative adaptation

rule. The implementation results of these AILC can be found in Tayebi and Islam (2004). In practice, the velocity signals in the adaptation law are not measured but estimated using the so-called "dirty derivative" which generates considerable amount of noise and this noise amplifies from iteration to iteration. A potential solution to this crucial practical problem is to design AILC schemes that are based upon the use of joint position signals only.

1.3 Objectives

The main objective of this thesis is to test experimentally various classical and adaptive ILC approaches on a 5-DOF industrial robot manipulator. The thesis also gives a comparison of these implementation results by observing the learning ability of the industrial robot manipulator executing the same desired trajectory repetitively over a finite time interval.

The thesis is divided into eight chapters. Chapter 2 gives the ILC research background, main idea of the ILC method as well as its introductory examples. A brief history of ILC methodology and its connection with other control paradigms will be covered in this chapter.

Some classical linear ILC schemes with their simulation results will be discussed in Chapter 3. Various classical ILC techniques for robot manipulators are presented in Chapter 4. Chapter 5 discusses some adaptive ILC schemes for uncertain nonlinear systems. The implementation results of these classical and adaptive ILC approaches on a 5-DOF CRS255 robot manipulator are presented in chapter 7.

Chapter 6 gives a brief discussion on the experimental platform that is used for the experimental evaluation of the ILC algorithms presented throughout the thesis. Finally, the conclusion and future work are given in Chapter 8.

Chapter 2

BACKGROUND

This chapter gives the main idea behind ILC method as well as a brief history of iterative learning control methodology and its connection with other control techniques.

2.1 Why ILC?

Most existing control methods, devised in the time domain, are based upon the use of a mathematical model of the controlled plant. In order to get a better performance, however, a more accurate model of the plant is required. In fact, the following problems can be encountered (Harris *et al.* 1993) during the modeling of a plant

a) The system is too complicated to understand or represent in a simple way.

b) The model is very difficult and/or expensive to evaluate.

c) It is not so easy to obtain the characteristics of some non-linear effects such as coulomb friction.

d) The plant parameters may be time-varying.

e) It is very difficult to predict or estimate the plant disturbances which may be large .

On the other hand, tracking tasks in most practical control applications have to be accomplished over a finite time interval. Generally, the scale of finite time interval can

processes. Generally, the term **Iterative** indicates a kind of action that needs the dynamic process to be repeatable, i.e., the dynamic system is deterministic and the tracking control tasks are repeatable over a finite time interval. The term **Learning** implies a gain of knowledge through the iterative process. Learning can be represented as a bridge between knowledge and experience, where insufficient knowledge is bridged by learning through repetitive practice. Now, the question is: How can we gather knowledge without dynamic process model? When a target trajectory is performed repeatedly, one can gain extra information from a new source such as past control input and output tracking error profiles, which can be viewed as a kind of "experience". This kind of "experience" serves as a new source of knowledge related to the dynamic process model, and reduces the need for the process model knowledge and computational efforts are less than any other existing control techniques. The new knowledge is learned from the previous control "experience" that provides the possibility of improving the tracking performance (see more on this issue from dedicated ILC home www.csois.usu.edu/ilc/).

2.3 Relation between ILC and other control techniques

In recent years, ILC has seen a growing interest in the control community for its simplicity and performance for systems that execute the same control task over and over. So, it is natural to know the main difference between ILC and other common control techniques. In most industrial applications, the performance of the process is generally improved by conventional control methods. Generally, the ILC method is combined with the conventional control method for further improvement of the control performance by utilizing the fact that the system is operated repeatedly.
It is well known in modern control engineering that if the system description is available, then the optimal solution can be generated by inverting the system description in order to produce a control input such that the system output follows the desired control task as close as possible. In this case, the controller performance totally depends on the system description. This method can be considered as a feedforward control scheme and applied successfully in robot control applications where it is called

inverse dynamics. If the system model, describing the mapping from input to output, is not totally known, then it is clear that the inverse dynamics technique will never achieve a perfect tracking of the desired control tasks.
Another well known method, namely identification, can be used if the system structure is known while the exact value of parameters is unknown. In fact, one can design a controller together with identification that can generate an appropriate control input to track the desired control task. This approach is referred to as adaptive control. It should be pointed out that the adaptive controller performance is good as long as the structure of the system is correct.
As a matter of fact, the ultimate goal of most existing control methods is to achieve exponential or asymptotic convergence in the time domain which is obviously inadequate if the objective is to achieve perfect tracking from the very beginning of the execution. ILC is a technique that can be applied to repetitive processes in order to get a perfect output tracking from the very beginning of the execution when the number of iterations goes to infinity. ILC differs from most existing control methods in the sense that it tries to utilize every possibility of the past control experience: the past tracking error profiles and control input signals in order to construct the current control input. This experience can be accumulated through memory based learning technique. Mainly, memory components are used to capture the past control information, and then the captured control information is retrieved to construct the feedforward part of the current control action.

2.4 Main idea

ILC is a relatively new control technique that can be used for improving the tracking performance and transient response of systems that execute the same control task repeatedly. The main idea behind this technique is to construct the optimal input in a recursive manner such that the system output can follow a desired control task perfectly from the very beginning of the execution. Consider the case where an input $U_k(t)$ is applied to a process that produces an output $Y_k(t)$ at the k-th iteration. The control input U_k and output trajectory Y_k are stored in the memory until the trial is over. The new control input U_{k+1}, where $k \in \{1, 2, 3.....\}$, is then updated based upon the use of the previous control information. In order to design successful ILC

algorithms, the following set of postulates are to be imposed:

(P1). The time duration of the target trajectory, T, is fixed for each operation.

(P2). The initial resetting condition is satisfied for each operation, i.e., $y_d(0) = y_k(0)$.

(P3). The input to the system U_k is stored along with the system output Y_k at each operation, and then past control information is retrieved to construct the current control input $U_{k+1}(t)$ for improving the tracking performance in the next operation.

A numerical example is the best way to make the concept of iterative learning control understandable. In the ILC design, the previous error and control input signals are mainly used to construct the current control input such that it can identify and compensate uncertain sources of system dynamics and improve the control performance from operation to operation.

Consider a desired control task, $\theta_d(t) = \sin 2\omega t$, over a finite time interval $[0, 20s]$ is given to a one DOF robot manipulator for perfect tracking which is shown in figure 2.1. A closed-loop transfer function of a single robot joint can be approximated as (Ye and Wang 2002)

$$P_r(s) = \frac{948}{S^2 + 42s + 948} \tag{2.1}$$

The main control objective is to design a learning controller such that the robot position trajectory $\theta_k(t)$ converges to the desired one $\theta_d(t)$, for all $t \in [0, 20s]$ when k tends to infinity. Consider the Laplace transform of the well-known ILC algorithm (Arimoto et al, 1984, Togai and Yamano 1985)

$$U_{(k+1)}(s) = U_k(s) + L(s)e_k(s) \tag{2.2}$$

where $L(s)$ is the learning compensator in Laplace form and $k \in \{1, 2, 3, ...\}$. A sufficient condition for the tracking error convergence is

$$\|1 - P_r(j\omega)L(j\omega)\| < 1 \tag{2.3}$$

This is a standard stability criterion for the ILC updating formula (2.2). The overall control scheme is depicted in figure 2.1. The simulation results, with three different values of $L(s)$, are presented in figure 2.2 and 2.3. From the simulation results, it is

range from milliseconds to months. Tracking over a finite-time interval implies the performance in transient process becomes more important. Often, perfect tracking performance is necessary from the very beginning of the system execution. Obviously, traditional control techniques designed in time domain are inadequate because they only ensure the performance at the steady state when the time goes to infinity.
When the system structure is unknown and perfect output tracking is required from the very beginning of the execution over a finite time interval, then ILC techniques can be applied for the repetitive process. The main advantage of ILC technique is that it requires less computational efforts and less *a priori* knowledge about the system dynamics as compared with other classical control methods. The ILC technique has the following key features:

1. A very simple structure - an integral action along the iteration-domain.
2. Output tracking control without using any *a priori* knowledge of the system state is the ultimate goal of ILC methods.
3. ILC differs from most existing control methods in the sense that it tries to capture and utilize the past control information completely in order to construct current control input. The previous control information can be realized through memory based learning. In this case, memory components are used to record the past control information: error profiles and control input signals over the entire time interval.
4. The current control input is usually generated by the combination of the error signals and input signals of the previous iteration.
5. The initial resetting condition, i.e. $y_d(0) - y_k(0) = 0$, plays an important role in the learning convergence of the iterative process.
6. The desired control tasks $y_d(t)$ must be identical for all repeated operations.

2.2 What is ILC?

ILC is a relatively new approach in system control engineering, aiming to improve the transient and tracking performance in the face of modeling uncertainty of repetitive

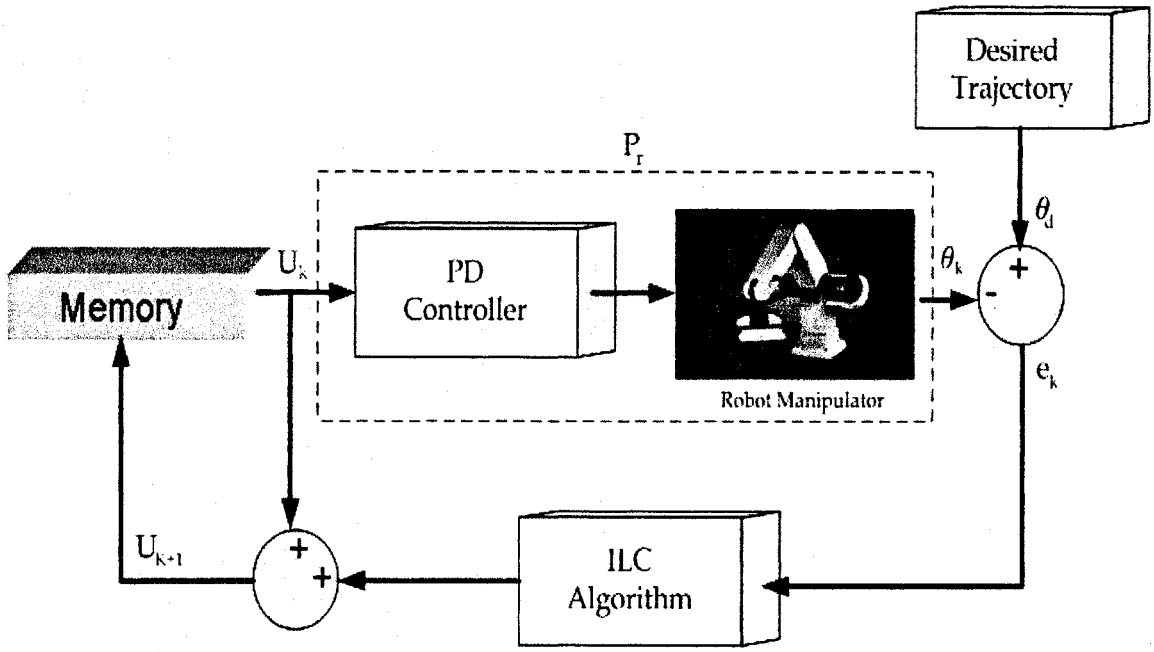

Figure 2.1: Example: Classical ILC structure for robot system

obvious that the tracking error can be reduced by using this learning technique (2.2). However, this algorithm shows good convergence only when the filter $L(s)$ satisfies equation (2.3).

2.5 Brief history of ILC

As robotic tasks are repetitive in nature, researchers have been motivated to investigate new control approaches that can use the repetitive nature of the robot dynamics in order to enhance the tracking performance iteratively. The idea of using information from the previous iteration was originally introduced by Uchiyama (1978) in order to improve the performance of robot motion by observing the tracking error from iteration to iteration. Since there was no specific details how the learning scheme improves the performance in repetitive trials, the ideas were not widely known until 1984. The first step to show the complete convergence proof of the learning process for the robot manipulators was proposed by Arimoto and his co-researcher (1984).

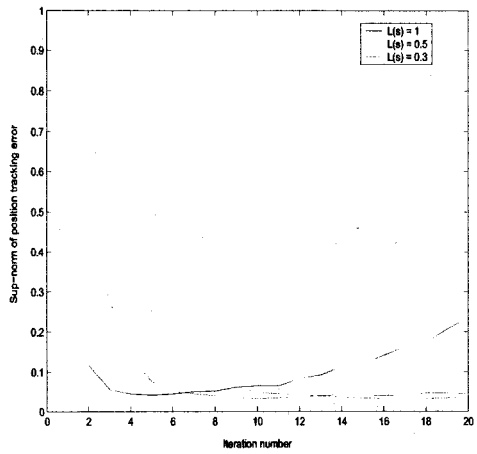

Figure 2.2: Sup-norm of the position tracking error versus the iteration number

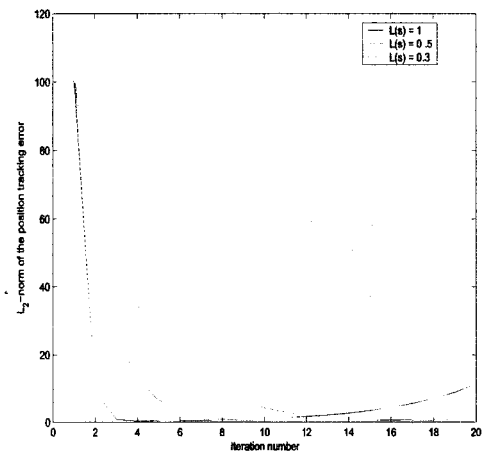

Figure 2.3: L_2-norm of the position tracking error versus the iteration number

The name which is called today iterative learning control was first presented as a betterment process (Arimoto *et al.* 1984). Casalino and Bartolini (1984) and Craig (1984), independently proposed ILC techniques that can also compensate for model errors and disturbances from operation to operation. Based on this idea, several ILC control schemes have been reported in the literature (Arimoto *et al.* 1985; Moore *et al.* 1989 and 1990; Chen *et al.* 1999; Chow and Fang 1988; Kurek and Zaremba 1993). These ILC schemes were developed as a feedforward control action applied directly to the open-loop system. The main drawback of the feedforward open-loop ILC control schemes is related to the fact that an inappropriate initial control effort may generate harmful effects for the system. To overcome this drawback, several feedback-based ILC schemes have been proposed in the literature (Kawamura *et al.* 1985; Atkeson and McIntyre 1986; Kuc *et al.* 1991; Jang *et al.* 1995; De Roover 1996; Moon *et al.* 1998; Doh *et al.* 1999; Arimoto *et al.* 2000).

Many contributions have been reported in the literature where ILC schemes are applied to robotic manipulators (Arimoto *et al.* 1985 and 1991; Mita and Kato 1985, Bondi *et al.* 1988; Kawamura *et al.* 1988; Bien *et al.* 1988; Moore *et al.* 1990; Poloni and Ulivi 1991; Horowitz *et al.* 1990, 1991 and 1993; Guglielmo and Sadegh 1996; Burdet *et al.* 1997; Moon *et al.* 1997; Norrlof and Gunnarsson 1997; Kazumasa *et al.* 1997; Kuc and Han 1999; Jiang *et al.* 1999; Lange and Hirzinger 1999; Norrlof

2000; Norrlof and Gunnarsson 2002; Kim *et al.* 2000; Wang 2000; Ye and Wang 2002). There are number of survey papers on ILC, see for instance (Moore 1993 and 1998 and Bien and Xu 1998). Detailed literature reviews and overall developments on ILC can be found in Moore (1999), Bien and XU (1998), Chen and Wen (1999). Due to some important implementation difficulties of classical ILC schemes, new types of ILC schemes have been extensively studied in the literature in the recent years. Motivated by Lyapunov direct method, the concept of energy function opens a new avenue for the ILC design. Various ILC algorithms based on Lyapunov energy function in the iteration-domain have been developed in Park *et al.* (1996 and 1998), Kuc *et al.* (2000), Xu *et al.* (2000), Chien *et al.*(2002). Choi and Lee and Chun-Te *et al.* (2003) developed another adaptive iterative learning control scheme with both time-domain and iteration-domain adaptation laws for uncertain robot manipulators. Most recently, the composite energy function (CEF), has been used to design AILC schemes for uncertain robot manipulators in Tayebi (2004). Implementation results of these adaptive ILC schemes for an industrial robotic system are studied by Tayebi and Islam (2004).

Chapter 3

CLASSICAL ILC FOR LINEAR SYSTEMS

A large part of the ILC research has been focused on the design of learning algorithms for linear systems. The basic difference of various ILC approaches is in how the error is used to construct the learning control schemes. The purpose of this approach is to find a suitable learning rule that allows the controller to learn from the previous operation, and improve the tracking performance from operation to operation. In this chapter, some classical linear ILC schemes will be discussed and simulation results will be presented.

3.1 P-type ILC scheme

The most general linear learning algorithm can be expressed as

$$u_{k+1}(t) = u_k(t) + \Gamma e_k(t) \tag{3.1}$$

where

$$e_k(t) = y_d(t) - y_k(t). \tag{3.2}$$

This is an open loop learning scheme and the excitation of the system at $k = 0$ has to be chosen arbitrarily. During the $(k+1)$-th iteration, the control input is updated in such a way that the previous control input at the k-th trial is refined by adding a correcting term depending on the tracking error at the k-th iteration. Thus, the combination of the previous control input and error data pair generates the current learning control input that allows the system to learn a desired task and improves

the tracking and transient performance from iteration to iteration. For the following linear system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{3.3}$$

if the initial resetting condition i.e., $y_d(0) - y_k(0) = \dot{y}_d(0) - \dot{y}_k(0) = 0$ and $\|I - \Gamma CB\| < 1 \ \forall k \in \mathbb{N}$ are satisfied then the P-type learning process is monotonically convergent (Arimoto *et al.* 1984) in the sense that

$$\|e_{k+1}\|_\Gamma \leq \|e_k\|_\Gamma$$

and $\|e_k\| \rightarrow 0$ when k goes to infinity, where

$$\|e\|_\Gamma^2 = \int_0^T e^T(t)\Gamma e(t)dt$$

$$\|e\| = \left[\int_0^T e^T(t)e(t)dt\right]^{\frac{1}{2}}$$

and Γ is a positive definite matrix. Therefore, one can conclude that the error between $y_d(t)$ and $y_k(t)$ approaches zero as k tends to infinity. In order to show the learning ability of the P-type ILC algorithm, let us consider the following simple linear system

$$\ddot{q}_k + 2\dot{q}_k + q_k = \tau_k(t) \tag{3.4}$$

The control objective is to track the following reference trajectory over a finite time interval $[0, 10s]$

$$q_d(t) = \sin 2\omega t. \tag{3.5}$$

where $\omega = 1$. By applying the P-type learning scheme to the system (3.4), the obtained results are shown in figures 3.1, 3.2 and 3.3. Figure 3.1 and 3.3 illustrate the evolution of the L_2-norm and sup-norm of the tracking error with respect to the number of iterations. Figure 3.2 shows the time-evolution of the desired trajectory $q_d(t)$ and the actual tracking trajectory $q_k(t)$.

3.1.1 Observation

- The P-type ILC algorithm is simpler in structure and easier to implement than other linear ILC algorithms. However, it is not robust against initial state or

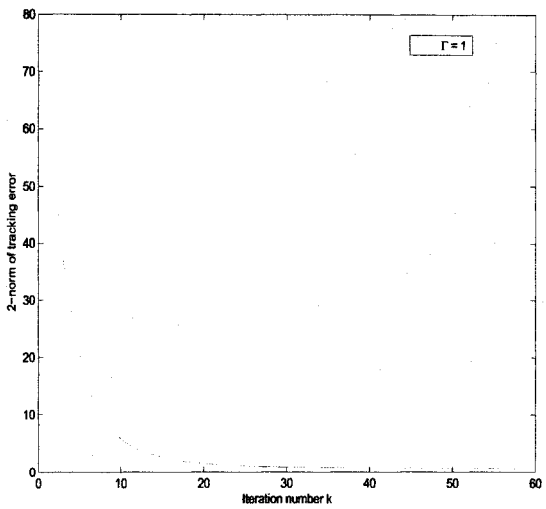

Figure 3.1: L_2-norm of the velocity tracking error versus the number of iterations under P-type ILC

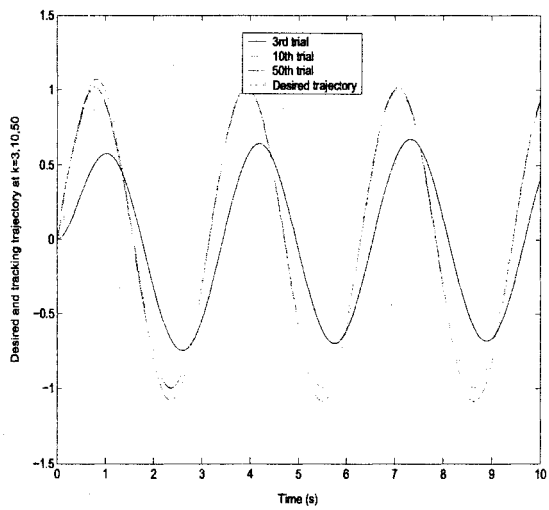

Figure 3.2: Desired trajectory $q_d(t)$ and the output $q_k(t)$ for k=3, 10 and 50 with $\Gamma = 1$ under P-type ILC

output error. It can be easily seen that, when $y_d(0) - y_k(0) \neq 0 \ \forall k \geq 0$, then the P-type algorithm can result in $u_k(0) \rightarrow \infty$ as iteration goes to infinity. In order to design robust P-type ILC schemes against uncertainties, one can add a scalar forgetting factor in (3.1) (Arimoto 1990, Saab 1994, Chien and Liu 1996). In this framework, one can show that the tracking error will be bounded under various assumptions and strict conditions, which is usually very difficult to meet in the real operation. In fact, Chien and Liu (1996) have shown that the bounds on the tracking errors are inversely proportional to the forgetting factor. This means that the smaller the forgetting factor, the larger is the tracking error bound.

- The P-type learning law cannot capture proper direction of errors that occur in the previous operations. To the best of one's knowledge, the convergence results of the P-type ILC schemes can be found in Saab (1994), Chien and Liu (1996) and Arimoto *et al.* (1990). As a matter of fact, these learning schemes showed limited success in proving theoretically the effectiveness for generalized continuous dynamic systems. However, P-type ILC is able to ensure the convergence of the tracking errors only without considering the initial state

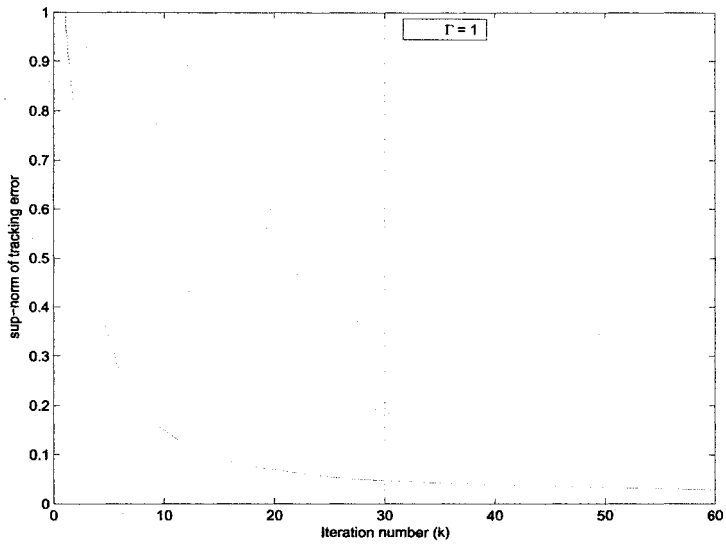

Figure 3.3: Sup-norm of the tracking error versus the number of iterations under P-type ILC

error, uncertainties and disturbances.

3.2 D-type ILC for linear LTI systems

The first and earliest ILC algorithm is the derivative type (Arimoto *et al.* 1984) which has the following form

$$u_{k+1}(t) = u_k(t) + \Gamma \dot{e}_k(t) \tag{3.6}$$

where Γ is a constant learning gain. Given an arbitrary control input, $u_k(t)$, that excites the system at the k-th trial and corresponding system output response, $y_k(t)$, then the error $e_k(t) = y_d(t) - y_k(t)$ may arise. The learning input $u_{k+1}(t)$ is constructed on the basis of the previous control input profiles $u_k(t)$ and derivative of the output tracking error profiles.

Consider the following LTI system:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{3.7}$$

where $x \in \mathbb{R}^n$ $u \in \mathbb{R}^r$, $y \in \mathbb{R}^r$denote state, input and output respectively. A, B and C are matrices with appropriate dimensions with $CB \neq 0$. Consider, $x_d(t)$ and

$y_d(t)$ as the desired state trajectory and the desired output trajectory, respectively. The aim is to find a recursive control law guaranteeing the boundedness of state $x_k(t)$, $\forall t \in [0,T]$ and $\forall k \in \mathbb{N}$, and the convergence of $x_k(t)$ to the desired state $x_d(t)$ over a finite interval [0,T] when k tends to infinity. If the following condition and assumptions are satisfied, then the D-type iterative learning process is convergent and the state trajectory $x_k(t)$ tends to the desired one $x_d(t)$ for all $t \in [0,T]$, when $k \rightarrow \infty$.

Condition

A. The induced operator norm $\|I - \Gamma CB\|$ has to meet the following condition

$$\|I - \Gamma CB\| < 1 \tag{3.8}$$

Assumptions

B. It is assumed that the resetting condition at each iteration is satisfied, i.e., $y_d(0) - y_k(0) = \dot{y}_d(0) - \dot{y}_k(0) = 0$, $\forall k \in \mathbb{N}$

and

C. $u_0(t)$ and $y_d(t)$ are continuous and continuously differentiable, respectively, namely $u_0(t) \in C[0,T]$ and $y_d(t) \in C^1[0,T]$

As a matter of fact, the D-type ILC scheme improves the tracking performance of the system from operation to operation in the sense that the λ-norm or the time-weighted norm of the tracking error, i.e.,

$$\|e_k\|_\lambda = \sup_{0 \le t \le T} \{e^{-\lambda t} \|e_k\|\}$$

tends to zero when $k \rightarrow \infty$.
In order to show the learning ability of the D-type ILC, we consider a simple linear system which is defined by the equation (3.4). The control objective is to track the following reference trajectory over a finite time interval $[0, 10s]$

$$q_d(t) = \sin 2\omega t \tag{3.9}$$

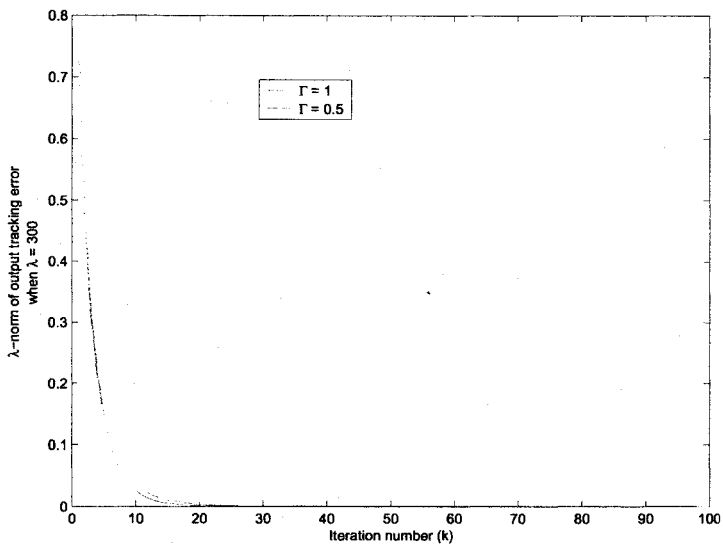

Figure 3.4: λ-norm of the tracking error versus the number of iterations under D-type ILC

with $\omega = 1$. By applying the D-type learning scheme to the system (3.4), the obtained results are shown in figures 3.4, 3.5 and 3.6. Figures 3.4, 3.5 and 3.6 illustrate the evolution of the L_2-norm and sup-norm of the tracking error with respect to the number of iterations. Figure 3.7 and 3.8 shows the time-evolution of the desired trajectory $q_d(t)$ and the actual tracking trajectory $q_k(t)$.

3.2.1 Observation

- The convergence analysis of the D-type iterative process is based upon the use of the exponentially weighted norm (λ-norm). Lee and Bien (1997) proved that the λ-norm of the tracking error can decay monotonically from operation to operation but the sup-norm or ∞-norm of the tracking error may increase to a large value before it converges to the desired level. This arguments can be clearly shown from the simulation results of the ∞-norm and the L_2-norm of the output tracking error in figures 3.5 and 3.6. The transient behavior is a serious problem in the practical application of ILC schemes.

- The derivative signals of dynamical systems are required in order to design and implement the D-type ILC scheme. In fact, most systems are only equipped

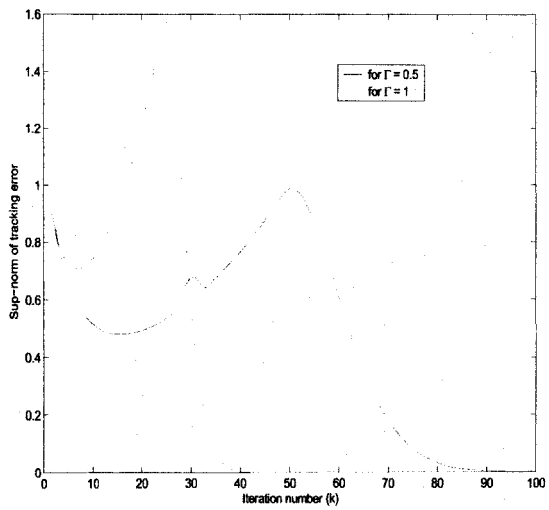

Figure 3.5: Sup-norm of the tracking error versus the number of iterations under D-type ILC

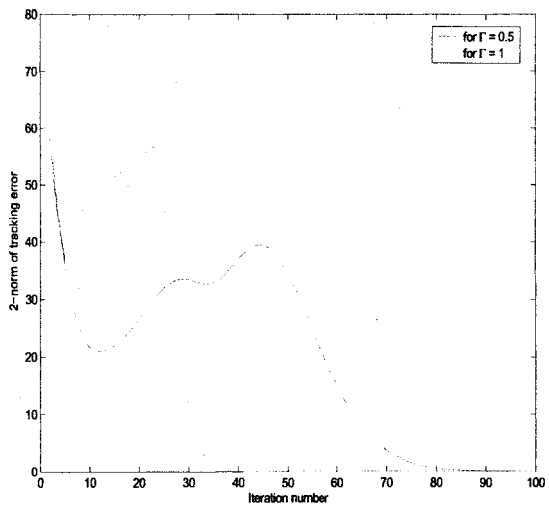

Figure 3.6: L_2-norm of the tracking error versus the number of iterations under D-type ILC

with high-precision sensors for position measurements, and velocity sensors are frequently omitted in order to make considerable savings in cost, volume and weight. Furthermore, measurements from the velocity sensors are often contaminated with a considerable amount of noise. In practice, the velocity signals are estimated from the joint positions, using a filtered derivative, which is often contaminated by severe noises. Therefore, the noise level reduces the effectiveness and performance of the learning process when the number of iteration increases.

3.3 PID-type learning scheme

The most general linear ILC algorithm presented in Arimoto (1985) is given as follows:

$$u_{k+1}(t) = u_k(t) + \Gamma \dot{e}_k(t) + \gamma e_k(t) + \phi \int_0^t e_k(\tau) d\tau \tag{3.10}$$

Where Γ, γ and ϕ are constant gain matrices. This algorithm forms a PID-like system for the error that are measured from the previous operation. It is already shown that when ILC algorithm (3.10) is applied to the system (3.7), the output

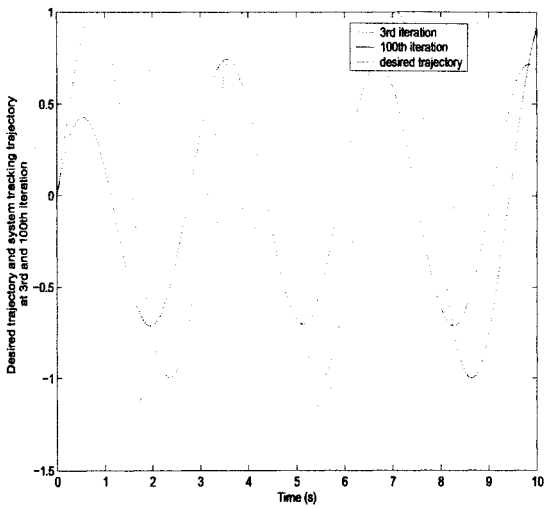

Figure 3.7: Desired trajectory $y_d(t)$ and the output tracking trajectory, $y_k(t)$, at the 3rd and 100th iteration with $\Gamma = 0.5$ under D-type ILC scheme.

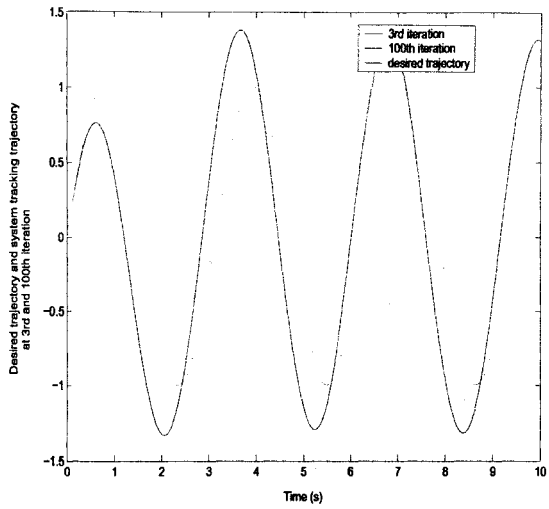

Figure 3.8: Desired trajectory $y_d(t)$ and the system output response, $y_k(t)$, at the 3rd and 100th iteration with $\Gamma = 1$ under D-type ILC scheme.

trajectory converges to the desired trajectory. However, this learning scheme is not robust against initial state error. Park and Bien (1999) generalized this algorithm and showed that the tracking performance can be improved by using the following updating rule against initial state error.

$$u_{k+1}(t) = u_k(t) + \Gamma \left(\dot{e}_k(t) + Q_0 e_k(t) + Q_1 \int_0^t e_k(\tau) d\tau \right) \tag{3.11}$$

It can be shown that when this updating formula is applied to the system (3.7), the output trajectory converges to the following form:

$$\lim_{k \to \infty} y_k(t) = y_d(t) + C_R e^{A_R t} \xi_0 \tag{3.12}$$

with

$$\begin{aligned} A_R &= \begin{bmatrix} 0 & I \\ -Q_1 & -Q_0 \end{bmatrix} \\ C_R &= \begin{bmatrix} I & 0 \end{bmatrix} \\ \xi_0 &= \begin{bmatrix} I \\ -Q_0 \end{bmatrix} C[y_k(0) - y_d(0)] \end{aligned}$$

The equation (3.12) implies that the output trajectory can be controlled in various ways by introducing the integral-term if satisfying the following condition

$$\|I - \Gamma CB\|_\infty \leq \rho < 1 \tag{3.13}$$

The tracking error generated by this algorithm depends on the eigenvalues of the matrix A_R, which can be selected through Q_0 and Q_1. In fact, the tracking error decreases as time increases, and the performance of the learning law (3.11) against initial state error is better than the PID law (3.10). The learning convergence of these iterative processes are based upon the use of the exponentially weighted norm. Therefore, the λ-norm of the tracking error decreases monotonically from iteration to iteration with the large value of λ but the sup-norm or ∞-norm of the tracking error may increase its value before it converges to the desired level.

3.3.1 Numerical Example

Applying PID schemes (3.10) and (3.11) to the system (3.4), the obtained results are shown in figures 3.9 to 3.12 for the following desired trajectory over the finite time interval $[0, 10s]$

$$q_d(t) = \sin 2\omega t \tag{3.14}$$

with $\omega = 1$. The convergence of the λ-norm of the output tracking error is investigated in the presence of initial state errors. The desired trajectory and output tracking trajectory are shown in figures 3.10 and 3.12 without the resetting condition.

3.3.2 Observation

- The main drawback of the PID learning scheme is that the convergence proof of the iterative process is based upon the use of the λ-norm. However, the ∞-norm of the tracking error may increase to a large value before it converges to zero. This transient behavior may cause serious problems in the real world operation because the learning algorithm usually depends on the previous error which may cause hardware failure.

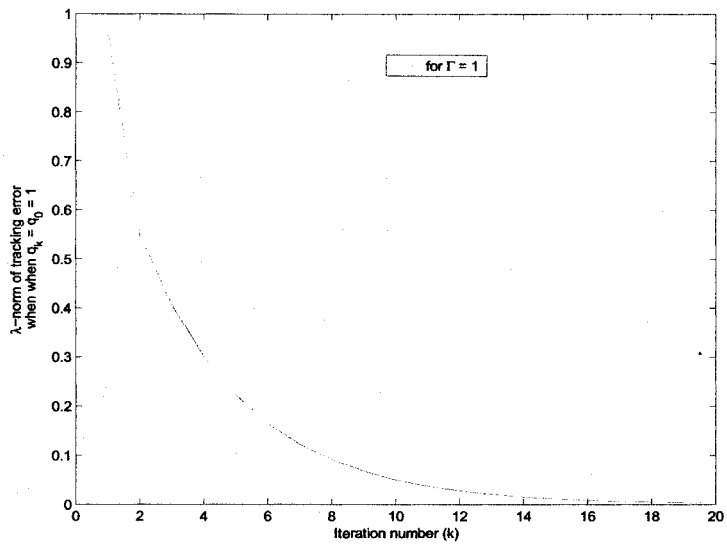

Figure 3.9: λ-norm of the output tracking error versus the number of iterations under the PID-type ILC law (3.10) without resetting condition

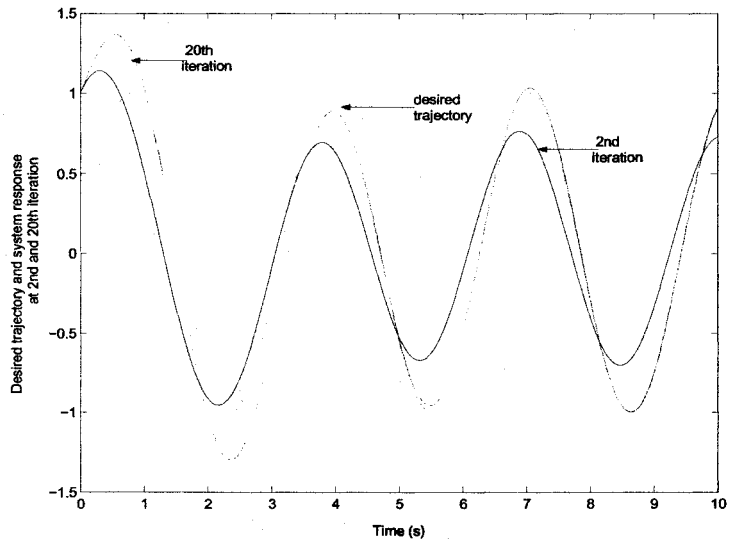

Figure 3.10: Desired trajectory and output tracking trajectory under the PID-type ILC law (3.10) without resetting condition with $\Gamma = 1$

3.4 Concluding remarks on Linear ILC methods

Based on the simulation results of classical ILC schemes, the following conclusion can be drawn:

- In the ILC method, it is assumed that the initial state of the process is equal to that of the desired trajectory for perfect tracking. However, it is practically impossible to set such initial resetting conditions in various plants perfectly. Therefore, the robustness of existing linear ILC schemes against random initial state error, noise and state disturbances is still an open problem.

- The learning stability and convergence analysis of these iterative processes are generally based upon the use of the λ-norm, which is a serious concern in practical applications. In practice, the sup-norm of the tracking error is more appropriate than any other measurements of error performance. However, the convergence analysis of ILC techniques is still an open field in ILC research, for further investigation in achieving monotonic convergence in a suitable norm-topology other than the exponentially-weighted norm.

- The most significant drawback of existing classical linear ILC schemes is that they are applied as a feedforward action directly to the open-loop system. Hence, these control schemes may generate harmful effects if the open-loop system is unstable or inappropriate control law is designed at the first operation.

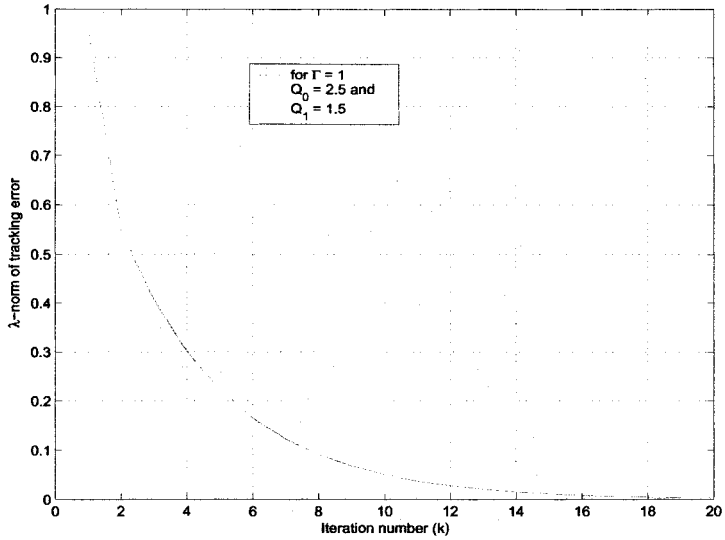

Figure 3.11: λ-norm of the output tracking error versus the number of iterations without initial resetting condition under control law (3.11)

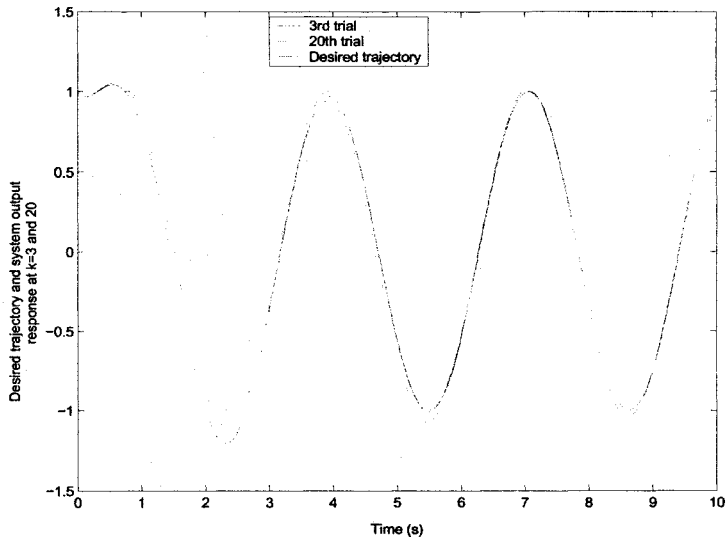

Figure 3.12: Reference trajectory and system output trajectory under PID control law (3.11) without initial resetting condition

Chapter 4

ILC FOR ROBOT MANIPULATORS

ILC research is motivated by robots doing repetitive tasks. Hence, there are several ILC applications to robotic manipulators in the literature. The existing results of these ILC algorithms are mainly characterized by the use of acceleration and velocity measurements, contraction mapping theorem, high feedback and learning gains, linearization of robot dynamics closer to the desired trajectory and *a priori* knowledge of the robot dynamics. In this section, some classical ILC algorithms for highly nonlinear and coupled dynamics of robotic manipulators are analyzed .

D-type ILC

A D-type ILC algorithm is given by

$$u_{k+1}(t) = u_k(t) - \Gamma \dot{e}_k(t) \tag{4.1}$$

where $e_k(t) = y_k(t) - y_d(t)$ and Γ is a constant matrix. A D-type ILC scheme is a simple learning law that generates the control action $u_{k+1}(t)$ at the $(k+1)$-th trial on the basis of the previous control input and the derivative of the error signals. The differentiation of the previous error plays a significant role in the robot dynamics for the learning of desired control tasks. The constant matrix Γ can be chosen arbitrarily to some extent and fixed constant without the description of system dynamics. It was shown in Arimoto *et.al* (1984) that the updating control input $u_{k+1}(t)$ can improve the system response in comparison with the previous control input $u_k(t)$. Arimoto

et.al (1984) have shown that the error at the $(k+1)$-th iteration, namely $e_{k+1}(t)$, is less than the tracking error at the k-th iteration, $e_k(t)$, in the sense of the λ-norm, and the system response $y_k(t)$ approaches to the desired trajectory, $y_d(t)$, as $k \to \infty$. The convergence analysis of this iterative process is based upon the use of the contraction mapping approach.

4.1 D-type ILC for robot manipulators

Using the Lagrangian equation, the dynamical equations of motion of a rigid robot manipulator can be expressed as

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\theta}}) - \frac{\partial L}{\partial \theta} = \tau \tag{4.2}$$

where L denotes the Lagrangian defined as $L = K - V$, the difference between the kinetic energy K and potential energy P of the robot manipulator, $\theta(t) \in \mathbb{R}^n$ is a vetor containing the joint angles and $\tau(t) \in \mathbb{R}^n$ is a vetor of the control input torques to be applied at each joint of robot manipulator. The kinetic energy K can be expressed as

$$K = \frac{1}{2}\dot{\theta}^T M(\theta)\dot{\theta} \tag{4.3}$$

with $M(\theta)$ is real symmetric, bounded and positive definite inertia matrix. It is also pointed out that the potential energy P is independent of $\dot{\theta}(t)$ and is a simple trigonometric function of $\theta(t)$. Therefore, the robot dynamics can be expressed as follows:

$$\begin{aligned} \dot{\theta} &= M^{-1}(\theta)q \\ \dot{q} &= -\frac{\partial K_1}{\partial \theta} - \frac{\partial P}{\partial \theta} + \tau \end{aligned} \tag{4.4}$$

where $K_1 = \frac{1}{2}q^T M^{-1}(\theta)q$. Consider, a desired motion trajectory, $\theta_d(t)$, as given in terms of the joint angle coordinates as $\theta(t) = \theta_d(t)$ for a fixed time interval $[0, T]$. Assuming that the derivative of the desired trajectory is twice continuously differentiable, $\theta_d(t) \in C^2[0, T]$.

Consider the following control law

$$\tau = \frac{\partial P}{\partial \theta} + K_p(\theta_d - \theta) + K_d(\dot{\theta}_d - \dot{\theta}). \tag{4.5}$$

Applying (4.5) to (4.4) leads to

$$\begin{aligned} \dot{\theta} &= M^{-1}(\theta)q \\ \dot{q} &= -\frac{\partial K_1}{\partial \theta} + K_p(\theta_d - \theta) + K_d(\dot{\theta}_d - \dot{\theta}) \end{aligned} \tag{4.6}$$

where K_p and K_d are symmetric positive definite constant matrices. The first term $\frac{\partial P}{\partial \theta}$ in the right hand side of the equation (4.5) is in charge of compensating the torque due to the gravity effect and the second and third terms represent the local position and angular velocity feedback, respectively. The effectiveness of this classical control structure has already been shown in Takegai and Arimoto (1981).

In general, it is very difficult to achieve a perfect tracking of the desired trajectory by using this control law. As a result, there will be always a tracking error with every trial. In order to reduce this error, the aim is to design an additional feed-forward learning control algorithm with this classical control structure (4.5) that can improve the tracking performance with every new operation and to achieve perfect output tracking when iteration goes to infinity. To meet this control objective, it is assumed that the joint velocities are available for feedback. Then, the overall control scheme for robot manipulators can be expressed as

$$\tau_k(t) = \tau_{fb}(t) + \tau_{kff}(t) \tag{4.7}$$

where $\tau_{fb}(t)$ is the feedback and $\tau_{kff}(t)$ is feed-forward learning control input. The fixed feedback control input is given as

$$\tau_{fb}(t) = \frac{\partial p}{\partial \theta} - K_P(\theta - \theta_d) - K_D(\dot{\theta} - \dot{\theta}_d) \tag{4.8}$$

This control input is used to stabilize the closed-loop system and keep its error bounded. The learning input $\tau_{kff}(t)$ is mainly used to identify and compensate uncertain nonlinear dynamics and disturbances when the robot system repeats its operations to follow the desired control trajectory so that the system can improve the tracking performance from operation to operation. This learning input at the k-th trial is modified on the basis of the previous control input signals $\tau_{kff}(t)$ and the time-derivative of the output tracking error $e_k(t)$ as

$$\tau_{(k+1)ff}(t) = \tau_{kff}(t) + \Gamma\frac{d}{dt}\dot{e}_k(t) \tag{4.9}$$

where Γ is an appropriate constant matrix and $e_k(t) = \theta_d(t) - \theta_k(t)$. In order to show the convergence of the D-type learning technique, the system has to satisfy the reseting condition, i.e. $\theta_d(0) - \theta_k(0) = \dot{\theta}_d(0) - \dot{\theta}_k(0) = 0$ for all $k \in \mathbb{N}$. If the learning gain Γ is such that the following inequality is satisfied

$$\|I_n - \Gamma M^{-1}(\theta)\|_\infty < 1 \tag{4.10}$$

for any θ, then $\lim_{k\to\infty} \|e_k(t)\|_\lambda = 0$ and $\lim_{k\to\infty} \|\theta_d(t) - \theta_k(t)\|_\lambda = 0$. The convergence proof of this learning algorithm can be found in Arimoto *et al.* (1985). The presented ILC scheme is tested experimentally on a 5-DOF CRS255 robot manipulator. The experimental results are presented in chapter 7.

4.2 PD-type ILC scheme for robot manipulators

The control objective is to find a control input $\tau_k(t)$ such that the system output $\theta_k(t)$ follows $\theta_d(t)$ for all $t \in [0, T]$ as closely as possible. In order to meet the control objective, PD-type ILC law is used for learning which ensures the convergence of $\theta_k(t)$ to the desired control trajectory $\theta_d(t)$ for all $t \in [0, T]$ when k tends to infinity. The overall control scheme comprises a classical PD structure plus an additional iteratively updated term designed as

$$\tau_k(t) = \tau_{fb}(t) + \tau_{kff}(t) \tag{4.11}$$

where $\tau_{fb}(t)$ is a classical PD feedback, which is used to stabilize the closed-loop system and keep the tracking error bounded. The learning term $\tau_{kff}(t)$ is updated iteratively as follows:

$$\tau_{(k+1)ff}(t) = \tau_{kff}(t) + (\Gamma + \phi\frac{d}{dt})e_k(t) \tag{4.12}$$

where Γ and ϕ are constant learning gain matrices. The iteratively updated control input is designed to learn and compensate the unstructured uncertainties and then improve the tracking performance from iteration to iteration. The convergence analysis of this learning scheme can be found in Arimoto (1985). This PD-type ILC algorithm has been tested on a 5-DOF robot manipulator CRS255. The experimental results are presented in chapter 7.

4.3 Concluding remarks on classical ILC for robot manipulators

From a theoretical point of view, it can be shown that the highly coupled nonlinear uncertain robot manipulators are able to learn a desired motion perfectly through repeated trials. However, there are many important problems arising during the practical implementation of the presented classical ILC mechanism which need to be explored further:

- The resetting condition of the robot manipulator is the first concern that has to be satisfied at the beginning of each operation in order to show the learnability of the presented learning schemes. Fortunately, the initial condition $\theta_k(0) - \theta_d(0) = 0$ is usually realized by the industrial robots with high precision. So, $\dot{\theta}_d(t) = \dot{\theta}_k(0) = 0$ for all $k \in \{1, 2, 3.....\}$ is also satisfied automatically.

- It should also be pointed out that the convergence condition (4.10) is not so restrictive because it is possible to a choose suitable Γ as $M(\theta)$ is symmetric, bounded and positive definite.

- The main problem is the presence of unknown disturbances in the robot dynamics, mainly due to the frictional torques that are exhibited during actual operation. However, if there are unknown disturbances $f(\theta, q)$ in the dynamic equation that satisfy the Lipschitz condition such that

$$\|f(\theta, q) - f(\theta', q')\|_\infty \leq \alpha_1 \|\theta - \theta'\|_\infty + \alpha_2 \|q - q'\|_\infty \tag{4.13}$$

  with constant α_1 and α_2 for any pair (θ, q) and (θ', q'), then the convergence of the learning process is also assured in a certain domain.

4.4 Passivity and dissipativity based ILC approach

In recent years the theory of dissipativity has become an essential part in the control theory in designing more elegant, robust and adaptive control algorithms for nonlinear systems. In contrast, with the Lyapunov theorem, where state variables are considered, the passivity theory is based on the input-output properties of a system.

It plays a crucial role in building a bridge between the energy conservation law in physics and the input-output characterization in the control system theory. Physically, a system is called dissipative if it does not produce energy. The storage function of a dissipative system, which represents the amount of stored energy, often plays an important role of a Lyapunov function of the system. This storage function provides a connection between the theory of dissipative systems and a variety of nonlinear control problems. The learnability of robot dynamics in terms of passivity and dissipativity concepts are discussed in this part of thesis based upon the work of Arimoto *et al.* (1996, 2000), Kawamura *et al.* (1984) and Naniwa (1996).

Learning ability of robot manipulators through passivity approach

The motion equation of rigid robot manipulators can be expressed as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau(t) \tag{4.14}$$

where $q(t)$ denotes the joint angle vector, M(q) is the inertia matrix, $C(q,\dot{q})$ is the matrix of Coriolis and centrifugal forces, G(q) is the vector of gravity forces and $\tau(t)$ is the vector composed of joint torques. The pair of input $\tau(t)$ and output $\dot{q}(t)$ of robot dynamics (4.14) satisfies the passivity condition, i.e.,

$$\int_0^t \dot{q}^T(\delta)\tau(\delta)d\delta \geq -E(0) \geq -\beta \tag{4.15}$$

with a positive constant $\beta \geq 0$. Where E represents the total energy of the robotic arm which can be expressed as

$$E(q,\dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + U(q) \tag{4.16}$$

This relationship plays a fundamental role in the convergence analysis of the learning process.

Algorithm 1

Consider the problem of tracking a reference trajectory $q_d(t)$ over a finite time interval $[0,T]$. The control objective is to find a learning control law such that the output

tracking error $\|e_k\| \rightarrow 0$ in the sense of the L_2-norm as $k \rightarrow \infty$. In order to do that, one can use P-type ILC algorithm (Arimoto *et al.* 1996, Kawamura *et al.* 1984 and Naniwa 1996) for refining the current control input recursively as

$$\tau_{k+1}(t) = \tau_k(t) - \Gamma e_k(t), \tag{4.17}$$

where $e_k(t) = \triangle y_k = q_k(t) - q_d(t)$. Subtracting the desired control input $\tau_d(t)$ on both sides of the above equation, equation (4.17) becomes

$$\triangle\tau_{k+1}(t) = \triangle\tau_k(t) - \Gamma e_k(t), \tag{4.18}$$

with $\triangle\tau_k(t) = \tau_k(t) - \tau_d(t)$ and

$$M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + G(q_d) = \tau_d(t). \tag{4.19}$$

The function $\tau_d(t)$ does not need to be computed but it is sufficient to know that this function does exist. Using equation (4.18), one can write

$$\Gamma^{-1}\|\triangle\tau_{k+1}\|^2 = \Gamma^{-1}\|\triangle\tau_k\|^2 + \Gamma\|e_k\|^2 - 2\int_0^t e_k^T(\delta)\triangle\tau_k(\delta)d\delta \tag{4.20}$$

where $\|e_k\|$ denotes the norm of $e_k(t)$ in $L^2[0,T]$. The input and output pair $(\triangle\tau_k, \triangle y_k)$ satisfies the passivity condition. Based on this phenomenon, one has the following inequality

$$\int_0^t e_k^T(\delta)\triangle\tau_k(\delta)d\delta \geq \frac{1}{2}E_{k+1} - \frac{1}{2}E_k + \beta\int_0^t \|e_k\|^2 d\delta \tag{4.21}$$

with a positive definite storage function E_k and a positive constant β. From (4.21), one has

$$\Gamma^{-1}\|\triangle\tau_{k+1}\|^2 + E_{k+1} \leq \Gamma^{-1}\|\triangle\tau_k\|^2 + E_k - 2\beta\|e_k\|^2 + \Gamma\|e_k\|^2 \tag{4.22}$$

This inequality implies that the sequence $[\Gamma^{-1}\|\triangle\tau_k\|^2 + E_k]$ is monotonically decreasing with increasing k if Γ is chosen as $0 < \Gamma < 2\beta$. As $[\Gamma^{-1}\|\triangle\tau_k\|^2 + E_k]$ is bounded from below and monotonically decreasing, then one can conclude that $\|e_k\| \rightarrow 0$, i.e., the output tracking trajectory $q_k(t)$ converge to desired one $q_d(t)$ when k tends to infinity.

The convergence proof of the above learning process can be found in Kawamura *et al.* (1984) and Naniwa (1996). The presented algorithm 1 has been experimentally tested on the 5-DOF CRS255 robot manipulator for tracking a repetitive circular trajectory over a finite time interval $[0, 63]$. The implementation results are presented in chapter 7.

Algorithm 2

In this section, we present a second learning scheme for robot manipulators on the basis of the passivity approach, which was developed by Arimoto *et al.* (2000). A linear combination of velocity and saturated position errors signals are used in the learning control input as follows

$$\tau_{k+1}(t) = \tau_k(t) - \Gamma \triangle y_k(t), \tag{4.23}$$

with

$$\triangle y_k(t) = (\dot{q}_k - \dot{q}_d) + \alpha sat(q_k - q_d), \tag{4.24}$$

where α is a positive constant. It is assumed that an ideal input $\tau_d(t)$ realizes the desired output $q_d(t)$. Therefore, subtracting $\tau_d(t)$ from both sides of (4.24), one achieves

$$\triangle \tau_{k+1}(t) = \triangle \tau_k(t) - \Gamma \triangle y_k(t), \tag{4.25}$$

with $\triangle \tau_k(t) = (\tau_k(t) - \tau_d(t))$ and

$$M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + G(q_d) = \tau_d(t) \tag{4.26}$$

The pair $(\triangle \tau_k, \triangle y_k)$ satisfies the passivity condition. Based on this condition, one can show that there exists a positive definite storage function $V_k(e_k(t), \dot{e}_k(t))$ in $e_k(t) = q_k(t) - q_d(t)$ and $\dot{e}_k(t) = \dot{q}_k(t) - \dot{q}_d(t)$ and a scalar-valued dissipation function κ with $\kappa > 0$ such that

$$\int_0^t \triangle y^T(\delta) \triangle \tau(\delta) d\delta \geq \frac{1}{2} V_{k+1}(e_{k+1}(t), \dot{e}_{k+1}(t)) - \frac{1}{2} V_k(e_k(t), \dot{e}_k(t)) + \int_0^t \kappa \| \triangle y_k \| d\delta. \tag{4.27}$$

Then, from (4.26), one gets

$$\Gamma^{-1} \| \triangle \tau_{k+1} \|^2 + E_{k+1} \leq \Gamma^{-1} \| \triangle \tau_k \|^2 + E_k + \| e_k \|^2 (\Gamma - 2\kappa) \| e_k \|^2 \tag{4.28}$$

which implies that the output tracking error $\triangle y_k \rightarrow 0$ in the sense of L^2-norm, as $k \rightarrow \infty$ if Γ is chosen as $0 < \Gamma < 2\kappa I$. Finally, one can conclude that $e_k(t) \rightarrow 0$ for all $t \in [0, T]$ when $k \rightarrow \infty$ as $\triangle y_k \rightarrow 0$ when $k \rightarrow \infty$.

The convergence proof of this result can be found in Arimoto *et al.* (2000). The above presented algorithm 2 has been implemented and evaluated on a 5-DOF CRS255 robot manipulator. The experimental results are shown in chapter 7.

Chapter 5

ADAPTIVE ILC

Most existing ILC schemes in the literature are based upon the contraction mapping approach. Recently, researchers have been working on the design of adaptive ILC schemes based on the use of Lyapunov theory. This new approach allows to avoid some of the technical difficulties attached to the contraction mapping based ILC such that the use of the output time derivative for systems with high relative degree, the use of the λ-norm and the requirement of the lipschitz condition. In this chapter, we will present some adaptive ILC schemes for uncertain nonlinear systems proposed in Xu (2002) and Tayebi (2004).

5.1 Adaptive ILC for uncertain nonlinear system

It is well known that traditional adaptive control techniques cannot deal with time varying parameters. Under repeatable control environments, if the time-varying parameters are assumed invariant in the iteration domain, then one can use a simple integrator along the iteration axis in order to update the unknown system parameters. The parametric updating law can be designed by choosing an appropriate Lyapunov-like composite energy function (CEF), which is a combination of a standard Lyapunov function and a L^2-norm of the parametric errors. To motivate this approach, we will present a simple adaptive ILC for uncertain systems (Xu, 2002). Consider the following nonlinear system at the k-th iteration

$$\dot{q}_k(t) = \phi(t)q_k^2(t) + \tau_k(t) \tag{5.1}$$

where $\phi(t) \in C^1[0,T]$ and k denotes the iteration or operation number. The control objective is to find a sequence of control input $\tau_k(t)$ for system (5.1) such that the system output trajectory $q_k(t)$ converges to the desired trajectory $q_d(t)$. It is assumed that the desired trajectory $q_d(t)$ and it's first derivative $\dot{q}_d(t)$ are bounded.

Adaptive ILC design

The error dynamics of this nonlinear system can be expressed as

$$\dot{e}_k(t) = -\phi(t)q_k^2(t) - \tau_k(t) + \dot{q}_d(t) \tag{5.2}$$

The control law is

$$\tau_k(t) = Ke_k(t) + \dot{q}_d(t) - \hat{\phi}_k(t)q_k^2(t) \tag{5.3}$$

with the parameter adaptation rule

$$\hat{\phi}_k(t) = \hat{\phi}_{k-1}(t) - q_k^2(t)e_k(t) \tag{5.4}$$

where $\hat{\phi}_{-1}(t) = 0$.

Learning convergence with composite energy function (CEF)

The proof of convergence of the above learning process has two parts. Part A finds an appropriate energy function, $V_k(t)$, and then proves that this function, $V_k(t)$, is a non-increasing sequence and bounded if $V_0(t)$ is bounded. In part B, the uniform convergence of the output tracking error to zero is proven as iteration goes to infinity.

Part A

In order to prove the convergence of the above ILC, one needs to find a suitable energy function which plays a similar role as Lyapunov function in adaptive control. Consider the following CEF

$$V_k(t) = \frac{1}{2}e_k^2(t) + \frac{1}{2}\int_0^t \tilde{\phi}_k^2(\sigma)d\sigma \tag{5.5}$$

with $\tilde{\phi}_k(t) = \phi(t) - \hat{\phi}_k(t)$, where $\hat{\phi}_k(t)$ is the estimated value of $\phi(t)$ and $\tilde{\phi}_k(t)$ is the estimation error. The difference of the energy function, $\triangle V_k(t)$, at the k-th operation

is given by

$$\triangle V_k = V_k - V_{k-1}, \tag{5.6}$$

$$\triangle V_k(t) = \frac{1}{2}e_k^2(t) + \frac{1}{2}\int_0^t (\tilde{\phi}_k^2 - \tilde{\phi}_{k-1}^2)d\sigma - \frac{1}{2}e_{k-1}^2(t). \tag{5.7}$$

One can write the first term of equation (5.7) as

$$\frac{1}{2}e_k^2 = \frac{1}{2}e_k^2(0) + \int_0^t e_k\dot{e}_k d\sigma. \tag{5.8}$$

Now, using the control law (5.3) and the error dynamics (5.2) in (5.8), one has

$$\frac{1}{2}e_k^2 = \frac{1}{2}e_k^2(0) + \int_0^t (-Ke_k^2 - \tilde{\phi}_k q_k^2 e_k)d\sigma. \tag{5.9}$$

Using the initial resetting condition, i.e., $e_k(0) = 0$, the equation leads to

$$\frac{1}{2}e_k^2 = \int_0^t (-Ke_k^2 - \tilde{\phi}_k q_k^2 e_k)d\sigma \tag{5.10}$$

Using the parametric updating law (5.4), the second term of the equation (5.7) becomes

$$\frac{1}{2}\int_0^t (\tilde{\phi}_k^2 - \tilde{\phi}_{k-1}^2)d\sigma = \int_0^t (\tilde{\phi} q_k^2 e_k - \frac{1}{2}q_k^4 e_k^2)d\sigma \tag{5.11}$$

So, the difference of the composite energy function becomes

$$\triangle V_k = -\int_0^t Ke_k^2 d\sigma - \frac{1}{2}e_{k-1}^2 - \frac{1}{2}\int_0^t q_k^4 e_k^2 d\sigma \leq 0 \tag{5.12}$$

Here, V_k is a non-increasing sequence. Therefore, V_k is bounded if V_0 is bounded. This implies that $e_k(t)$ and $\frac{1}{2}\int_0^t \tilde{\phi}_k^2(\sigma)d\sigma$ are bounded for all $k \in N$ as the iteration goes to infinity. In order to show the boundedness of V_0, one can take the derivative of V_0, i.e.,

$$\dot{V}_0 = \dot{e}_0 e_0 + \frac{\tilde{\phi}_0^2}{2} = -Ke_0^2 + \frac{\tilde{\phi}_0^2}{2} - \tilde{\phi}_0 q_0^2 e_0 \tag{5.13}$$

At $k = 0$, $\hat{\phi}_{-1}(t) = 0\ \forall t \in [0, T]$, then $\dot{V}_0$ becomes

$$\dot{V}_0 = -Ke_0^2 + \frac{\tilde{\phi}_0^2}{2} + \tilde{\phi}_0\hat{\phi}_0 = -Ke_0^2 - \frac{\tilde{\phi}_0^2}{2} + \tilde{\phi}_0\phi \tag{5.14}$$

Using Young's inequality, one can write

$$\tilde{\phi}_0\phi \leq \kappa\tilde{\phi}_0^2 + \frac{1}{4\kappa}\phi^2 \tag{5.15}$$

with $\kappa > 0$. Hence,

$$\dot{V}_0 \leq -Ke_0^2 - \tilde{\phi}_0^2(\frac{1}{2} - \kappa) + \frac{1}{4\kappa}\phi^2 \tag{5.16}$$

with $0 < \kappa < \frac{1}{2}$.
Since, $\phi(t) \in C^1[0,T]$ i.e., there exists a finite bound $\phi_m \geq \phi(t)$ for all $t \in [0,T]$. Hence, one can conclude that V_0 is negative semi-definite outside the region

$$\frac{1}{4\kappa}\phi_m^2 \geq Ke_0^2 + \tilde{\phi}_0^2(\frac{1}{2} - \kappa) \tag{5.17}$$

which also implies the boundedness of V_0 for all $t \in [0,T]$.

PART B

Uniform convergence of the tracking error:
Using the equation (5.7) repeatedly, one achieves

$$V_k(t) = V_0(t) + \sum_{m=1}^{k} \Delta V_m \tag{5.18}$$

$$\lim_{k\to\infty} V_k(t) \leq V_0(t) - \lim_{k\to\infty} \sum_{m=1}^{k} \int_0^t Ke_m^2 d\sigma - \lim_{k\to\infty} \sum_{m=1}^{k-1} e_m^2(t) \tag{5.19}$$

So, one can write

$$V_0(t) - \lim_{k\to\infty} V_k(t) \geq \lim_{k\to\infty} \sum_{m=1}^{k} \int_0^t Ke_m^2 d\sigma + \lim_{k\to\infty} \sum_{m=1}^{k-1} e_m^2(t) \tag{5.20}$$

which implies that $\|e_k(t)\| \to 0$ for all $t \in [0,T]$ as $k \to \infty$, since $V_k(t)$ is bounded $\forall t \in [0,T]$ and $\forall k \in [0,T]$ when k goes to infinity.

5.1.1 Numerical example

Consider the nonlinear dynamic system

$$\dot{q}_k(t) = \phi(t)q_k^2(t) + \tau_k(t) \tag{5.21}$$

With the unknown time varying parameter

$$\phi(t) = cos\pi t + 3$$

where $t \in [0, 20]$. Consider, the desired trajectory $q_d(t) = sin2\pi t + 0.5$ over $t \in [0, 20s]$ with $q_k(0) = 0.5$. In this case, the objective is to design a bounded control law $\tau_k(t)$ for the system (5.21) that guarantees the boundedness of $q_k(t)$ over the finite time interval $[0, 20s]$ and the convergence of output trajectory, $q_k(t)$, to the desired one, $q_d(t)$. Using the control algorithm (5.3) and the parameter updating law (5.4), the results are shown in figure 5.1 and 5.2 for different values of K.

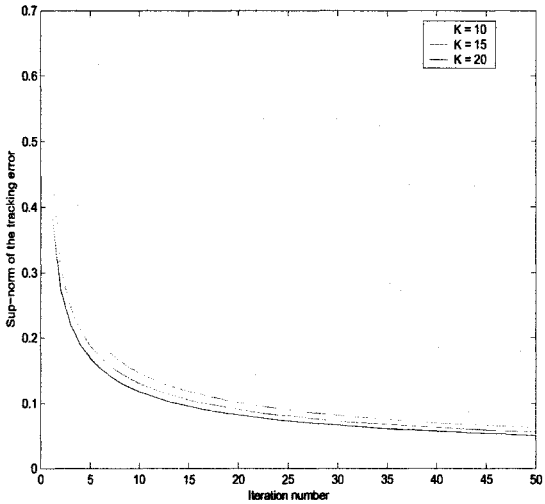

Figure 5.1: Sup-norm of the output tracking error versus the iteration number

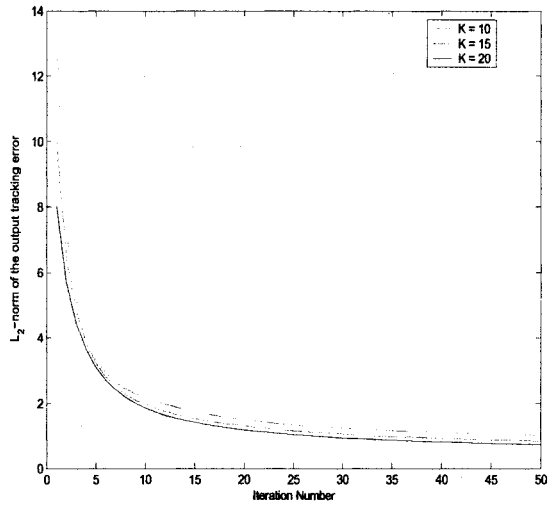

Figure 5.2: L_2-norm of the output tracking error versus the iteration number

5.2 Adaptive ILC for robot manipulators

In this part of the thesis, we present a new generation of adaptive iterative learning control (AILC) techniques proposed in (Tayebi 2004) for the trajectory tracking control problem of rigid robot manipulators with unknown parameters and subject to external disturbances. The learning schemes are designed based upon the use of a composite energy function, which reflects the system energy. The idea of using CEF in the ILC method opens a new avenue for the learning control design and convergence analysis. The control strategy consists of using a classical PD structure plus an additional iteratively updated term designed in order to cope with the unknown parameters and disturbances.

Preliminaries

The equations of motion of a n degrees-of-freedom rigid robot manipulator can be expressed as

$$M(q_k)\ddot{q}_k + C(q_k, \dot{q}_k)\dot{q}_k + G(q_k) = \tau_k(t) + d_k(t) \tag{5.22}$$

where $t \in \mathbb{R}_+$ denotes the time, $k \in \mathbb{N}$ represents the iteration or trial number. The signals $q_k(t) \in \mathbb{R}^n$, $\dot{q}_k(t) \in \mathbb{R}^n$ and $\ddot{q}_k(t) \in \mathbb{R}^n$ are the joint position, velocity and acceleration vectors at the k-th iteration. $M(q_k) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C(q_k, \dot{q}_k)\dot{q}_k \in \mathbb{R}^n$ is the vector of coriolis and centrifugal forces. $G(q_k) \in \mathbb{R}^n$ is the vector of gravitational forces, $\tau_k(t) \in \mathbb{R}^n$ is the control input containing the torques and forces and $d_k(t) \in \mathbb{R}^n$ is the vector containing the unmodeled dynamics and other unknown external disturbances.
In order to design an adaptive ILC scheme, the following assumptions need to be imposed

A. The desired control trajectory $q_d(t)$, its first derivative $\dot{q}_d(t)$, second derivative $\ddot{q}_d(t)$ and disturbance $d_k(t)$ are bounded $\forall t \in [0, T]$ and $\forall k \in \mathbb{N}$.

B. The resetting condition is satisfied, *i.e.*, $q_d(0) - q_k(0) = \dot{q}_d(0) - \dot{q}_k(0) = 0$, $\forall k \in \mathbb{N}$.

C. The joint positions and velocity measurements are available for feedback.

The following structural properties of robot manipulators are well known and are utilized in order to facilitate the adaptive ILC design

L_1) $M(q_k) \in \mathbb{R}^{n\times n}$ is symmetric, bounded, and positive definite.

L_2) The matrix $\dot{M}(q_k) - 2C(q_k, \dot{q}_k)$ is skew symmetric

L_3) $G(q_k) + C(q_k,\dot{q}_k)\dot{q}_d(t) = \Psi(q_k,\dot{q}_k)\xi(t)$, where $\Psi(q_k,\dot{q}_k) \in \mathbb{R}^{n\times(m-1)}$ is a known matrix and $\xi(t) \in \mathbb{R}^{(m-1)}$ is an unknown bounded vector.

L_4) $\|C(q_k,\dot{q}_k)\| \leq k_c\|\dot{q}_k\|$ and $\|G(q_k)\| < k_g$, $\forall t \in [0,T]$ and $\forall k \in \mathbb{N}$, where k_c and k_g are unknown positive parameters.

Adaptive ILC schemes

In this section, three adaptive ILC schemes and their convergence properties are presented (Tayebi 2004). The overall control scheme is shown in figure 5.3.

Scheme 1

Consider the robot system (5.22) with properties L_1, L_2 and L_3 under the following control law

$$\tau_k(t) = K_P\tilde{q}_k(t) + K_D\dot{\tilde{q}}_k(t) + \phi(q_k, \dot{q}_k, \dot{\tilde{q}}_k)\hat{\theta}_k(t), \tag{5.23}$$

with the parametric adaptation law

$$\hat{\theta}_k(t) = \hat{\theta}_{k-1}(t) + \Gamma\phi^T(q_k, \dot{q}_k, \dot{\tilde{q}}_k)\dot{\tilde{q}}_k(t), \tag{5.24}$$

where $\hat{\theta}_{-1}(t) = 0$, $\tilde{q}_k(t) = q_d(t) - q_k(t)$ and $\dot{\tilde{q}}_k(t) = \dot{q}_d(t) - \dot{q}_k(t)$. The matrix $\phi(q_k, \dot{q}_k, \dot{\tilde{q}}_k) \in \mathbb{R}^{n\times m}$ is known and can be expressed as

$$[\Psi(q_k, \dot{q}_k)\ \ sgn(\dot{\tilde{q}}_k)] \tag{5.25}$$

where $\text{sgn}(\dot{\tilde{q}}_k)$ is the vector obtained by applying the signum function to all elements of $\dot{\tilde{q}}_k(t)$. The matrices $K_P \in \mathbb{R}^{n\times n}$, $K_D \in \mathbb{R}^{n\times n}$ and $\Gamma \in \mathbb{R}^{n\times n}$ are symmetric positive definite. If assumptions A and B are satisfied, then $\tilde{q}_k(t)$, $\dot{\tilde{q}}_k(t)$ and $\int_0^t \hat{\theta}_k^T(\tau)\Gamma^{-1}\hat{\theta}_k(\tau)d\tau$ are bounded for all $t \in [0,T]$ and all $k \in \mathbb{N}$ and $\lim_{k\to\infty} \tilde{q}_k(t) = \lim_{k\to\infty} \dot{\tilde{q}}_k(t) = 0$, $\forall t \in [0,T]$

Convergence analysis with composite energy function

The proof of convergence of the above ILC scheme has two parts (Tayebi 2004). Part A finds an appropriate energy function E_k and then proves that this function is a non-increasing sequence and bounded if E_0 is bounded over $t \in [0, T]$. In the second part, the uniform convergence of the output tracking error is proven when iteration goes to infinity.

PART A

Consider the following CEF

$$E_k(\dot{\tilde{q}}_k(t), \tilde{q}_k(t), \tilde{\theta}_k(t)) = V_k(\dot{\tilde{q}}_k(t), \tilde{q}_k(t)) + \frac{1}{2}\int_0^t \tilde{\theta}_k^T(\tau)\Gamma^{-1}\tilde{\theta}_k(\tau)d\tau$$

with $\tilde{\theta}_k(t) = \theta(t) - \hat{\theta}_k(t)$, where $\theta(t) = [\xi^T(t) \;\; \beta]^T \in \mathbb{R}^m$, $\hat{\theta}_k(t) = [\hat{\xi}_k^T(t) \;\; \hat{\beta}_k(t)]^T$ is the estimated value of $\theta(t)$ and $\tilde{\theta}_k(t)$ is the estimation error. The unknown parameter $\xi(t)$ is defined in (L_3) and β is obtained according to (L_1) and (A) such that $\| M(q_k)\ddot{q}_d - d_k \| \leq \beta \; \forall t \in [0, T]$ and $\forall k \in \mathbb{N}$.

The first term $V_k(\dot{\tilde{q}}_k(t), \tilde{q}_k(t))$ is a combination of the kinetic and potential energy of the manipulator,

$$V_k(\dot{\tilde{q}}_k(t), \tilde{q}_k(t)) = \frac{1}{2}\dot{\tilde{q}}_k^T M(q_k)\dot{\tilde{q}}_k + \frac{1}{2}\tilde{q}_k^T K_P \tilde{q}_k \tag{5.26}$$

One can rewrite V_k as follows:

$$V_k(\dot{\tilde{q}}_k(t), \tilde{q}_k(t)) = V_k(\dot{\tilde{q}}_k(0), \tilde{q}_k(0)) + \int_0^t (\dot{\tilde{q}}_k^T M\ddot{\tilde{q}}_k + \frac{1}{2}\dot{\tilde{q}}_k^T \dot{M}\dot{\tilde{q}}_k + \dot{\tilde{q}}_k^T K_P \tilde{q}_k)d\tau \tag{5.27}$$

Using the initial resetting condition (B), equations (5.22) and properties (L_2, L_3), equation (5.27) leads to

$$V_k(\dot{\tilde{q}}_k(t), \tilde{q}_k(t)) \;\; \leq \;\; \int_0^t \dot{\tilde{q}}_k^T(\phi(q_k, \dot{q}_k, \dot{\tilde{q}}_k)\tilde{\theta}_k - K_D\dot{\tilde{q}}_k)d\tau$$

The difference of the energy function E_k at k-th iteration is given by

$$\begin{aligned} \triangle E_k &= E_k - E_{k-1} \\ \triangle E_k &= V_k - V_{k-1} - \frac{1}{2}\int_0^T (\bar{\theta}_k^T\Gamma^{-1}\bar{\theta}_k + 2\bar{\theta}_k^T\Gamma^{-1}\tilde{\theta}_k)d\tau \end{aligned}$$

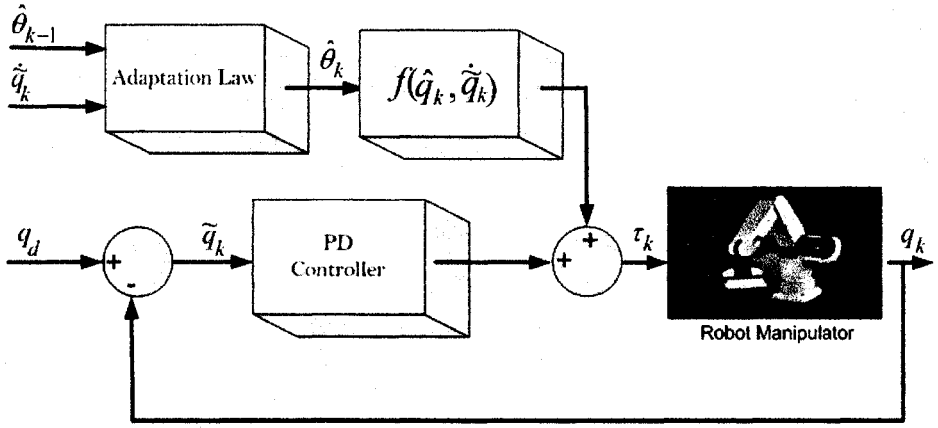

Figure 5.3: Adaptive ILC structure

where $\bar{\theta}_k = \hat{\theta}_k - \hat{\theta}_{k-1}$.
Substituting (5.24), (B) and $V_k(\dot{\tilde{q}}_k(t), \tilde{q}_k(t))$ in $\triangle E_k$, one has

$$\triangle E_k \leq -V_{k-1} - \frac{1}{2}\int_0^t \dot{\tilde{q}}_k^T(\phi(q_k, \dot{q}_k, \dot{\tilde{q}}_k)\Gamma\phi^T(q_k, \dot{q}_k, \dot{\tilde{q}}_k) + 2K_D)\dot{\tilde{q}}_k d\tau \leq 0$$

which implies that E_k is a non-increasing sequence. Hence, one can conclude that the energy function E_k is bounded as E_0 is uniformly bounded. This implies that $\tilde{q}_k(t)$, $\dot{\tilde{q}}_k(t)$ and $\int_0^t \tilde{\theta}_k^T(\tau)\Gamma^{-1}\tilde{\theta}_k(\tau)d\tau$ are bounded for all $k \in N$.
To show the boundedness of E_0 for all repeated operations and operation time period, one can take the derivative of E_0, i.e.,

$$\dot{E}_0 \leq \dot{\tilde{q}}_0^T(\phi(q_0, \dot{q}_0, \dot{\tilde{q}}_0)\tilde{\theta}_0 - K_D\dot{\tilde{q}}_0) + \frac{1}{2}\tilde{\theta}_0^T\Gamma^{-1}\tilde{\theta}_0 \tag{5.28}$$

Since $\hat{\theta}_{-1}(t) = 0\ \forall[0, T]$, thus $\hat{\theta}_0(t) = \Gamma\phi^T(q_0, \dot{q}_0, \dot{\tilde{q}}_0)\dot{\tilde{q}}_0(t)$ and $\dot{E}_0$ leads to

$$\dot{E}_0 \leq -\dot{\tilde{q}}_0^T K_D\dot{\tilde{q}}_0 + (\hat{\theta}_0^T + \frac{1}{2}\tilde{\theta}_0^T)\Gamma^{-1}\tilde{\theta}_0 \tag{5.29}$$

$$\leq -\dot{\tilde{q}}_0^T K_D\dot{\tilde{q}}_0 - \frac{1}{2}\tilde{\theta}_0^T\Gamma^{-1}\tilde{\theta}_0 + \theta^T\Gamma^{-1}\tilde{\theta}_0 \tag{5.30}$$

Using Young's inequality, one has for any $\mathcal{K} > 0$

$$\theta^T\Gamma^{-1}\tilde{\theta}_0 \leq \mathcal{K}\|\Gamma^{-1}\tilde{\theta}_0\|^2 + \frac{1}{4\mathcal{K}}\|\theta\|^2,$$

Hence, $\dot{E}_0$ becomes

$$\dot{E}_0 \leq -\rho_1 \|\dot{\tilde{q}}_0\|_2^2 - \rho_2 \|\tilde{\theta}_0\|^2 + \frac{1}{4\mathcal{K}} \|\theta\|^2, \tag{5.31}$$

with $\rho_1 = \lambda_{min}(K_D)$, $\rho_2 = \frac{1}{2}\lambda_{min}(\Gamma^{-1}) - \mathcal{K}\lambda_{max}^2(\Gamma^{-1})$ and $0 < \mathcal{K} \leq \frac{\lambda_{min}(\Gamma^{-1})}{2\lambda_{max}^2(\Gamma^{-1})}$. Since $\xi^T(t)$ is bounded over $t \in [0, T]$, then $\theta(t)$ is bounded for all $t \in [0, T]$, i.e., $\|\theta(t)\|_{\infty e} \leq \theta_{max}$. Therefore, one can write from (5.31) that $\dot{E}_0(t) \leq \frac{1}{4\mathcal{K}} \|\theta\|_{max}^2$, which implies that $\dot{E}_0(t)$ is uniformly continuous and bounded over $[0, T]$. Finally, one can conclude that $E_0(t)$ is bounded over $[0, T]$ as $\dot{E}_0(t)$ is bounded over the finite interval $t \in [0, T]$.

PART B

One can write $\triangle E_k$ repeatedly

$$E_k \leq E_0 - \sum_{j=1}^{k} V_{j-1} \tag{5.32}$$

$$\leq E_0 - \frac{1}{2} \sum_{j=1}^{k} \tilde{q}_{j-1}^T K_P \tilde{q}_{j-1} - \frac{1}{2} \sum_{j-1}^{k} \dot{\tilde{q}}_{j-1}^T M(q_{j-1}) \dot{\tilde{q}}_{j-1} \tag{5.33}$$

which implies that

$$E_0 \geq \frac{1}{2} \sum_{j=1}^{k} \tilde{q}_{j-1}^T K_P \tilde{q}_{j-1} + \frac{1}{2} \sum_{j-1}^{k} \dot{\tilde{q}}_{j-1}^T M(q_{j-1}) \dot{\tilde{q}}_{j-1} \tag{5.34}$$

Hence, the boundedness of $E_0(t)$ implies that $\dot{\tilde{q}}_k(t)$ and $\tilde{q}_k(t)$ converges to zero pointwisely as the iteration goes to infinity.

Observation

The control design requires m iterative parameters, which is generally larger than the number of degrees-of-freedom n. Therefore, large memory components are necessary to implement this scheme in real-time applications. It also requires the knowledge of the matrix $\psi(q_k, \dot{q}_k)$.

Scheme 2

Consider the control law for the dynamical equation of the robot manipulator (5.22) with properties (L_1, L_2, L_4) (Tayebi 2004).

$$\tau_k(t) = K_P\tilde{q}_k(t) + K_D\dot{\tilde{q}}_k(t) + \eta(\dot{\tilde{q}}_k)\hat{\theta}_k(t) \tag{5.35}$$

with the parametric adaptation law

$$\hat{\theta}_k(t) = \hat{\theta}_{k-1}(t) + \Gamma\eta^T(\dot{\tilde{q}}_k)\dot{\tilde{q}}_k(t), \tag{5.36}$$

where $\hat{\theta}_{-1}(t) = 0$. The control gains $K_P \in \mathbb{R}^{n\times n}$ and $K_D \in \mathbb{R}^{n\times n}$ and learning gain $\Gamma \in \mathbb{R}^{2\times 2}$ are symmetric positive definite matrices. The function $\eta(\dot{\tilde{q}}_k)$ can be defined as $\eta(\dot{\tilde{q}}_k) = [\dot{\tilde{q}}_k \; sgn(\dot{\tilde{q}}_k)]$. If the assumptions A and B are satisfied, then $\tilde{q}_k(t)$, $\dot{\tilde{q}}_k(t)$, $\int_0^t \tilde{\theta}_k^T(\tau)\Gamma^{-1}\tilde{\theta}_k(\tau)d\tau$ are bounded for all $t \in [0, T]$ and all $k \in \mathbb{N}$ and $\lim_{k\to\infty} \tilde{q}_k(t) = \lim_{k\to\infty} \dot{\tilde{q}}_k(t) = 0$, $\forall t \in [0, T]$.

Learning convergence

Consider, the Lyapunov-like composite energy functions

$$E_k(\dot{\tilde{q}}_k(t), \tilde{q}_k(t), \tilde{\theta}_k(t)) = V_k(\dot{\tilde{q}}_k(t), \tilde{q}_k(t)) + \frac{1}{2}\int_0^t \tilde{\theta}_k^T(\tau)\Gamma^{-1}\tilde{\theta}_k(\tau)d\tau \tag{5.37}$$

with $\tilde{\theta}_k(t) = \theta(t) - \hat{\theta}_k(t)$. The vector $\theta(t)$ is defined as $\theta = [\alpha \;\; \delta]^T \in \mathbb{R}^2$. The unknown parameters α and δ can be determined as follows:

$$\begin{aligned} \dot{\tilde{q}}_k^T(M(q_k)\ddot{q}_d + C(q_k, \dot{q}_k)\dot{q}_d + G(q_k) - d_k) &\leq \|\dot{\tilde{q}}_k\|(\beta + k_g + k_c\|\dot{q}_d\| \; \|\dot{q}_k\|) \\ &\leq \|\dot{\tilde{q}}_k\|(\beta + k_g + k_c\|\dot{q}_d\|^2 + k_c\|\dot{q}_d\| \; \|\dot{\tilde{q}}_k\|) \end{aligned}$$

where k_c, k_g are unknown positive parameters and β is obtained according to (L_1, A) such that $\| M(q_k)\ddot{q}_d - d_k \| \leq \beta$. As $\dot{q}_d(t)$ is bounded, the unknown parameter α and δ are defined as follows:

$$\dot{\tilde{q}}_k^T(M(q_k)\ddot{q}_d + C(q_k, \dot{q}_k)\dot{q}_d + G(q_k) - d_k) \leq \dot{\tilde{q}}_k^T(\alpha\dot{\tilde{q}}_k + \delta sgn(\dot{\tilde{q}}_k)) \tag{5.38}$$

$$\leq \dot{\tilde{q}}_k^T\eta(\dot{\tilde{q}}_k)\theta \tag{5.39}$$

where $\alpha = k_c Sup_{t\in[0,T]} \|\dot{q}_d\|$ and $\delta = \beta + k_g + k_c Sup_{t\in[0,T]} \|\dot{q}_d(t)\|^2$
The remaining proof of convergence of this adaptive iterative process is similar to the proof of scheme 1.

Observation

- The control design does not require any *a priori* knowledge of the robot dynamics which is an interesting and significant development in the ILC research.
- In contrast with other classical and adaptive ILC schemes, the control scheme is simpler in structure and easier to implement in view of the necessary memory space and computing power for the real-time application.
- In this framework, the acceleration measurements and the bounds of the robot parameters are not needed and the only requirement on the control and learning gains is the positive definiteness condition.

Scheme 3

Consider the robot model (5.22) with properties (L_1, L_2, L_4) under the following control Law

$$\tau_k(t) = K_P\tilde{q}_k(t) + K_D\dot{\tilde{q}}_k(t) + \hat{\delta}_k(t)sgn(\dot{\tilde{q}}_k(t)), \tag{5.40}$$

with the adaptation law

$$\hat{\delta}_k(t) = \hat{\delta}_{k-1}(t) + \gamma\dot{\tilde{q}}_k^T(t)sgn(\dot{\tilde{q}}_k(t)), \tag{5.41}$$

where $\hat{\delta}_{-1}(t) = 0$. The gains $K_P \in \mathbb{R}^{n\times n}$ and $K_D \in \mathbb{R}^{n\times n}$ are symmetric positive definite matrices, and γ is a positive scalar. If $(K_D - \alpha I)$ is positive semi-definite, with $\alpha = k_c sup_{t\in[0,T]}\|\dot{q}_d(t)\|$ and properties (A, B), then $\tilde{q}_k(t)$, $\dot{\tilde{q}}_k(t)$ and $\int_0^t \hat{\delta}_k^2(\tau)d\tau$ are bounded for all $t \in [0, T]$ and $k \in \mathbb{N}$ and $\lim_{k\to\infty} \tilde{q}_k(t) = \lim_{k\to\infty} \dot{\tilde{q}}_k(t) = 0$, $\forall t \in [0, T]$

Learning convergence

In order to show the convergence of the above ILC scheme, the following composite energy function can be used

$$E_k = V_k(\dot{\tilde{q}}_k(t), \tilde{q}_k(t)) + \frac{1}{2}\int_0^t \gamma^{-1}\tilde{\delta}_k^2(\tau)d\tau, \tag{5.42}$$

with $\tilde{\delta}_k(t) = \delta - \hat{\delta}_k(t)$, where δ is the unknown parameter. The unknown parameter δ is obtained as follows:

$$\begin{aligned} \dot{\tilde{q}}_k^T(M(q_k)\ddot{q}_d + C(q_k,\dot{q}_k)\dot{q}_d + G(q_k) - d_k) &\leq \|\dot{\tilde{q}}_k\|(\beta + k_g + k_c\|\dot{q}_d\|\ \|\dot{q}_k\|) \qquad (5.43) \\ &\leq \|\dot{\tilde{q}}_k\|(\beta + k_g + k_c\|\dot{q}_d\|^2 + k_c\|\dot{q}_d\|\ \|\dot{\tilde{q}}_k\|) \end{aligned}$$

where k_c, k_g are unknown parameters and β is obtained according to (L_1, A) such that $\| M(q_k)\ddot{q}_d - d_k \| \leq \beta$. Then the equation (5.43) leads to

$$\begin{aligned} \dot{\tilde{q}}_k^T(M(q_k)\ddot{q}_d + C(q_k,\dot{q}_k)\dot{q}_d + G(q_k) - d_k) &\leq \dot{\tilde{q}}_k^T(\alpha\dot{\tilde{q}}_k + \delta sgn(\dot{\tilde{q}}_k)) \\ &\leq \dot{\tilde{q}}_k^T\eta(\dot{\tilde{q}}_k)\theta \qquad (5.44) \end{aligned}$$

where $\alpha = k_c Sup_{t\in[0,T]}\ \|\dot{q}_d\|$ and $\delta = \beta + k_g + k_c Sup_{t\in[0,T]}\ \|\dot{q}_d(t)\|^2$
The remaining proof of convergence of this adaptive iterative process is similar to the proof of scheme 1.

Observation

- The number of iterative variables used in this learning scheme is just one, which is an interesting fact because it saves memory components and computing power in real-time applications.

5.3 Concluding remarks on adaptive ILC approach

- The presented control strategy can be used in a straightforward manner for industrial robot manipulators that are already working under a PD controller by just adding the iterative term to the control input in order to enhance the tracking performance from operation to operation.

The presented adaptive ILC schemes have been implemented and evaluated on a 5DOF CRS255 industrial robot manipulator. The experimental results will be presented in chapter 7.

5.4 Adaptive ILC without using the joint velocity in the parametric adaptation law

Many industrial robot manipulators today are only equipped with high precision sensors for the position measurements. Velocity sensors are frequently omitted due to saving cost, volume and weight. Therefore, in the presented AILC implementation on the robot manipulators, the velocity signals are not measured but estimated from the joint position signals using a filtered derivative which produces considerable amount of noise. A major drawback of any ILC scheme in practical applications is that it amplifies the noise present in the measurements, which leads to poor learning performance and forces to stop the learning process. In this section, we present a new AILC scheme for rigid robot manipulators that does not use the velocity in the parametric adaptation law. As a result, the learning process of the proposed AILC strategy can be continued until the output trajectory $\theta_k(t)$ converge to the desired control trajectory $\theta_d(t)$ without having any difficulties from the measurement noise. The control scheme consists of a linear parameterized feedback control structure and a feed-forward learning term in order to cope with the unknown parameters and disturbances. It is worth noting that, at this point in time, we don't have any rigorous proof for this control scheme.

Adaptive Scheme 4

The following adaptive ILC control law is proposed for the repetitive robot dynamics (5.22)

$$\tau_k(t) = \nu z_k(t) + \eta(\tilde{q}_k)\hat{\theta}_k(t) \tag{5.45}$$

with the learning parametric adaptation law

$$\hat{\theta}_k(t) = \hat{\theta}_{k-1}(t) + K_l \eta^T(\tilde{q}_k)\tilde{q}_k(t) \tag{5.46}$$

with $\hat{\theta}_{-1}(t) = 0$. Where $\nu \in \mathbb{R}^{n\times n}$ and $K_l \in \mathbb{R}^{2\times 2}$ are symmetric positive definite matrices. The matrix $\eta(\tilde{q}_k)$ is defined as $\eta(\tilde{q}_k) = [\tilde{q}_k \; sign(\tilde{q}_k)]$. The convergence analysis of the above adaptive learning control technique will be part of our future research work. The presented AILC scheme 4 has been tested experimentally on a

5-DOF robot manipulator as shown in figure 6.1. The experimental results will be presented in chapter 7.

Chapter 6

EXPERIMENTAL SETUP

In this chapter, we introduce the experimental platform used for the real-time implementation of the presented ILC algorithms.

6.1 Experimental platform

The robot used in this thesis is the 5-DOF CRS255 (CATALYST5) industrial robot manipulator manufactured by CRS robotics depicted in figure 6.1. The robot arm is constructed of high tensile strength aluminum alloy components. The side panels of this manipulator are held rigid by cross members and linked by an aluminum plate. This construction technique provides lightweight while contributing to the rigidity of the robot, which in part allows for the high speed and accuracy of the system. As the gravity forces are not counterbalanced, motors for vertical joints are equipped with automatic brakes to prevent the collapse of the manipulator configuration if the motor power supply at each joint is interrupted.
CRS255 has five revolute joints powered by five DC motors. At each joint, an incremental encoder is employed for joint position measurement. The robot system comes with a CRS-C500 control box, which contains five PID feedback controllers, operating at each motor and their structure cannot be modified. In order to implement the learning control strategy to CRS robotic system, one has to by-pass the CRS-C500 controller through the Quanser open-architecture control mode which allows the designer to use Simulink for real-time control implementations. In order to do this job,

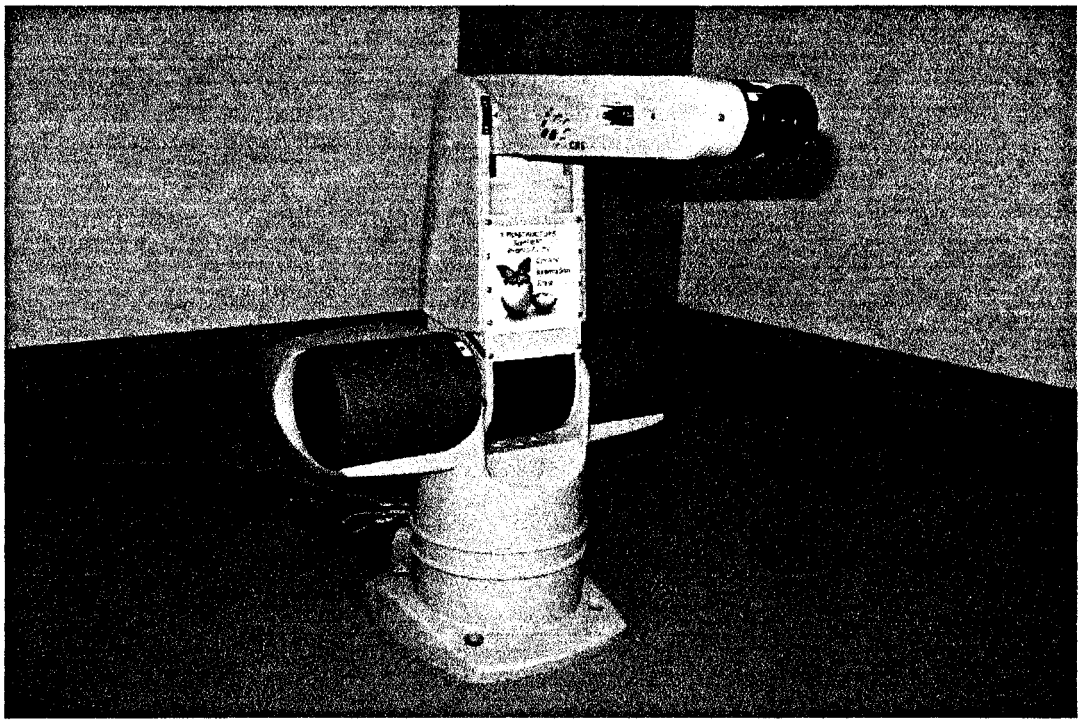

Figure 6.1: 5-DOF CRS255 industrial robot manipulator

a Quanser-MultiQ acquisition board is used together with a Quanser-WinCon software to generate real-time code from Simulink. A switch mounted on the CRS-C500 control box allows the designer to switch back and forth from the Quanser-open-architecture mode to the CRS mode. The following components are installed in the host/supervisor Pentium III PC: MATLAB/Simulink/Realtime Workshop/Control systems toolbox, WinCon, Visual C++ professional, RTX and NT or Windows 2000.

6.2 Real-time implementation

In this thesis, several classical and adaptive ILC algorithms are implemented and evaluated on the 5-DOF CRS255 robot manipulator. The implementation block diagram is depicted in figure 6.2. The experimental results will be presented in chapter 7.

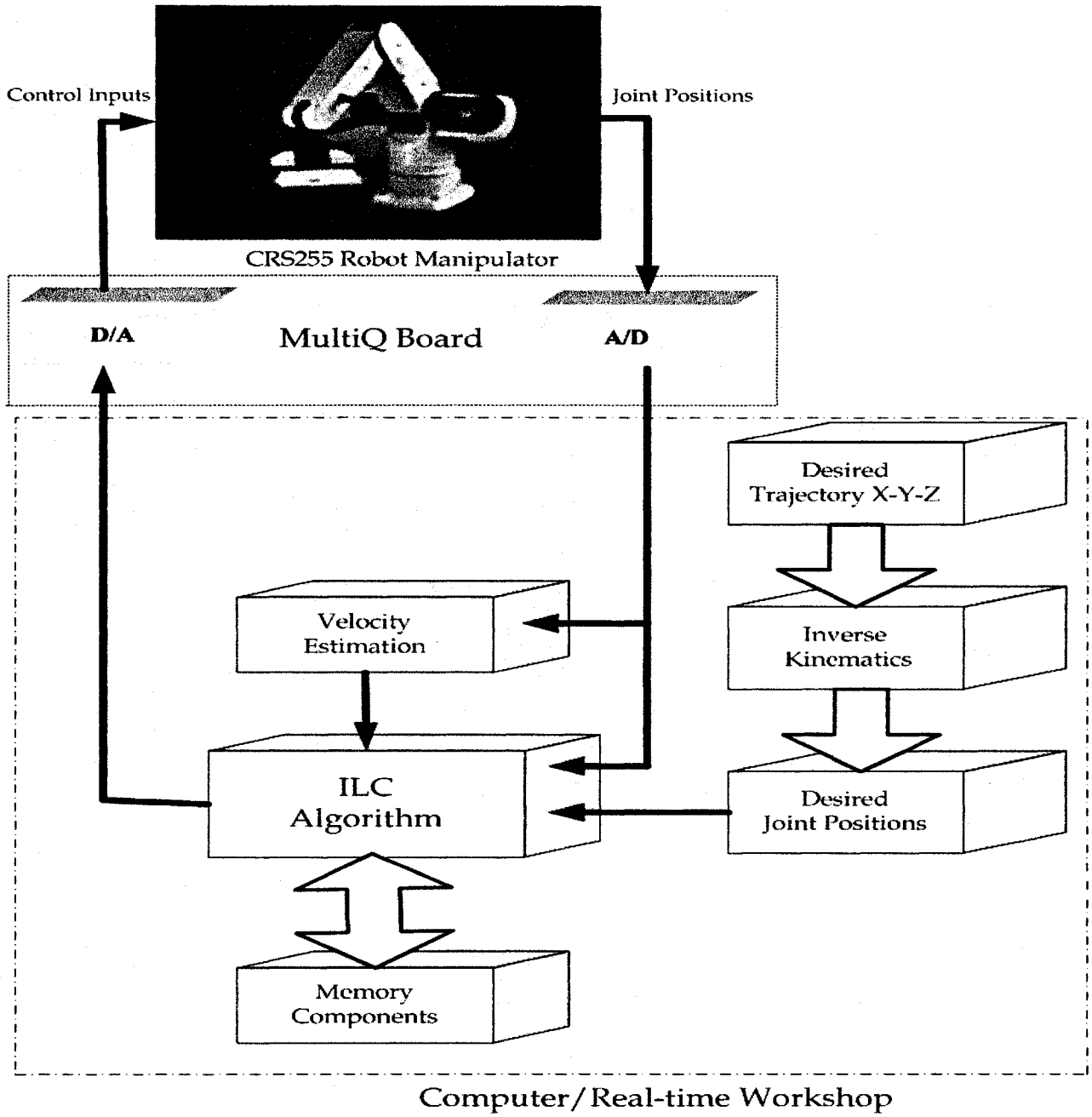

Figure 6.2: ILC implementation block diagram on a 5DOF CRS255 robot manipulator

Chapter 7

Implementation Results

In this chapter, the presented ILC schemes for robot manipulators are tested experimentally on the robot system CRS255 (CATALYST5) shown in figure 6.1.

7.1 Classical ILC approach

The classical ILC schemes discussed in the thesis are implemented on the 5-DOF CRS255 robot manipulator in order to enhance its tracking performance from operation to operation. Generally, the repeatable control environment implies an identical control task and the same initialization condition for all repeated control operations, i.e., $q_d(0) - q_k(0) = \dot{q}_d(0) - \dot{q}_k(0) = 0$.
The control objective is to track an identical circular trajectory in the Y-Z plane as shown in figure 7.1, which can be described as

$$\begin{aligned} y(t) &= 6 + \sqrt{72}\cos 0.1t \ mm \\ z(t) &= 6 + \sqrt{72}\sin 0.1t \ mm \end{aligned}$$

with $t \in [0, 63s]$. In order to realize this desired control trajectory, robot joints 1, 2 and 3 are used. The desired joint variables $q_d^1(t)$, $q_d^2(t)$ and $q_d^3(t)$ corresponding to the control task for the first three joints are obtained by taking the inverse kinematics of the world cordinates $x(t)$, $y(t)$, and $z(t)$. Under a repeatable control environment, the robot dynamics is assumed to be repetitive for all $t \in [0, 63]$ in the sense that the robot parameters never change when the robot repeats the same control tasks.

In the implementation, the joint position signals are available from sensors for feedback and the velocity signals are not measured but estimated because the robot system has no velocity sensors.

Algorithm 1

Two experimental results with the algorithm 1 are shown in figures 7.2 to 7.9. The sampling period is taken as 0.01s. The control and learning parameters used in the experimental tests are listed in table 1A and table 1B. The output tracking error convergence of robot joint positions are investigated in the sense of the various norm-topologies.

Table 1A			
Parameters	Link 1	Link 2	Link 3
K_P	2.5	1.2	2.2
K_D	0.0005	0.0005	0.0005
Learning gain Γ	2	2	3

Table 1B			
Parameters	Link 1	Link 2	Link 3
K_P	2	1	2
K_D	0.0005	0.0005	0.0005
Learning gain Γ	2	2	2

Observation

- The P-type ILC is simple in structure and easy to implement in real-time applications. However, in our several experimental results, we noticed that the convergence of the tracking error is not monotonic. We also noticed the overshoot at the beginning of the robot trajectory after certain iteration which forces the learning process to stop.

Algorithm 2

The learning algorithm 2 is tested experimentally on the same 5-DOF robot manipulator in order to track a desired circular trajectory under a repeatable control environ-

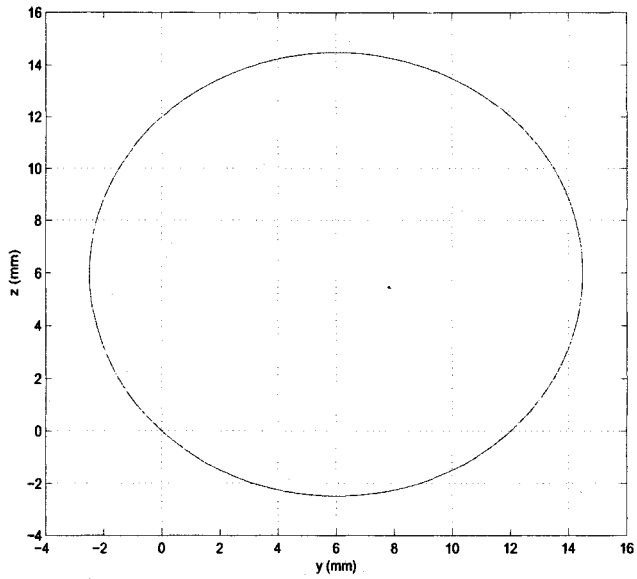

Figure 7.1: Desired tracking trajectory for the experimental study of classical ILC algorithms

ment. To implement this learning control technique, one has to estimate the velocity signals using the filtered derivative i.e., $\dot{q}_k$ is replaced by $\frac{s}{1+\lambda_f s} q_k$, with $\lambda_f = \frac{1}{2\pi f_c}$ in the control law. In this implementation, the cutoff-frequency of the low-pass filter for the three joints is taken as $f_{c1} = f_{c2} = f_{c3} = 0.1592Hz$. The desired trajectory is a circle shown in figure 7.1. The saturation function is taken as

$$sat(e_i) = \begin{cases} e_i & \text{if } \mid e_i \mid < 0.1 \\ 1 & \text{if } e_i \geq 0.1 \\ -1 & \text{if } e_i \leq -0.1 \end{cases} \tag{7.1}$$

Table 1			
Parameters	Link 1	Link 2	Link 3
K_P	2	2	2.5
K_D	0.005	0.005	0.005
Learning gain α	1	1	1
Learning gain ϕ	1	1	1

Applying the algorithm 2, with the control and learning parameters listed in table 1 and table 2, we obtain the two experimental results shown in figures 7.10 to 7.14. The

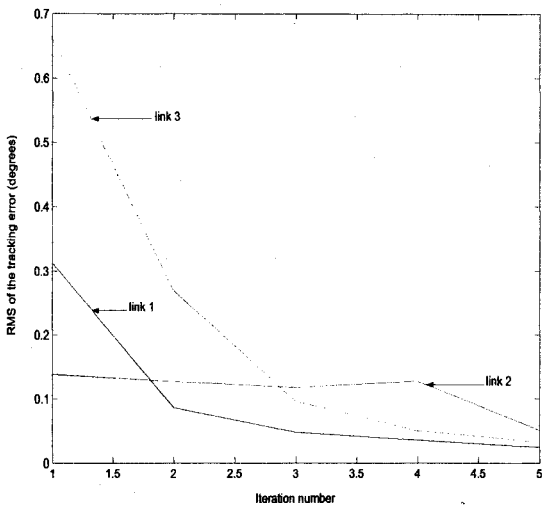

Figure 7.2: RMS of the tracking error versus the iteration number for joints 1, 2, and 3 under the P-type scheme with the parameters listed in table 1A

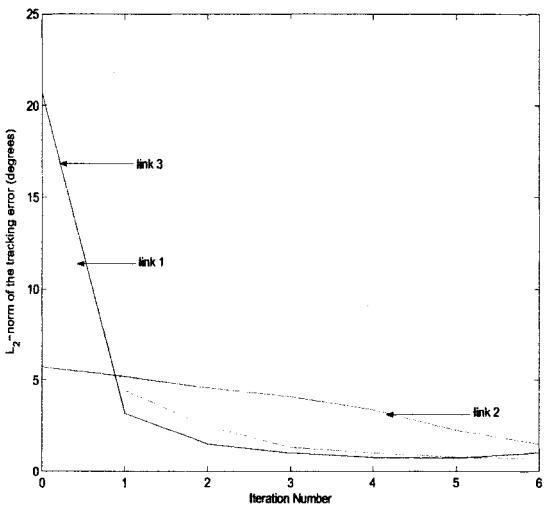

Figure 7.3: L_2-norm of the tracking error versus the iteration number for joints 1, 2, and 3 under the P-type ILC scheme with the parameters listed in table 1B

sampling period is taken as 0.01s. The output tracking error convergence of robot joint positions are investigated in the sense of the L_2-norm and the sup-norm.

Table 2			
Parameters	Link 1	Link 2	Link 3
K_P	2	2	3
K_D	0.5	0.5	0.5
Learning gain α	2	2	2
Learning gain ϕ	1	1	1

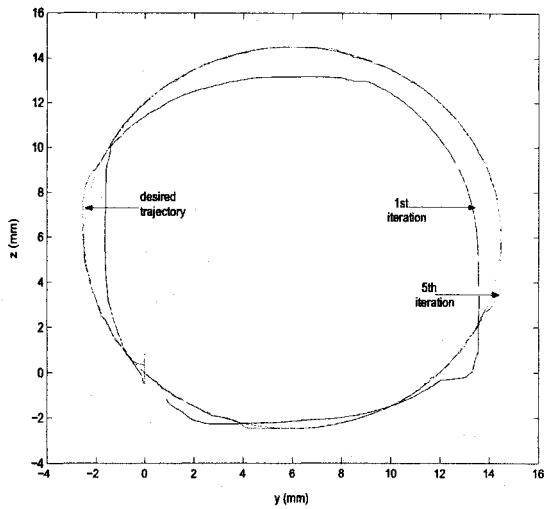

Figure 7.4: Desired trajectory and robot trajectory at the 1st and the 5th iteration under the P-type ILC scheme with the parameters listed in table 1A

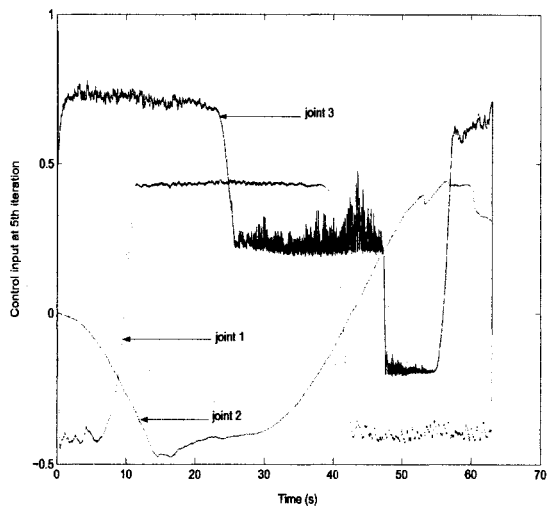

Figure 7.5: control input for joints 1, 2 and 3 at the 5th iteration under the P-type ILC algorithm with the parameters listed in table 1A

Observation

- From our implementation results, we observed that the convergence of the tracking error is not monotonic. We also noticed the overshoot at the beginning of the robot trajectory after certain iteration which forces us to stop the learning process.

D-type ILC scheme

In order to study the effectiveness and feasibility of the D-type ILC scheme in practice, this learning algorithm is tested experimentally on the 5-DOF CRS255 robot manipulator tracking a circular trajectory repeatedly over a fixed time interval $[0, 63s]$. In this implementation, velocity signals in the control and learning laws are not measured but estimated using the filtered derivative($\dot{q}_k$ is replaced by $\frac{s}{1+\lambda_f s} q_k$ with $\lambda_f = \frac{1}{2\pi f_c}$). The acceleration signals are also estimated twice from the joint positions using filtered derivatives. In this design, the cutoff-frequency for the three joints is taken as $f_{c1} = f_{c2} = f_{c3} = 0.1592Hz$.

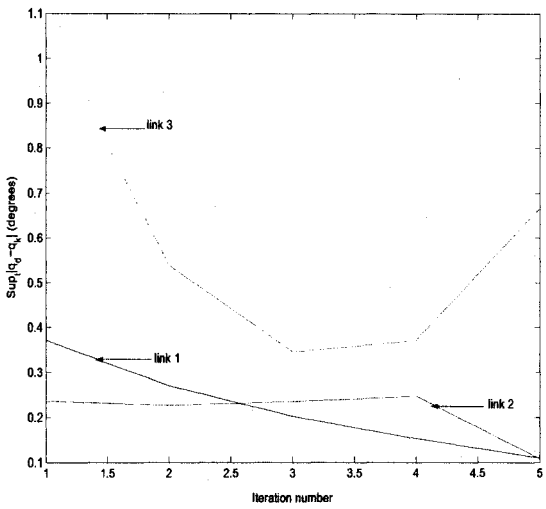

Figure 7.6: Sup-norm of the tracking error versus the number of iterations for link 1, 2 and 3 under P-type ILC scheme with the parameters listed in table 1A

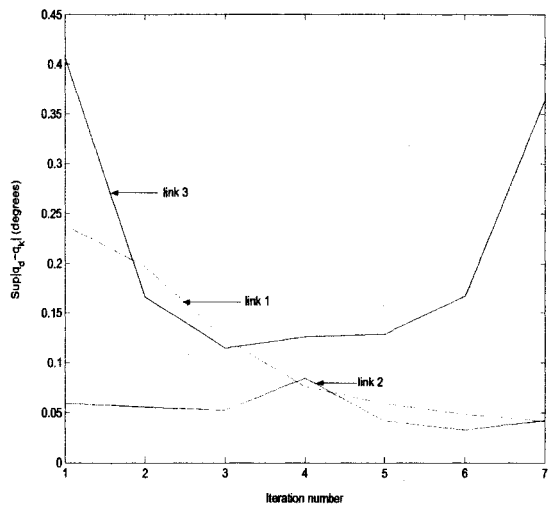

Figure 7.7: Sup-norm of the tracking error versus the number of iterations for link 1, 2 and 3 under P-type ILC scheme with the parameters listed in table 1B

Experiment 1:

Using the D-type learning control scheme, with the parameters listed in table 3, the experimental results shown in figures 7.15 to 7.17 are obtained. The sampling period is taken as 0.01s. The convergence of the output tracking error is investigated in the sense of the L_2-norm and the sup-norm topology.

Table 3			
Parameters	Link 1	Link 2	Link 3
K_P	1	1	3
K_D	1	1	1
Learning gain Γ	0.9	0.9	0.9

Experiment 2:

The objective of this test is to track a desired trajectory only for the first joint of the 5-DOF robot manipulator. The desired joint position trajectory for this test is

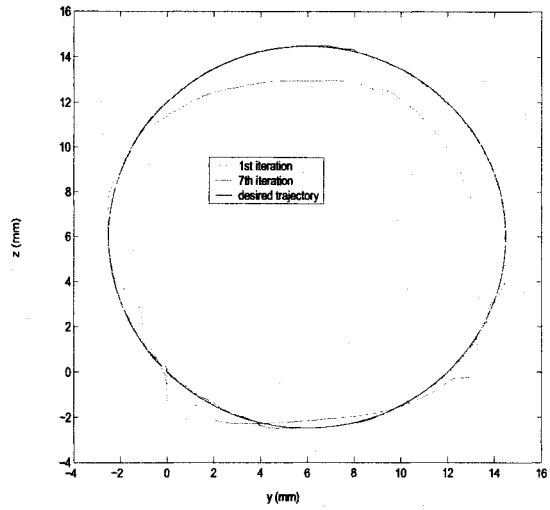

Figure 7.8: Desired trajectory and robot trajectory at the 1st and the 7th iteration under P-type ILC scheme with the parameters listed in table 1B

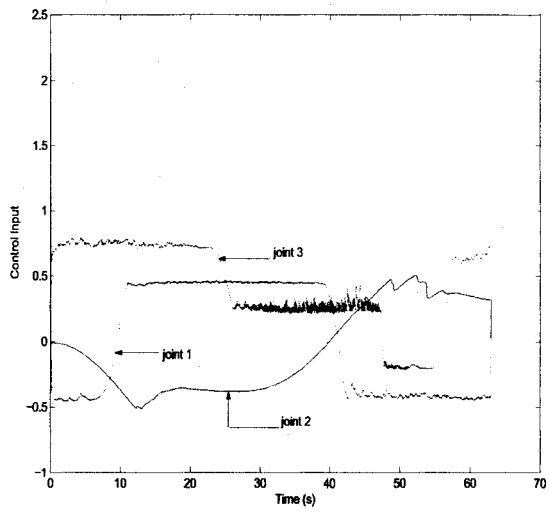

Figure 7.9: Control input for joints 1, 2 and 3 at the 7th iteration under the P-type ILC scheme with the parameters listed in table 1B

given by $q_1^d(t) = r \sin \omega(t)$ with $r = 1$ and $\omega = \frac{2\pi}{T}$. The time length of this control task is taken as $6s$. Using the D-type ILC algorithm, with $K_p = 1$, $K_D = 0.005$ and $\Gamma = 0.5$, the experimental results are shown in figures 7.18 to 7.20. The sampling period is taken as 0.01s. The convergence of the tracking error is investigated in various norm-topologies.

Observation

- In this design, the joint accelerations are estimated from the joint position using the filtered derivative which leads to an amplification of the measurement noise. As a result, the measurement noise reduces the effectiveness of the D-type ILC scheme in real-time applications (Heinzinger *et al.* 1992, Oh *et al.* 1994 and Wang and Cheah 1998).

- The convergence proof of the D-type ILC scheme is based upon the use of the global Lipschitz condition (GLC). In fact, for robotic applications, this condition is not so restrictive since it is possible to choose many Γ's owing to symmetric, boundness and positive definiteness property of the inertia matrix.

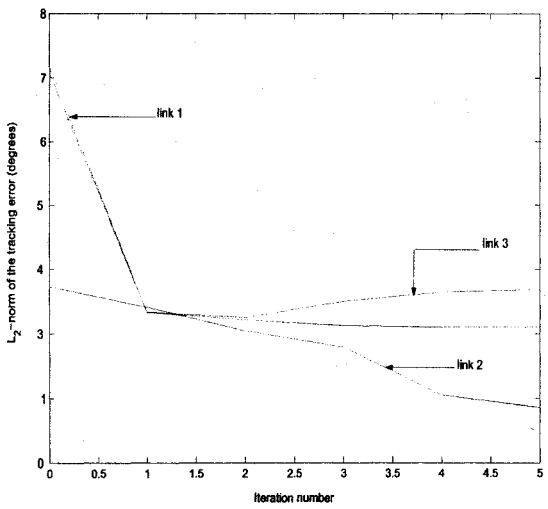

Figure 7.10: L_2-norm of the tracking error versus the iteration number for link 1, 2 and 3 under the SP-D type ILC with the parameter listed in table 1

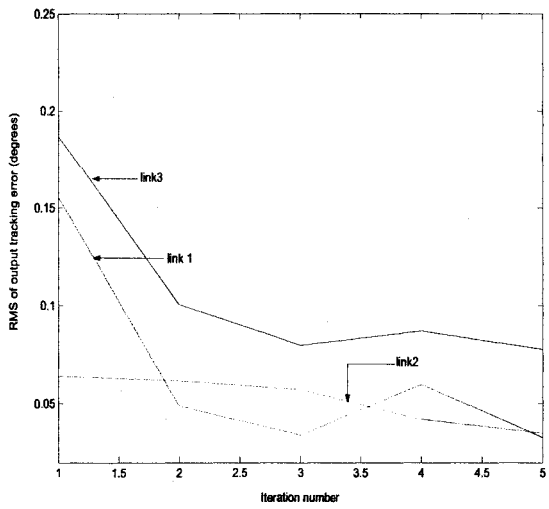

Figure 7.11: L_2-norm of the tracking error versus the iteration number for link 1, 2 and 3 under the SP-D type ILC with the parameter listed in table 2

However, the convergence analysis of most existing D-type learning schemes are starightforward and ensures zero tracking errors in the absence of uncertainities and noises.

- In our experimental tests, we noticed that the convergence of the tracking error is not monotonic. The reason of the non-monotonic convergence is closely related with the λ-norm, which plays an important role in the proof of convergence of this learning method. In many control applications like robot tracking, the sup-norm is more appropiate for the measurement of error performance than the λ-norm topology. Therefore, it is important to show the convergence of the iterative process in the sup-norm topology because the absolute magnitude of the variables is of major concern to protect hardware components from failure.

PD-type ILC scheme

The PD learning control algorithm presented in the thesis is tested and evaluated on the same industrial robot manipulator depicted in figure 6.1, which tracks a circular

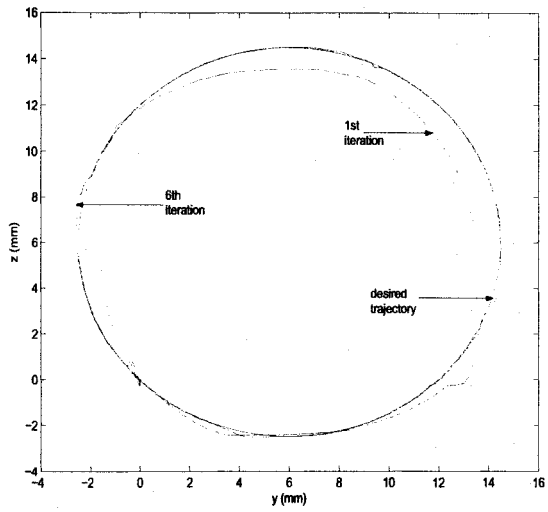

Figure 7.12: Desired trajectory and robot trajectory at the 1st and the 6th iteration under the SP-D type ILC with the parameter listed in table 1

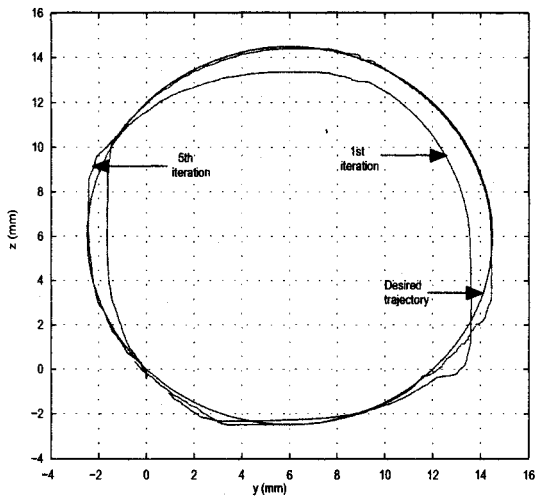

Figure 7.13: Desired trajectory and robot trajectory at the 1st and the 5th iteration under the SP-D type ILC with the parameter listed in table 2

trajectory in the $(Y-Z)$ plane over a finite time interval $[0, 63s]$. The velocity signals in the learning and control laws are replaced by the filtered derivative of the robot joint positions (i.e. $\dot{q}_k$ is replaced by $\frac{s}{1+\lambda_f s}q_k$ with $\lambda_f = \frac{1}{2\pi f_c}$, where cutoff-frequency $f_c = 0.1592Hz$). The implementation results of the PD-type ILC scheme, with the parameters listed in table 4, are shown in figures 7.21 to 7.24. The convergence of the L_2-norm and the sup-norm of the tracking error for joints 1, 2, and 3 are investigated.

Table 4			
Parameters	Link 1	Link 2	Link 3
K_P	2	2	3
K_D	0.5	0.5	0.5
Learning gain Γ	2	2	3
Learning gain ϕ	0.1	0.1	0.1

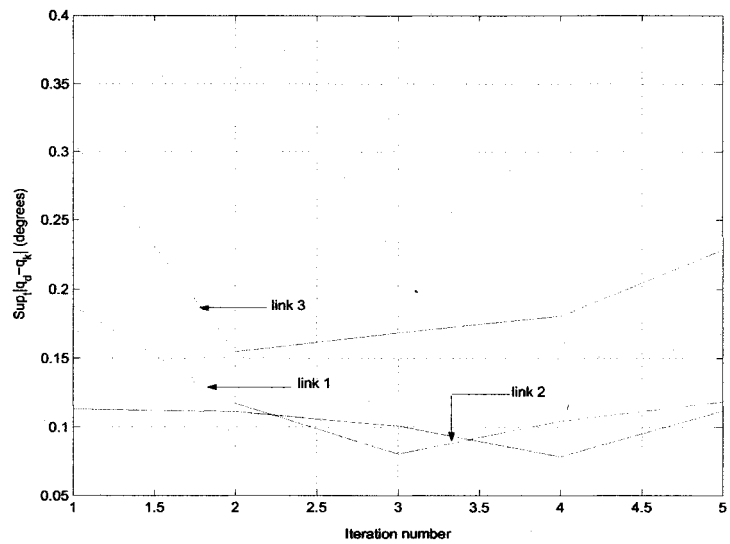

Figure 7.14: Sup-norm of the tracking error versus the number of iterations for link 1, 2 and 3 under SP-D type ILC scheme with the parameters listed in table 2

Observation

- From our experimental tests, we observed that the convergence of the tracking error of the PD-type ILC scheme is not monotonic in the sense of the L_2-norm and the sup-norm. This is due to the fact that the convergence analysis of this iterative process is based upon the use of the λ-norm.

7.2 Concluding remarks on classical ILC approach for robot manipulators

ILC is now a powerful control technique for repetitive processes. Various ILC schemes have been proposed in the literature, since the early 1980's. Despite much research, very few ILC schemes have been applied in commercial products. For practical applications in the real world, however, a number of limitations found in the existing ILC algorithms need to be dealt with.

- In some ILC schemes the tracking error can increase quite largely before it converges to the desired level even if the convergence and stability conditions

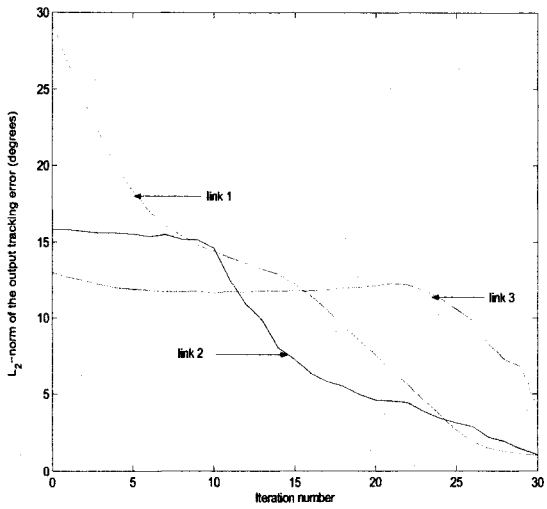

Figure 7.15: The L_2-norm of the tracking error as a function of the iteration number for link 1, 2 and 3 under D-type ILC scheme with the parameter listed in table 3

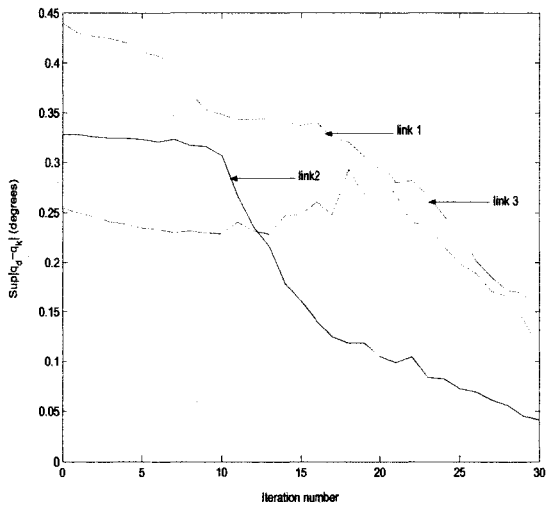

Figure 7.16: Sup-norm of the tracking error versus the iteration number for link 1, 2 and 3 under the D-type ILC scheme with the parameter listed in table 3

are satisfied (Jang and Longman 1994 and Longman 2000). This undesirable phenomenon comes from the fact that the λ-norm is generally used to prove the convergence of the iterative process. In real-time applications, the λ-norm is not a satisfactory measure of error performance because a large overshoot of the tracking error may occur before it converges to the desired level which may cause system hardware failure. A possible solution to make the convergence monotonic in sup-norm (Owens 1992) is to use a high-gain feedback. However, this is not a practical solution because high-gain may saturate the actuator dynamics. There are very few theoretical results that ensure monotonic convergence of the iterative process in the literature (Ishihara *et al.* 1992, Hillenbrand and Pandit 2000 and Tayebi and Zaremba 1999). Therefore, it is important to analyze the convergence property in the sense of the sup-norm because the absolute magnitude is more adequate as an error measurement of performance than the λ- norm.

- Most existing ILC frameworks are based upon the use of the contraction mapping approach and require a certain *a priori* knowledge of plant dynamics.

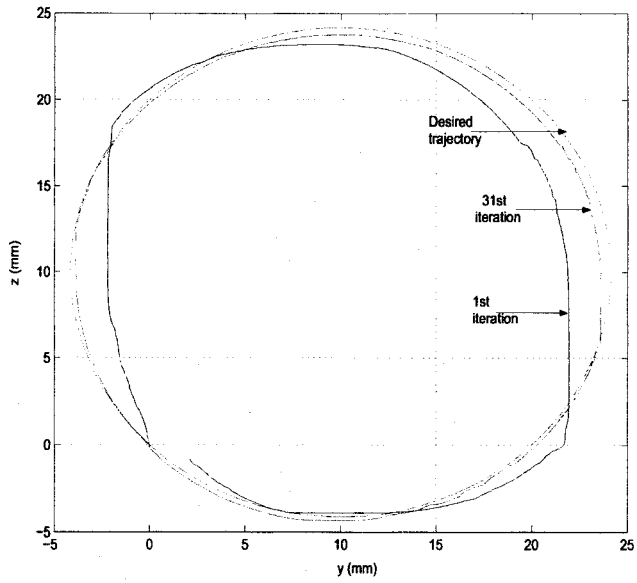

Figure 7.17: Desired trajectory and robot trajectories at the 1st and the 31st iteration under the D-type ILC algorithm with the parameter listed in table 3

Some of the limitations of traditional ILC algorithms have been discussed in Xu (2002).

- The number of iterative variables used in existing classical ILC methods for robot manipulators is equal to the number of degrees-of-freedom n. As a result, large memory components and high computing power are essential to store past control information in real-time applications.

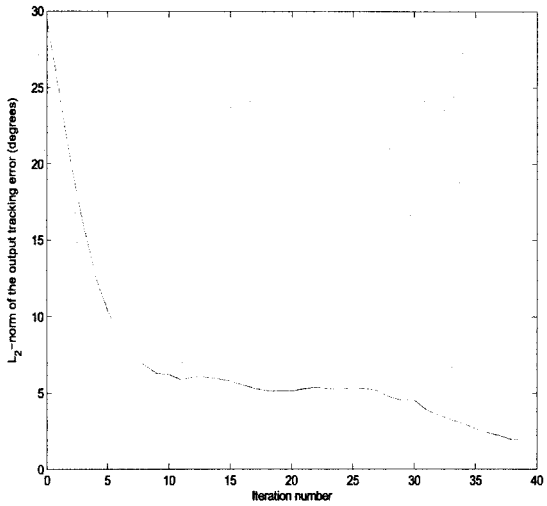

Figure 7.18: The L_2-norm of the tracking error as a function of the iteration number for link 1 with the D-type ILC scheme with $K_p = 1$, $K_D = 0.005$ and $\Gamma = 0.5$,

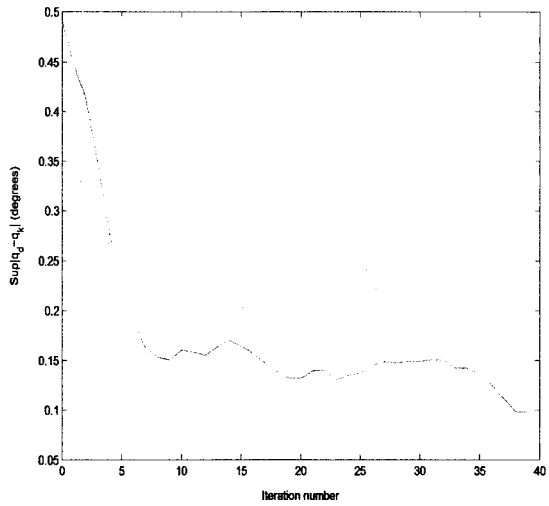

Figure 7.19: Sup-norm of the tracking error versus the iteration number for link 1 under the D-type ILC scheme with $K_p = 1$, $K_D = 0.005$ and $\Gamma = 0.5$,

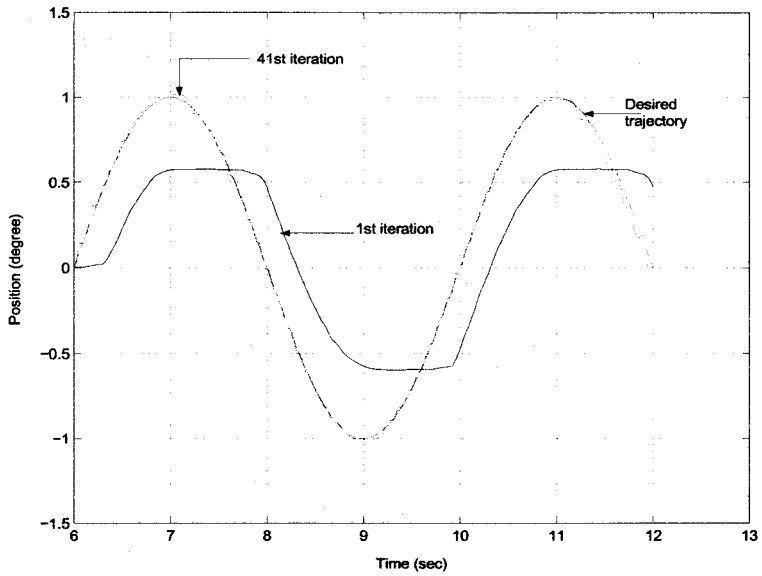

Figure 7.20: Desired trajectory and position trajectories of the robot joint 1 at the 1st and 41st iteration under the D-type ILC with $K_p = 1$, $K_D = 0.005$ and $\Gamma = 0.5$

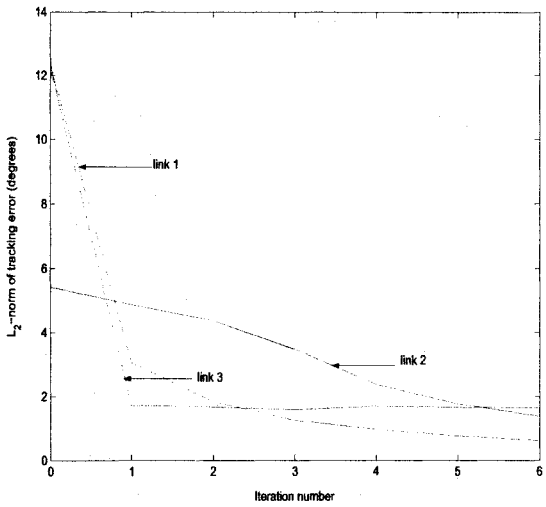

Figure 7.21: L_2-norm of the tracking error versus the iteration number for link 1, 2 and 3 under the PD-type ILC scheme

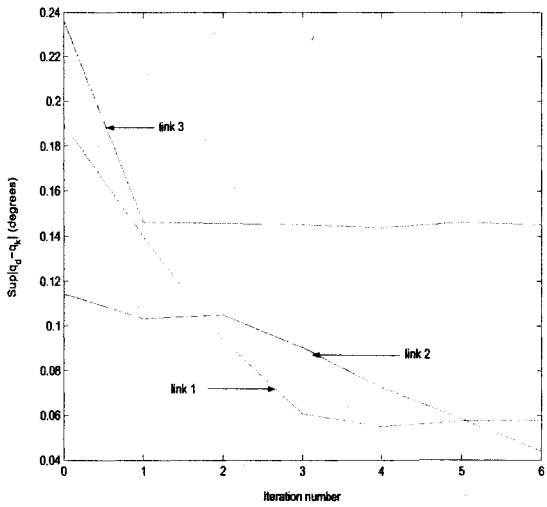

Figure 7.22: Sup-norm of the tracking error versus the iteration number for link 1, 2 and 3 under the PD-type ILC scheme

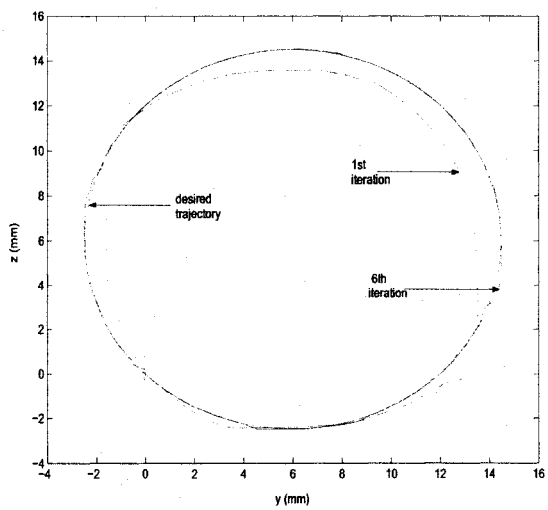

Figure 7.23: Desired trajectory and robot tracking trajectory at the 1st and the 6th iteration under PD-type ILC scheme

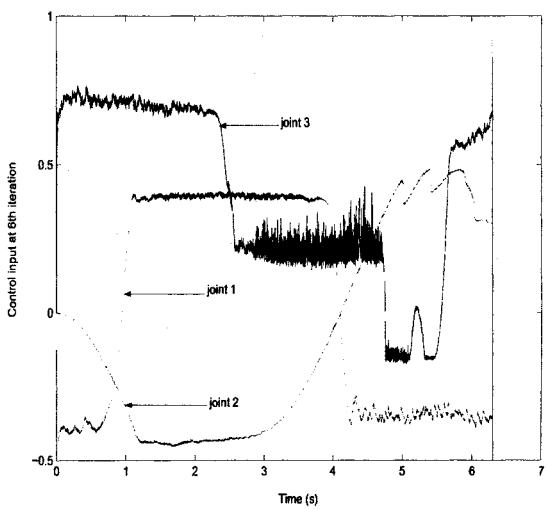

Figure 7.24: Control input for joints 1, 2 and 3 at the 6th iteration with the PD-type ILC algorithm

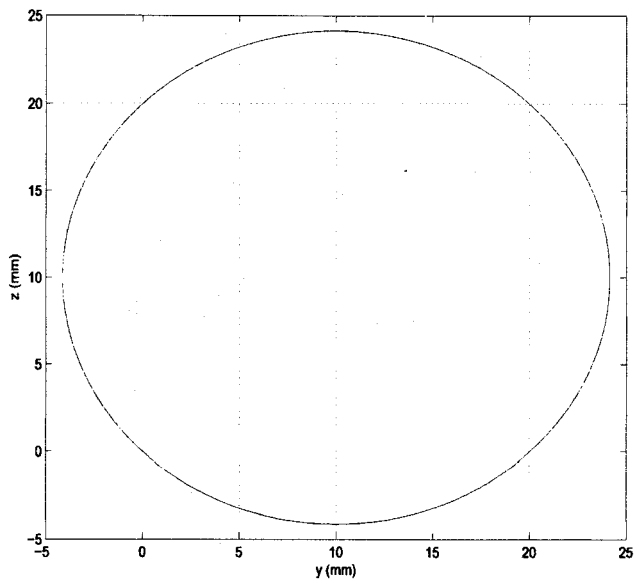

Figure 7.25: Desired tracking trajectory for the experimental evaluation of the Adaptive ILC approach

7.3 Adaptive ILC approach

In this section, the four adaptive ILC algorithms for robot manipulators presented in the thesis are tested experimentally on the 5-DOF CRS255 robot system. The obtained experimental results are compared with experimental results of classical ILC algorithms. The main target of AILC experiments is to track a circular trajectory in the (Y-Z) plane under a repeatable control environment over the finite time interval $[0, 63s]$ which is shown in figure 7.25. This desired circular trajectory can be expressed as

$$\begin{aligned} y(t) &= 10 + \sqrt{200}\cos 0.1t \ mm \\ z(t) &= 10 + \sqrt{200}\sin 0.1t \ mm \end{aligned}$$

Only the first three joints of the robot manipulator are used to realize this circular motion. The desired joint positions $q_d^1(t)$, $q_d^2(t)$ and $q_d^3(t)$ are generated by taking the inverse kinematics of the world co-ordinates $x(t) = 0$, $y(t)$ and $z(t)$. The initial resetting condition, i.e., $q_d(0) - q_k(0) = \dot{q}_d(0) - \dot{q}_k(0) = 0$ is satisfied for each iteration. In practice, the signum function used in the adaptation law can cause the chattering

phenomenon. In order to avoid this problem and generate a smooth control input, we replaced the signum function, i.e., $sgn(\dot{\tilde{q}}_k)$, by a saturation function defined as follows

$$sat(\dot{\tilde{q}}_k) = [sat(\dot{\tilde{q}}_{1,k}), ..., sat(\dot{\tilde{q}}_{n,k})]^T$$

with

$$sat(\dot{\tilde{q}}_{i,k}) = \begin{cases} \dot{\tilde{q}}_{i,k} & \text{if } \mid \dot{\tilde{q}}_{i,k} \mid < 0.01 \\ 1 & \text{if } \dot{\tilde{q}}_{i,k} \geq 0.01 \\ -1 & \text{if } \dot{\tilde{q}}_{i,k} \leq -0.01 \end{cases} \tag{7.2}$$

where $i \in \{1, 2, 3\}$. In order to implement presented AILC algorithms on the 5-DOF CRS255 robot manipulator, we need to estimate the velocity signal using the filtered derivative in the control and learning laws, i.e., $\dot{\tilde{q}}_k$ is replaced by $\frac{s}{1+\lambda_f s}\tilde{q}_k$ with $\lambda_f = \frac{1}{2\pi f_c}$, where f_c is the cut-off frequency.

Adaptive scheme 1

The presented AILC scheme 1 is tested experimentally on the CRS255 robot manipulator, where the known matrix $\phi(q_k, \dot{q}_k, \dot{\tilde{q}}_k)$ given by the following equation

$$\phi(q_k, \dot{q}_k, \dot{\tilde{q}}_k) = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{15} & \phi_{16} & \phi_{17} & \phi_{18} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} & \phi_{26} & \phi_{27} & \phi_{28} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} & \phi_{35} & \phi_{36} & \phi_{37} & \phi_{38} \end{bmatrix}. \tag{7.3}$$

with

$\phi_{11} = -\cos q_2 \sin(q_2 + q_3)\dot{q}_3 - \sin(2q_2 + q_3)\dot{q}_1,$

$\phi_{12} = -0.5\cos(q_2 + q_3)\sin(q_2 + q_3)\dot{q}_3 - 0.5\cos(q_2 + q_3)\sin(q_2 + q_3)\dot{q}_1,$

$\phi_{13} = 0.5\sin q_2 \cos q_2 \dot{q}_1,$

$\phi_{14} = -2\sin q_2 \cos q_2 \dot{q}_1,$

$\phi_{15} = \phi_{16} = \phi_{17} = 0,$

$\phi_{18} = sgn(\dot{\tilde{q}}_1),$

$\phi_{21} = 0.5\sin(2q_2 + q_3)\dot{q}_1 - 1.5\sin q_3 \dot{q}_3,$

$\phi_{22} = 0.25\cos(q_2 + q_3)\sin(q_2 + q_3)\dot{q}_1,$

$\phi_{23} = -0.25\cos q_2 \sin q_2 \dot{q}_1,$

$\phi_{24} = \cos q_2 \sin q_2 \dot{q}_1,$

$\phi_{25} = -0.5\sin q_2,$

$$\phi_{26} = -\sin q_2,$$
$$\phi_{27} = -0.5\sin(q_2 + q_3),$$
$$\phi_{28} = sgn(\dot{\tilde{q}}_2),$$
$$\phi_{31} = 0.5\cos q_2 \sin(q_2 + q_3)\dot{q}_1 + 0.5\sin q_3 \dot{q}_2,$$
$$\phi_{32} = 0.25\cos(q_2 + q_3)\sin(q_2 + q_3)\dot{q}_1,$$
$$\phi_{33} = \phi_{34} = \phi_{35} = \phi_{36} = 0,$$
$$\phi_{37} = -0.5\sin(q_2 + q_3),$$
$$\phi_{38} = sgn(\dot{\tilde{q}}_3).$$

Experiment 1:

Using the adaptive scheme 1, we obtained the experimental results shown in figures 7.26 to 7.28. The cutoff-frequency for the velocity estimations are taken as $f_{c1} = f_{c2} = f_{c3} = 0.08Hz$. The sampling period is taken as 3 ms. The learning and control gains used in this experiment are taken as follows: $K_P = diag(1, 1, 4)$, $K_D = 0.05I_{3\times3}$ and $\Gamma = 2I_{8\times8}$, where $I_{i\times i}$ is an $i \times i$ identity matrix.

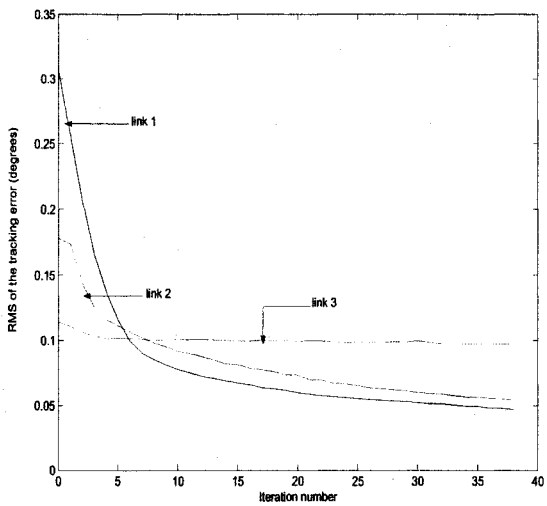

Figure 7.26: RMS of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 1 with the 1st experimental test

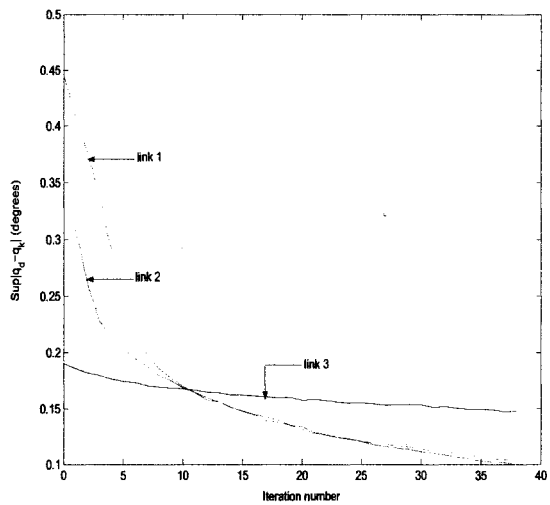

Figure 7.27: Sup-norm of the tracking error versus the iteration number for link 1, 2 and 3 under adaptive learning scheme 1 with the 1st experimental test

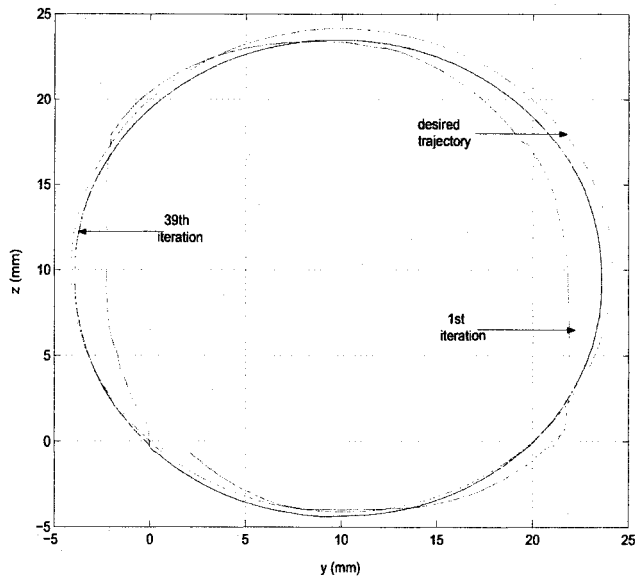

Figure 7.28: Target trajectory and robot trajectory at the 1st and the 39th iteration under the adaptive learning scheme 1 with the 1st experimental test

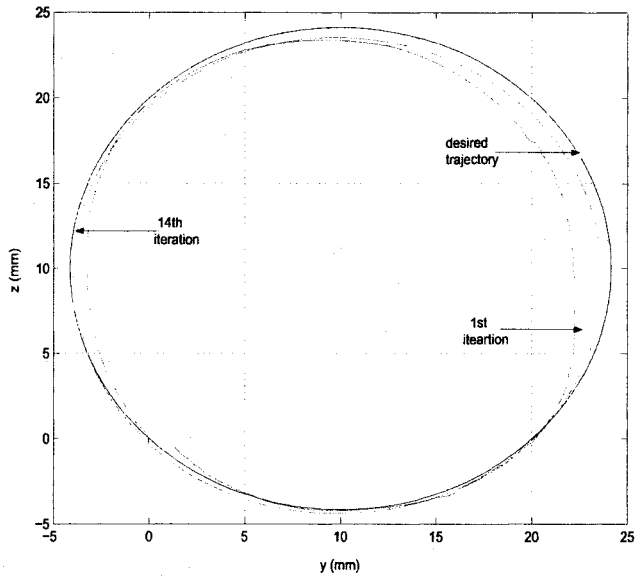

Figure 7.29: Target trajectory and robot trajectory at the 1st and the 14th iteration with the adaptive learning control scheme 1 with the 2nd experimental test

Experiment 2:

In order to examine the convergence speed of this adaptive scheme, we performed another experimental test on the same system with the control and learning parameters as follows: $K_P = diag(1, 1, 4)$, $K_D = 0.05 I_{3\times3}$ and $\Gamma = 8 I_{8\times8}$. The implementation results are shown in figures 7.29 to 7.31. The sampling period and the cutoff-frequency for the velocity estimations are taken as 3 ms and $f_{c1} = f_{c2} = f_{c3} = 0.08Hz$ respectively.

Observation

- The control law uses 8 updating parameters. As a result, the real-time implementation requires a considerable amount of memory components and computing power.
- The design also requires the knowledge of the matrix $\psi(q_k, \dot{q}_k, \dot{\tilde{q}}_k)$.

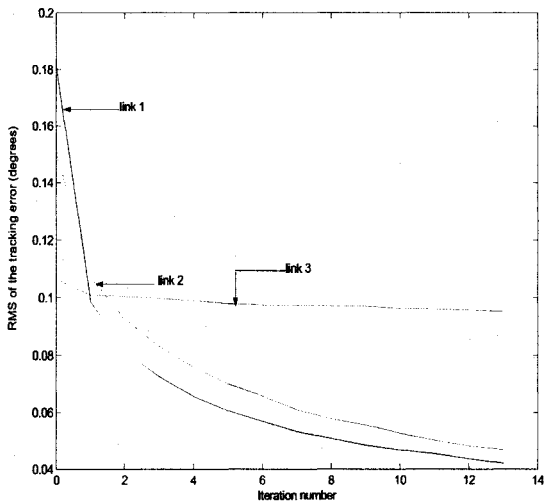

Figure 7.30: RMS of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning control scheme 1 with the 2nd experimental test

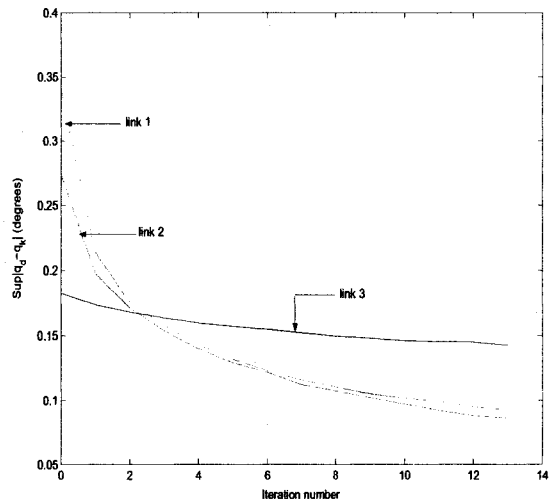

Figure 7.31: Sup-norm of the tracking error versus the iteration number for link 1, 2 and 3 under the adaptive control scheme 1 with the 2nd experimental test

Adaptive scheme 2

Experiment 1:

Using the AILC scheme 2 presented in the thesis, with $K_P = diag(1, 1, 4)$, $K_D = 0.005I_{3\times3}$ and $\Gamma = 3I_{2\times2}$, the experimental results are shown in figures 7.32 to 7.34. In this test, the sampling period and the cutoff-frequency for the velocity estimation are used as follows: 1 ms and $f_{c1} = f_{c2} = f_{c3} = 0.1592Hz$.

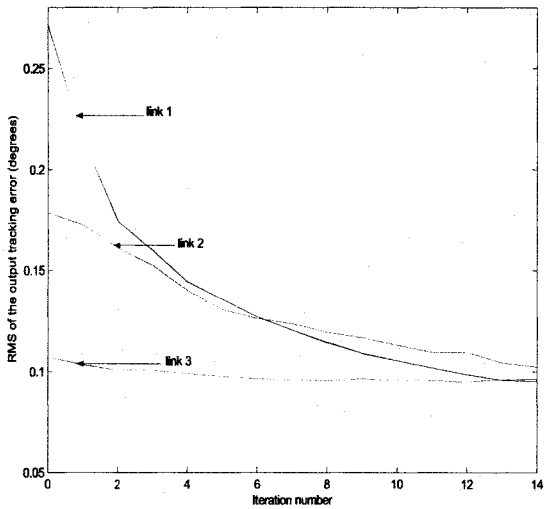

Figure 7.32: The RMS of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 2 with the 1st experimental test

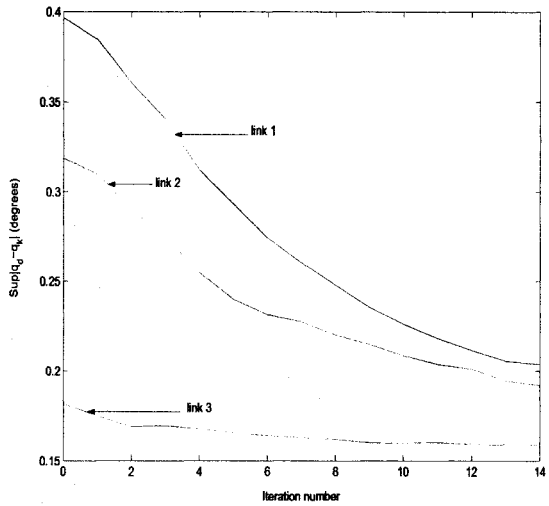

Figure 7.33: Sup-norm of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 2 with the 1st experimental test

Experiment 2:

The control and learning gains for this implementation are taken as follows: $K_P = diag(2, 1, 4)$, $K_D = 0.05I_{3\times3}$ and $\Gamma = 4I_{2\times2}$. The implementation results are shown in figures 7.35 to 7.37 for robot tracking the same circular trajectory over a finite time interval. The time length of the desired control trajectory is taken as 63s with the sampling period of 1 ms. In this test, the cutoff-frequencies used for the velocity estimation for the three robot joints are as follows: $f_{c1} = 0.08Hz$, $f_{c2} = 0.08Hz$ and $f_{c3} = 0.1592Hz$.

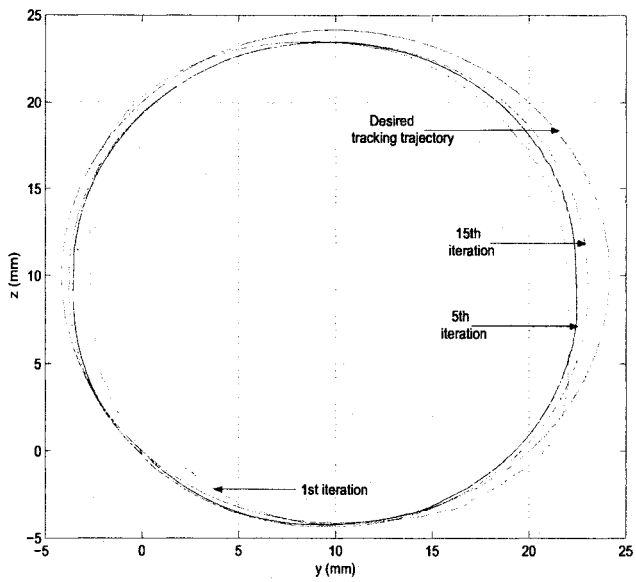

Figure 7.34: Desired trajectory and robot trajectory at the 1st, the 5th and the 15th iteration with the adaptive learning control scheme 2 with the 1st experimental test

Experiment 3:

Using the AILC scheme 2, with the control and learning parameters as follows: $K_P = diag(1, 1, 4)$, $K_D = 0.05 I_{3\times3}$ and $\Gamma = 8 I_{2\times2}$, we obtain the experimental test results shown in figures 7.38 to 7.40 for robot tracking a circular trajectory over a finite time interval $[0, 63s]$. The sampling period is taken as 1 ms. In this test, the cutoff-frequencies used for the velocity estimation for the three joints are as follows: $f_{c1} = 0.08Hz$, $f_{c2} = 0.08Hz$ and $f_{c3} = 0.08Hz$.

Experiment 4:

In order to examine the convergence speed of this adaptive scheme, another experimental test is performed on the same robot system using the control and learning parameters as follows: $K_p = diag(1, 1, 4)$, $K_D = 0.05 I_{3\times3}$ and $\Gamma = 20 I_{2\times2}$. The sampling period and the cutoff-frequency for the velocity estimations are taken as 1 ms and $f_{c1} = f_{c2} = f_{c3} = 0.0177Hz$ respectively. The experimental results are shown in figures 7.41 to 7.43.

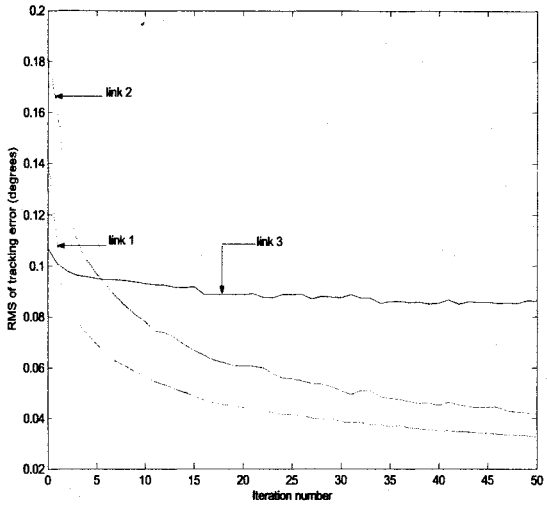

Figure 7.35: RMS of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 2 with the 2nd experimental test

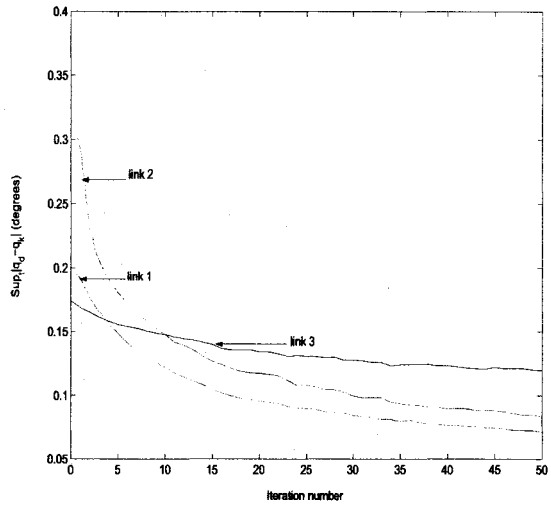

Figure 7.36: Sup-norm of the tracking error versus the iteration number for link 1, 2 and 3 under the adaptive control scheme 2 with the 2nd experimental test

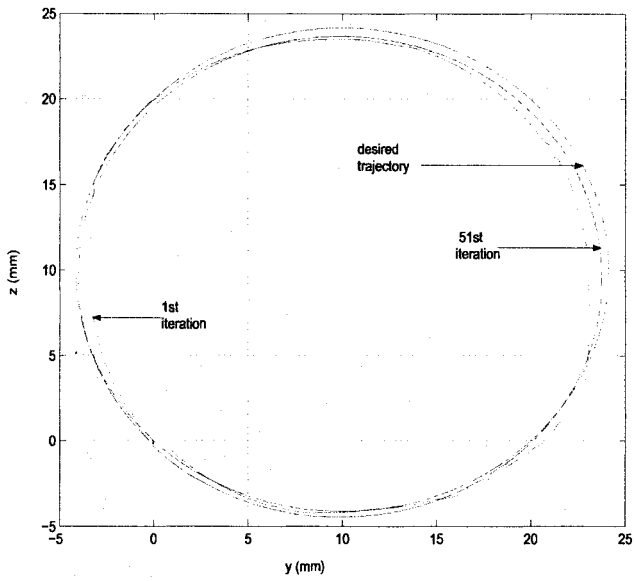

Figure 7.37: Target trajectory and robot trajectory at the 1st and the 51st iteration under the adaptive learning scheme 2 with the 2nd experimental test

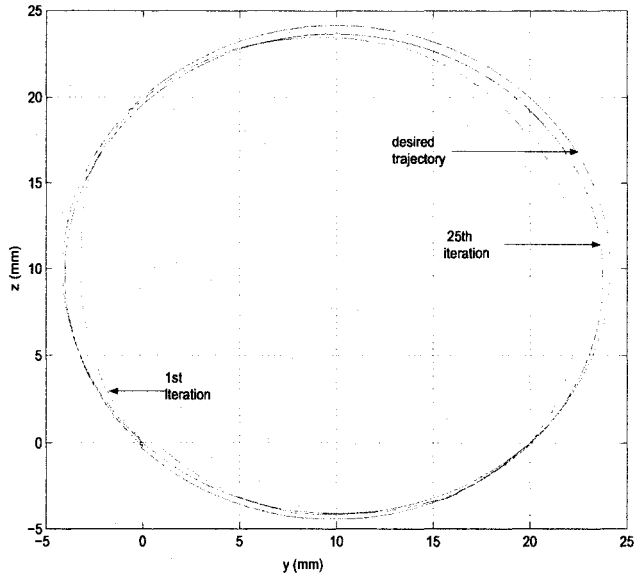

Figure 7.38: Target trajectory and robot trajectory at the 1st and the 25th iteration under the adaptive learning scheme 2 with the 3rd experimental test

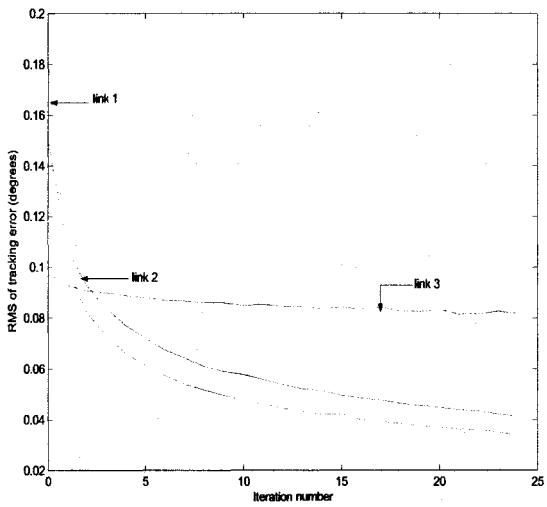

Figure 7.39: RMS of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 2 with the 3rd experimental test

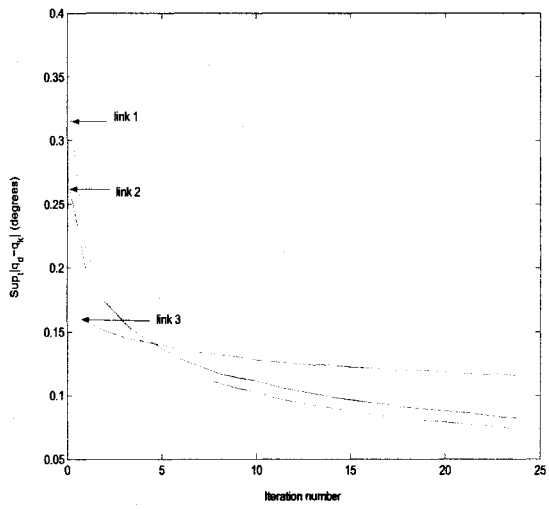

Figure 7.40: Sup-norm of the tracking error versus the iteration number for link 1, 2 and 3 under the adaptive control scheme 2 with the 3rd experimental test

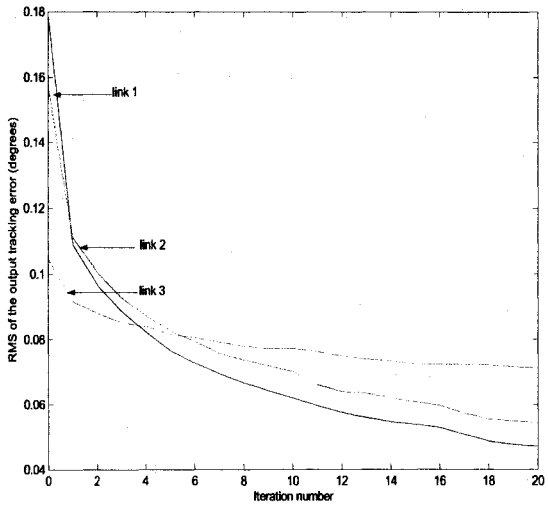

Figure 7.41: The RMS of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 2 with the 4th experimental test

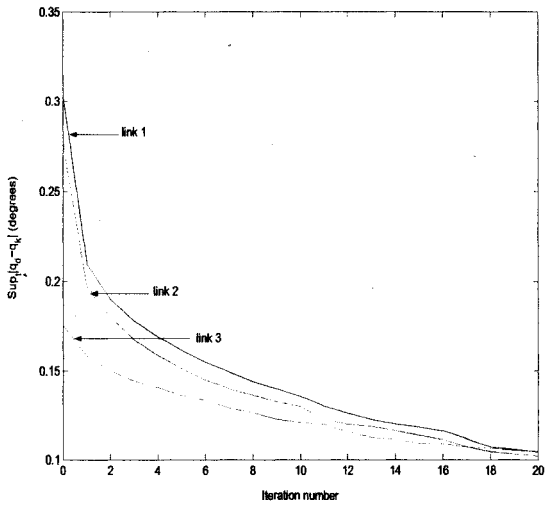

Figure 7.42: Sup-norm of the output tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 2 with the 4th experimental test

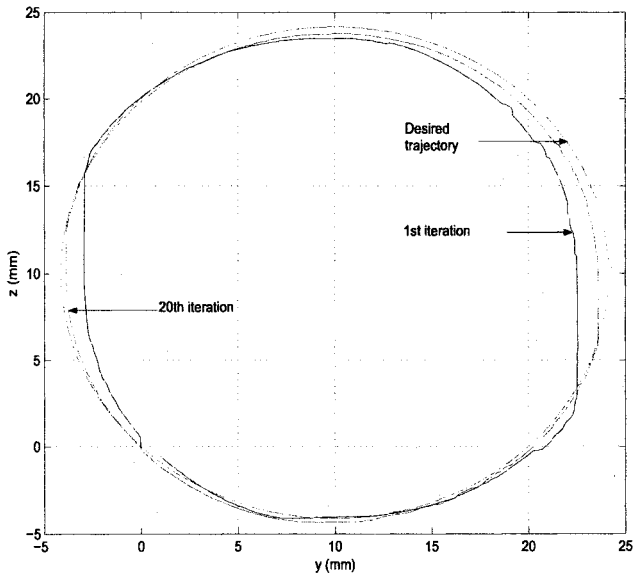

Figure 7.43: Desired trajectory and robot trajectory at the 1st and the 20th iteration under the adaptive learning scheme 2 with the 4th experimental test

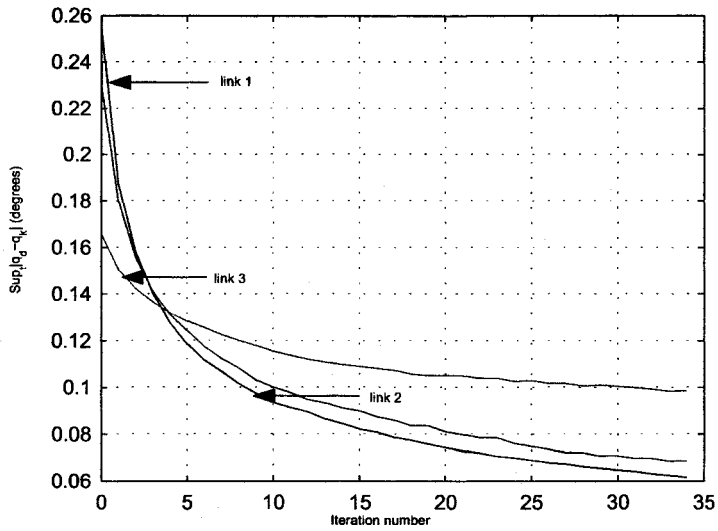

Figure 7.44: The sup-norm of the output tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 2 with the 5th experimental test

Experiment 5:

The control and learning gains for this experiment are taken as follows: $K_P = diag(1, 1, 4)$, $K_D = 0.05 I_{2\times 2}$ and $\Gamma = 20 I_{2\times 2}$. The implementation results are shown in figure 7.44 and 7.45 for robot tracking a circular trajectory. The time length of the desired control task is taken as 63s with the sampling period of 1 ms. In this test, the cutoff-frequencies used for the velocity estimation for the three robot joints are as follows: $f_{c1} = f_{c2} = f_{c3} = 0.02Hz$.

Observation

- The control approach does not require any *a priori* knowledge of the robot dynamics.
- The number of iterative variables used in this design is equal to two which is an interesting fact in real-time applications because it saves memory space and computing power.

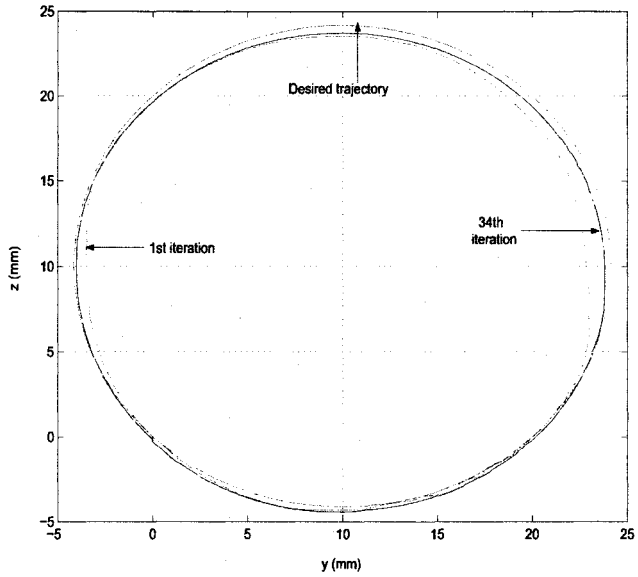

Figure 7.45: Desired trajectory and robot trajectory at the 1st and the 34th iteration under the adaptive learning control scheme 2 with the 5th experimental test

- The acceleration measurements and the bounds of the robot parameters are not needed for this design and implementation.
- The convergence of the iterative process can be guaranteed for any K_P, K_D, Γ symmetric positive definite.

Adaptive Scheme 3

Experiment 1:

The adaptive scheme 3 presented in the thesis is implemented on the 5DOF CRS255 robot system with the control and learning parameters as follows: $K_P = diag(2, 1, 4)$, $K_D = 0.05 I_{3\times3}$ and $\gamma = 8$. The sampling period and the cutoff-frequencies for the velocity estimations are taken as 1 ms and $f_{c1} = f_{c2} = f_{c3} = 0.08Hz$ respectively. The implementation results are shown in figures 7.46 to 7.48.

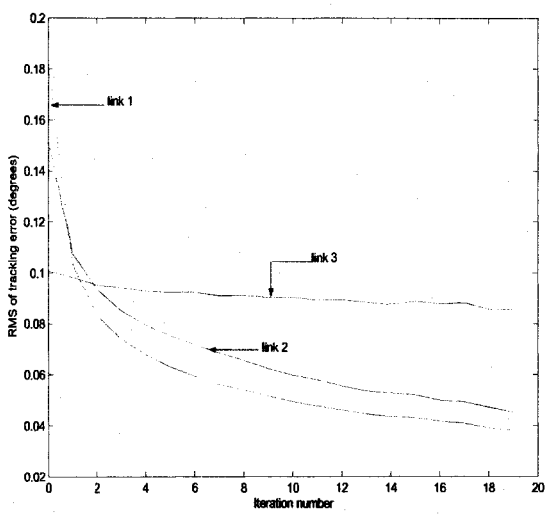

Figure 7.46: RMS of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 3 with the 1st experimental test

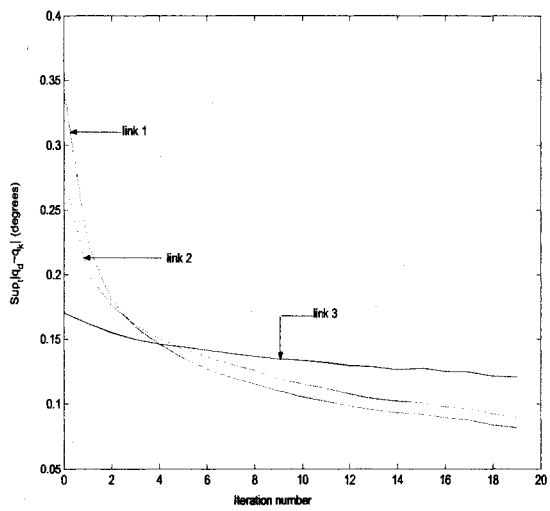

Figure 7.47: Sup-norm of the tracking error versus the iteration number for link 1, 2 and 3 under the adaptive control scheme 3 with the 1st experimental test

Experiment 2:

Using the adaptive scheme 3, with $K_P = diag(1,1,4)$, $K_D = 0.005 I_{3\times 3}$ and $\gamma = 10$, we obtained the experimental results shown in figures 7.49 to 7.51 for the 5-DOF CRS255 robot manipulator tracking a circular trajectory over the finite time interval $[0, 63s]$. In this experiment, the cutoff-frequencies used for the velocity estimations are as follows: $f_{c1} = 0.08Hz$, $f_{c2} = 0.08Hz$ and $f_{c3} = 0.0637Hz$. The sampling period is taken as 1 ms.

Experiment 3:

The control and learning gains for this experimental evaluation are taken as follows: $K_P = diag(1,1,4)$, $K_D = 0.05 I_{3\times 3}$ and $\gamma = 15$. The implementation results are shown in figures 7.52 to 7.54 for the same robot system tracking a circular trajectory over a finite time interval. The time length of the desired control task is taken as 63s. The sampling period and the cutoff-frequencies for the velocity estimations are taken as 1 ms and $f_{c1} = f_{c2} = f_{c3} = 0.0159Hz$ respectively.

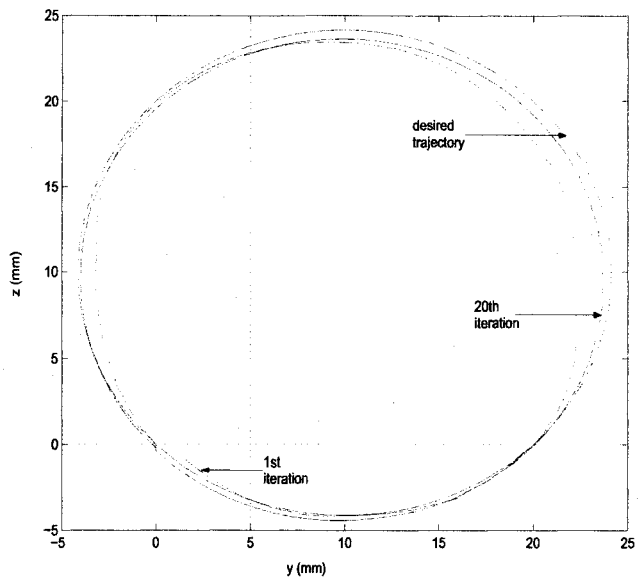

Figure 7.48: Target trajectory and robot trajectory at the 1st and the 20th iteration under the adaptive learning scheme 3 with the 1st experimental test

Experiment 4:

In order to investigate the convergence speed of the AILC scheme 3, we performed another experimental test on the 5-DOF CRS255 robot system with the control and the learning parameters as follows: $K_p = diag(0.5, 0.5, 2)$, $K_D = 0.05 I_{3\times 3}$ and $\gamma = 15$. The sampling period is taken as 1 ms. In this test, the cutoff-frequency used for the velocity estimations is as follows: $f_{c1} = f_{c2} = f_{c3} = 0.08Hz$. The convergence of the output tracking error of the robot joint positions are investigated. The implementation results are shown in figures 7.55 to 7.57.

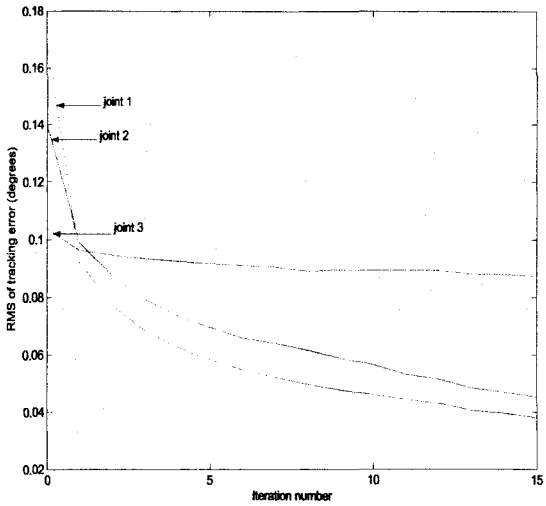

Figure 7.49: RMS of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 3 with the 2nd test

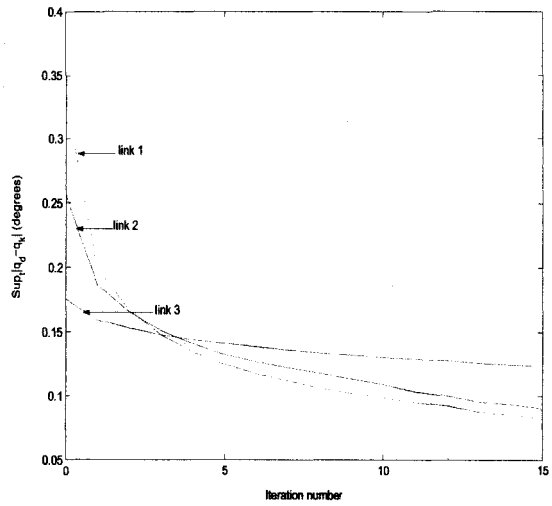

Figure 7.50: Sup-norm of the tracking error versus the iteration number for link 1, 2 and 3 under the adaptive control scheme 3 with the 2nd test

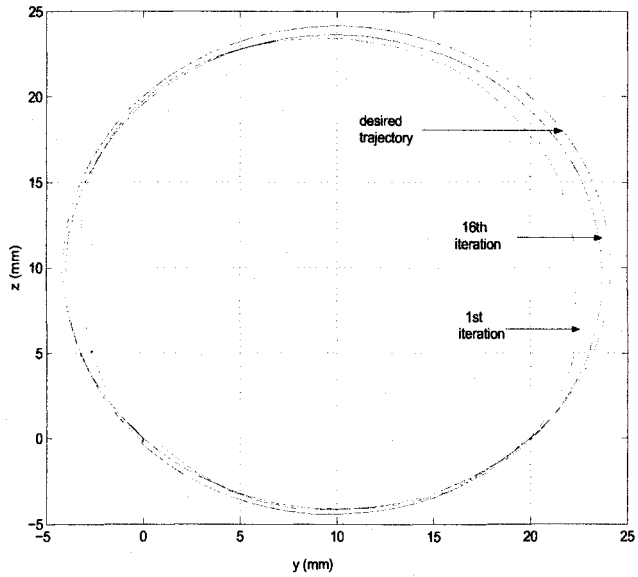

Figure 7.51: Target trajectory and robot trajectory at the 1st and the 16th iteration with adaptive learning scheme 3 with the 2nd test

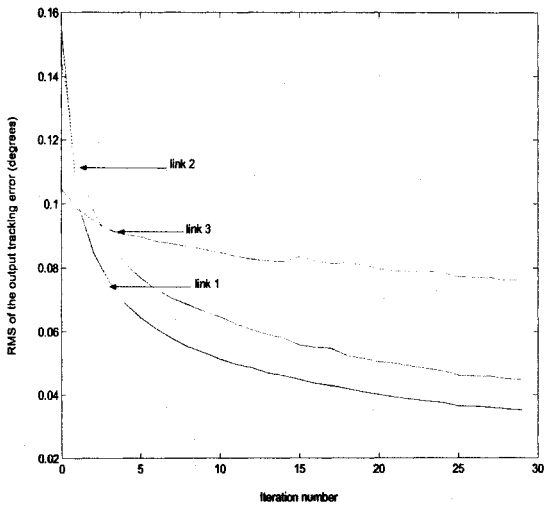

Figure 7.52: The RMS of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 3 with the 3rd experimental test

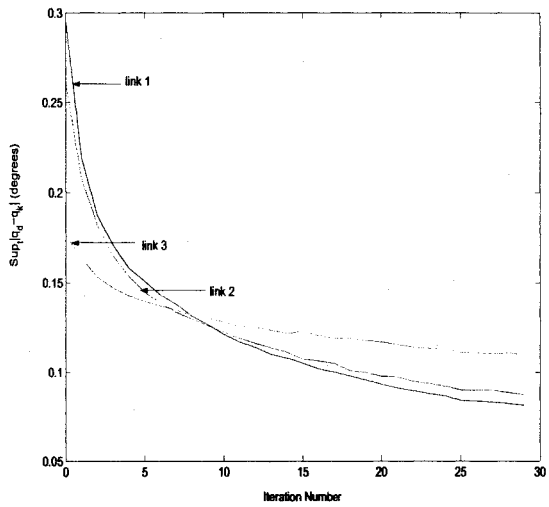

Figure 7.53: Sup-norm of the tracking error as a function of the iteration number for link 1, 2 and 3 with the adaptive learning scheme 3 with the 3rd experimental test

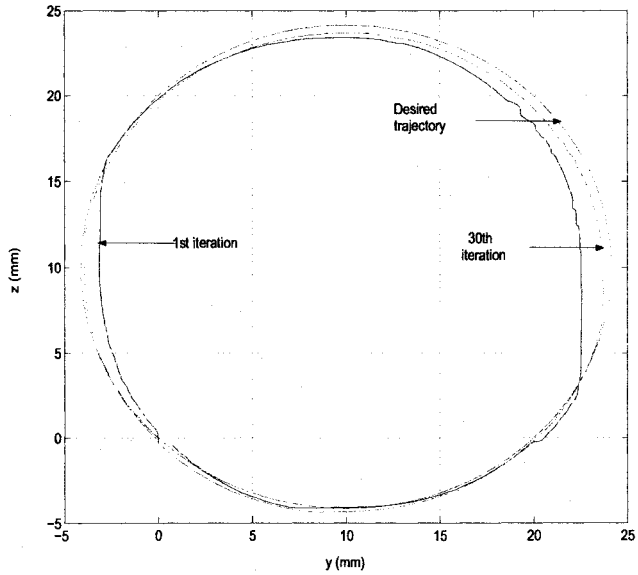

Figure 7.54: Desired trajectory and robot trajectory at the 1st and the 30th iteration with the adaptive scheme 3 with the 3rd experimental test

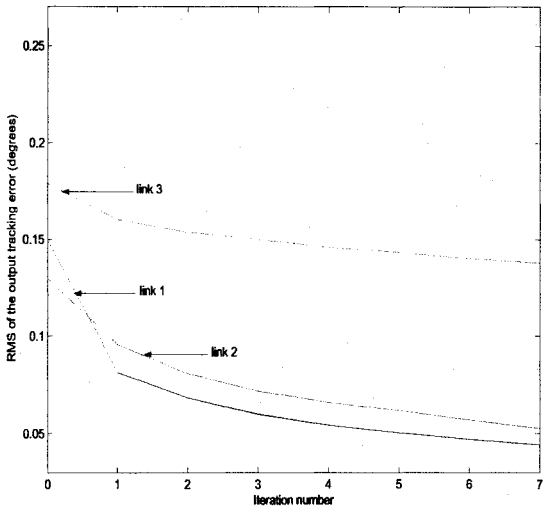

Figure 7.55: The RMS of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 3 with the 4th experiment

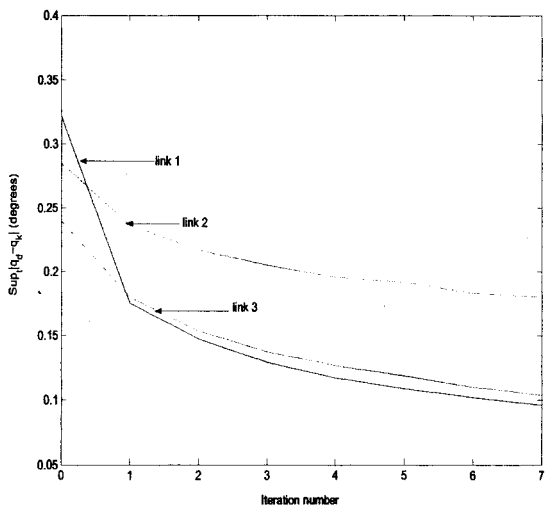

Figure 7.56: Sup-norm of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive learning scheme 3 with the 4th experiment

Observation

- The control approach requires only one iterative variable which is an interesting development for real-time applications of ILC technique in the sense that it requires less memory space and computational effort.
- The design does not require any *a priori* knowledge of the robot dynamics which is also a significant development in the ILC research.

Adaptive scheme 4

The presented AILC scheme 4 is tested experimentally on the CRS255 robot manipulator. In order to generate a smooth control input and avoid the chattering, we replaced the signum function in the adaptation law by a saturation function defined in equation 7.2.

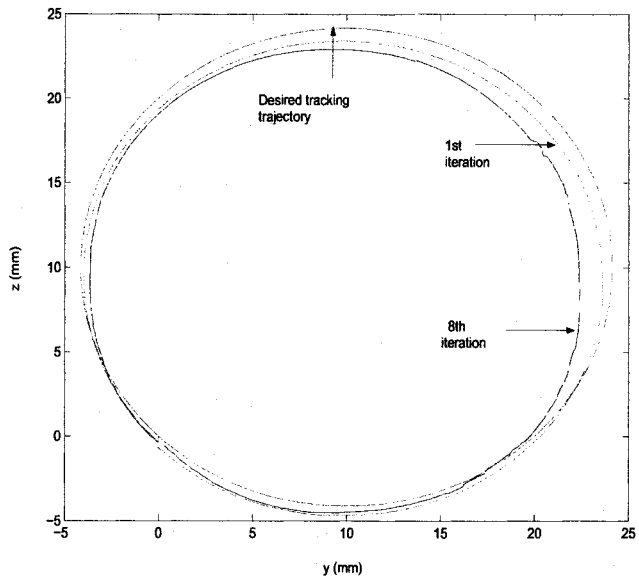

Figure 7.57: Desired trajectory and robot trajectory at the 1st and the 8th iteration with the adaptive learning scheme 3 with the 4th experiment

Experiment 1:

The adaptive scheme 4 is implemented on the 5-DOF CRS255 robot manipulator to track a desired circular trajectory which is shown in figure 7.25. The control and learning gains for this experimental evaluation are taken as follows:$\alpha = 1$, $\nu = diag(2, 2, 2)$ and $K_l = 8I_{2\times 2}$. The sampling period is taken as 0.001 s. The convergence of the output tracking error of the robot joint positions are investigated. The implementation results are shown in figures 7.58 to 7.59.

Experiment 2:

In this experimental evaluation, our main target is to track a circular trajectory in the (Y-Z) plane which can be expressed as follows: $y(t) = 25 + \sqrt{625}\cos 0.1t$ mm and $z(t) = 25 + \sqrt{625}\sin 0.1t$ mm with $t \in [0, 63]$. Using the AILC scheme 4, with $\alpha = 1$, $\nu = diag(1, 1, 2)$ and learning gain $K_l = 0.5I_{2\times 2}$, we obtained the experimental results shown in figure 7.60 and 7.61 for the 5-DOF CRS255 robot manipulator. In this test, the sampling period is taken as 0.001 s .

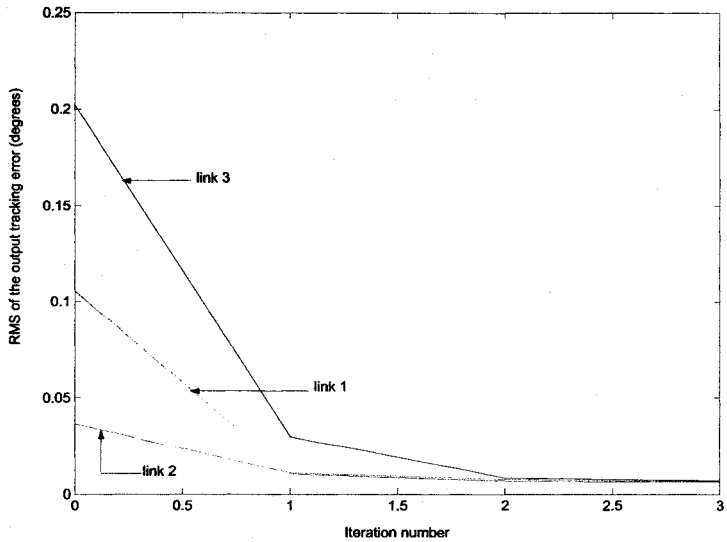

Figure 7.58: The RMS of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive scheme 4 with the 1st experiment

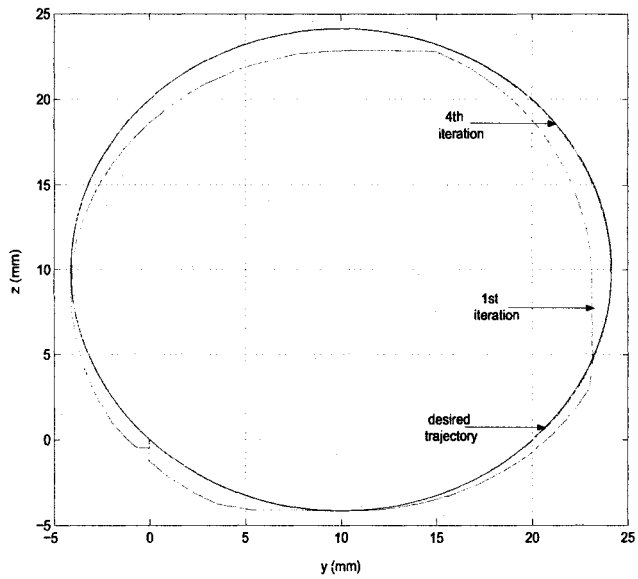

Figure 7.59: Desired trajectory and robot trajectory at the 1st and the 4th iteration under the adaptive scheme 4 with the 1st experiment

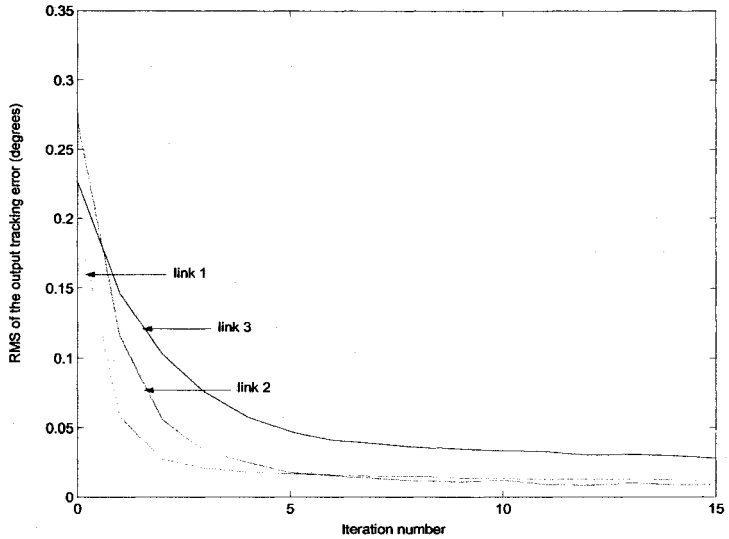

Figure 7.60: The RMS of the tracking error as a function of the iteration number for link 1, 2 and 3 under the adaptive scheme 4 with the 2nd experiment

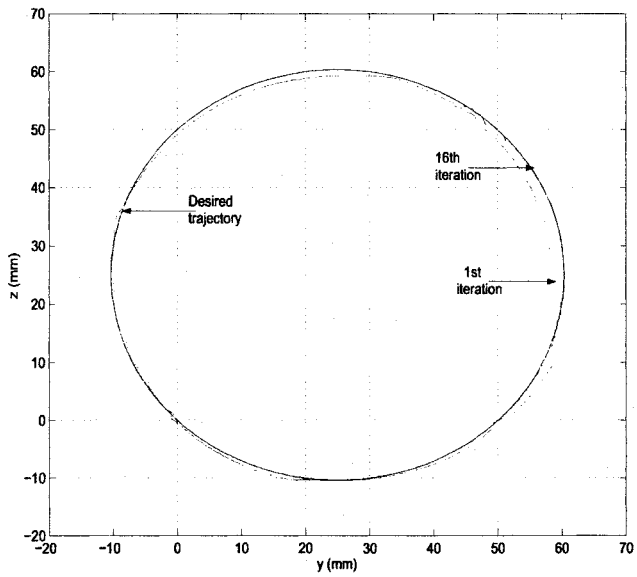

Figure 7.61: Desired trajectory and robot trajectory at the 1st and the 16th iteration under the adaptive scheme 4 with the 2nd experiment

Observation

- The design does not require *a priori* knowledge of the robot dynamics.
- It requires only two updating parameters which reduces memory components and computing power in real-time applications.
- The key feature of this AILC scheme is that the parametric adaptation law does not require velocity signals. Therefore, the noise accumulation through the iterative process is considerably reduced. As a result, the learning process can be continued until the robot trajectory converges to the desired one.
- Unfortunately, we don't have any proof of convergence for this algorithm. The proof of convergence of this learning process will be part of our future research work.

7.4 Concluding remarks on Adaptive ILC approach

From our several experimental tests, one can draw the following conclusions:

- The beauty of the implemented adaptive ILC algorithms (Tayebi 2004) that is they can be used in a straightforward manner for any industrial robot manipulator that is already working under a traditional PD controller by just adding the iterative term in order to enhance the tracking performance from operation to operation.
- In order to achieve faster convergence of the output tracking error, a high learning control gain can be used. In this case, lower cut-off frequency in the estimated velocity signals is necessary to reduce the noise effect in the adaptation laws that are based on the use of the velocity estimation.
- The "dirty derivative" signals in the adaptation laws is the main drawback of the presented AILC approach in the real-time implementation because the filtered derivative signals are contaminated by noise which amplifies from iteration to iteration. As a result, the learning process has to stop because of the chattering phenomenon that appears when the number of iteration is increased. As a

matter of fact, the amount of noise amplification in the real-time application can be attenuated by reducing the cut-off frequency. On the other hand, undesirable oscillations may occur in the robot joints causing the robot to become unstable at low cut-off frequencies.

Chapter 8

Conclusion and future work

8.1 Conclusion

In this thesis several classical and adaptive iterative learning control algorithms have been tested experimentally on a 5-DOF CRS255 (CATALYST5) robot manipulator. The carried out tests have shown that the adaptive scheme 3 and 4 produce the best results in the sense of the requirement of memory components, computing power, *a priori* knowledge of robot dynamics and the tracking error convergence. Based on the implementation results, one can draw the following conclusions:

- In contrast with the experimental results of classical ILC schemes, the tracking error convergence of the adaptive ILC approach shows monotonic convergence.
- The presented adaptive ILC approach is simple in the sense that it requires only a positive definiteness condition on the control and learning gains. Moreover, compared with other classical and adaptive ILC schemes, the implemented adaptive ILC schemes require less memory space and computing power. In this approach, the acceleration measurements and the bounds of the robot parameters are not needed.
- The AILC framework allows the designer to make the output tracking error and its time derivative arbitrarily small, at the first operation, over the finite time interval [0, T], by increasing the minimum eigenvalues of the control and learning gains.

- Today, many industrial robot manipulators are equipped with high precision sensors for the position measurements. Velocity sensors are frequently omitted due to saving cost, volume and weight. Therefore, the velocity signals in the AILC schemes 1, 2 and 3 are not measured but estimated from the joint encoder positions using a filtered derivative. As a result, in practice, the learning performance of these adaptive schemes is seriously affected at high cut-off frequency of the filtered derivative signals because of the amplification of the measurement noise.

- In practice, the signum function used in the adaptation law might cause the chattering phenomenon. In order to avoid this problem and to smooth out the control input, the signum function can be replaced by a saturation function which also leads to a poor learning performance.

- From our experimental evaluation of the AILC schemes it is seen that a faster convergence rate can be achieved by increasing the learning and control gains. However, chattering phenomenon appears when the number of iterations is increased. This is due to the fact that the parametric adaptation law uses the "dirty derivative" tracking error signals obtained from the previous iteration. As a matter of fact, this noise amplification problem forces us to stop the learning process after a certain umber of iterations.

8.2 Future Work

In order to avoid the noise amplification problem and improve the learning performance of the AILC schemes 1, 2 and 3, one has to design P-type adaptive ILC schemes that do not require velocity measurements. In fact, the adaptive ILC scheme 4 is an interesting step towards achieving that goal, since it provides quite interesting results, but, unfortunately, we don't have any proof of convergence for this algorithm yet.

Bibliography

Arimoto S. and F. Miyazaki, 1984. Stability and robustness of PID feedback control for robot manipulators of sensory capability. *In robotic Research: 1st International Symposium*, MIT press, Boston, MA, pp. 783-799.

Arimoto S., S. Kawamura and F. Miyazaki, 1984. Bettering operation of robots by learning. *Journal of Robot Systems*, vol 1, pp. 123-140.

Arimoto S., 1990. Learning control theory for robotic motion. *Internation Journal of Adaptive Control and Signal Processing*, vol 4, pp. 543-564.

Arimoto S., 1985. Mechanical theory of learning with application to robot control. *In Proc. of the 4th Yale Workshop on Application of Adaptive Systems Theory*, pp. 379-388.

Arimoto S., S. Kawamura, F. Miyazaki and S. Tamaki, 1984. Learning control theory for dynamical systems. *Proc. of 24th Conference on Decision and Control*, Ft. Lauderdale, Florida, pp. 1375-1380.

Arimoto S., S. Kawamura and F. Miyazaki, 1985. Can mechanical robots learn by themselves? *In proceedings of 2nd International Symposium of Robotics Research*, Kyoto, Japan, pp. 127-134

Arimoto S., 1996. *Control theory of non-linear mechanical systems: A passivity-based and circuit-theoretic approach* , Oxford university press, Oxford.

Arimoto S., P. T. A. Nguyen and T. Naniwa, 2000. Learning of robot tasks on the basis of passivity and impedance concepts. *Robotics and Autonomous Systems*, vol. 32, pp.79-87.

Atkeson C. G. and J. McIntyre, 1986. Robot Trajectory Learning Through Practice. *IEEE International Conference on Robotics and Automation*, pp. 1737-1742.

Bondi P., G. Casalino and L. Gambardella, 1988. On the iterative learning control theory for robotic manipulators. *IEEE Journal of Robotics and Automation*, vol. 4, no.1, pp. 14-22.

Bien Z. and J.-X Xu, 1998, Iterative Learning Control: Analysis, Design, Integration and Application. *Kluwer Academic Publishers.*

Casalino G. and G. Bartolini, 1984. A Learning Procedure for the Control of Movements of Robotic Manipulators. *IASTED Symposium on Robotics and Automation*, Amsterdam, pp. 108-111.

Chen Y., Wen, Z. Gong and M. Sun, 1999. An iterative learning controller with initial state learning. *IEEE Trans. on Automatic Control*, vol. 44, no. 2, pp. 371-376.

Choi J. Y. and J. S. Lee, 2000. Adaptive iterative learning control for uncertain robotic systems. *IEE-Proc., Control Theory Appl.*, vol. 147, no. 2, pp. 217-223.

Chow T. W. S. and Y. Fang, 1998. An iterative learning control method for continuous-time systems based on 2-D system theory. *IEEE Trans. on Circuits and Systems-1*, vol. 45, no.4, pp. 683-689.

Craig J. J., 1984. Adaptive Control of Manipulators Through Repeated Trials. *Proceedings of the American Control Conference*, San Diego, pp. 1566-1574.

De Luca A., G. Paesano and G. Ulivi, 1992. A frequency domain approach to learning control: Implementation for a robot manipulator. *IEEE Trans. On Industrial Electronics*, vol. 39, no. 1, pp. 1-10.

De Roover D., 1996. Synthesis of a robust iterative learning controller using an H_∞ approach. *Proceedings of 35th IEEE Conference on Decision and Control*, Kobe, Japan, pp. 3044-3049.

French M. and E. Rogers, 2000. Nonlinear iterative learning by an adaptive Lyapunov technique. *International Journal of Control*, vol. 73, no. 10, pp. 840-850.

Ge S. S., 1998. Advanced Control Techniques of Robotic Manipulators. *Proceedings of the American Control Conference*, Philadelphia, Pennsylvania.

Gu Y. L. and N. K. Loh, 1987. Learning control in robotic systems. *Proceedings of IEEE International Sypmposium on Intelligent Control*, Philadelphia, Pennsylvania, pp. 360-364.

Guglielmo K. and N. Sadegh, 1996. Theory and implementation of a repetitive robot controller with cartesian trajectory description. *Journal of Dynamic Systems, Measurement and Control*, vol 118, pp. 15-21.

Ham C., Z. H. Qu and J. H. Kaloust, 1995. Nonlinear learning control for a class of nonlinear systems based on Lyapunov's direct method. *Proceedings of the American Control Conference*, Seattle, USA, pp. 3024-3028.

Ham C., Z. Qu and R. Johnson, 2000. A nonlinear iterative learning control for robot manipulators in the presence of actuator dynamics. *International Journal of Robotics and Automation*, vol. 15, no. 3, pp. 119-130.

Ham C., Z. Qu and R. Johnson, 2000. A nonlinear iterative learning control for robot manipulators in the presence of actuator dynamics. *International Journal of Robotics and Automation*, vol. 15, no. 3, pp. 119-130.

Harris C. J., C. J. Hoore and M. Brown, 1993. *Intelligent Control: Aspects of Fuzzy Logic and Neural Nets*, World Scientific Publishing Co. Ltd., Singapore.

Heinzinger G., D. Fenwick , B. Paden and F. Miyazaki, 1992. Stability of learning control with disturbances and uncertain initial conditions. *IEEE Transactions on Automatic Control*, vol. 37, pp. 110-114.

Hillenbrand S. and M. Pandit, 2000. An iterative learning control with reduced sampling rate for plants with variations of initial states. *Internation Journal of Control*, vol. 73, no. 10, pp. 882-889.

Horowitz R., W. Messner and J. B. Moore, 1991 . Exponential convergence of a learning controller for robot manipulators. *IEEE Transactions on Automatic Control*, vol. 36, no.7, pp. 402-411.

Horowitz R., 1993. Learning control of robot manipulators. *ASME Journal of Dyn. Syst., Meas. and Control*, Vol. 115, pp. 402-411.

Horowitz R., 1994. Learning control applications to mechatronics. *JSME International Journal.* vol. 37, no.3, pp. 421-429.

Hsu Chun-Te, C. J. Chien. and C. Y. Yao, 2003. A New Algorithm of Adapative Iterative Learning Control for Uncertain Robotic Systems. *American Control Conference*, pp. 830-834.

Huang S. N., K. K. Tan and T. H. Lee, 2002. Necessary and sufficient condition for convergence of iterative learning algorithm. *Automatica*, vol. 38, pp. 1257-1260.

Ishihara T., K. Abe and H. Takeda, 1992. A discrete-time design of robust iterative learning controllers. *IEEE Transcations on Systems, Man and Cybernetics*, vol. 22, no. 1, pp. 74-78.

Jang H. S. and Longman R. W., 1994. A new learning control law with monotonic decay of the tracking error norm. *In proceedings of the 15-th Annual Allerton Conference on Communication, Control and Computing*, UIUC, USA, pp. 314-323.

Jang T. J., C. H. Cho and H.S. Ahn, 1995. Necessary and sufficient condition for convergence of iterative learning algorithm. *Automatica*, vol. 38, pp. 1257-1260.

Kawamura S., F. Miyazaki and S. Arimoto, 1984. Iterative learning control for robotic systems. *International conference on Industrial Electronics, Control and Instrumentation (IECON)*, pp. 393-398.

Kazumasa K. and R. Horowitz, 1997. Repetitive and Adaptive Control of Robot Manipulators with Velocity Estimation. *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 204-216.

Kavli T., 1992. Frequency domain synthesis of trajectory learning controller for robot manipulators. *Journal of Robotic Systems*, vol. 9, No. 5, pp. 663-680.

Khorrami F. and Ozguner, 1988. Decentralized Control of Robot manipulators via state and proportional-integral feedback. *IEEE International Conference on Robotics and Automation*, Philadelphia, PA, pp. 1198-1203.

Kawamura S., F. Miyazaki and S. Arimoto, 1988. Realization of robot motion based on a learning method. *IEEE Transactions on Systems, man and cybernetics*, vol. 18, N. 1, pp. 126-134.

Kuc T. Y. and J. S. Lee, 1991. An adaptive iterative learning control of uncertain robotic systems. *Procedings of IEEE Conference on Decision and Control*, Britain, pp. 1206-1211.

Kuc T. Y., K. Nam and J. S. Lee, 1991. An iterative learning control of robot manipulators. *IEEE Transactions on Robotics and Automation*, vol. 7, N. 6. pp. 835-841.

Kuc T. Y. and W. G. Han, 2000. An adaptive PID learning control of robotic manipulators. *Automatica*, vol. 36, pp. 717-725.

Lee H. S. and Z. Bien, 1997. A note on convergence property of iterative learning controller with respect to sup-norm. *Automatica*, vol. 33, no. 8. pp. 1591-1593.

Lewis F.L., C. T. Abdallah and D. M. Dawson, 1993. *Control of robot manipulators Macmillan*, New York.

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

Mita T. and E. Kato., 1985. Iterative control and its application to motion control of robot arm - a direct approach to servo-problems. *In Proc. of 24th Conference on Decision and Control*, Ft. Lauderdale, USA, pp. 1393-1398.

Mita T. and E. Kato., 1985. Iterative control of robot manipulators. *In Proc. of 15th International Symposium on Industrial Robots*, pp. 665-672.

Moore K. L., 1998. Iterative learning control : An Expository Overview, *In Applied and Computational Controls, Signal Processing, and Circuits*, vol. 1, no. 1, pp. pp. 151-214.

Moore K. L., M. Dahleh and S. P. Bhattacharyya, 1990. Some results on iterative learning control. *In procedings of 1990 IFAC World Congress*, Tallin, Estonia, USSR.

Moore K. L., and Jian-Xin Xu 2000. Special issue on iterative learning control. *International Journal of Control*, vol.73, no.10, pp. 819-823.

Moon J. H., T. Y. Doh and M. J. Chung, 1997. An iterative learning control scheme for robot manipulators. *In Proc. of International Conference on Intelligent Robots and Systems*, Grenoble, France.

Moon J. H., T. Y. Doh and M. J. Chung, 1998. A Robust Approach to Iterative Learning Control Design for Uncertain Systems. *Automatica*, vol. 34, no. 8, pp. 1001-1004.

Naniwa T., 1996. Trajectory control of a DD robot by learning, *Dessertation conducted by H. Muraoka at Yamaguchi University*, Japan.

Norrlof M., 2000. Iterative learning control: analysis, design and experiments. *Ph.D. dissertation*, Linkopings University, Linkoping, Sweden.

Norrlof M. and S. Gunnarsson, 2002. Some new results on current iteration tracking error ILC. *IEEE Transactions on Robotics and Automation* vol. 18, no. 4, pp. 636-641.

Norrlof M. and S. Gunnarsson, 2002. An adaptive approach to iterative learning control with experiments on an industrial robot. *IEEE Transactions on Robotics and Automation* vol. 18, no. 2, pp. 245-251.

Gunnarsson S. and M. Norrlof, 1997. On the use of learning control for improved performance in robot control systems. *In Proc. European Control Conference*, Brussels, Belgium.

Oh S. R., M. S. Lim, H. S. Ahn and K. B. Kim, 1994. Experimental study on iterative learning control algorithms for direct drive robot arm. *Proceedings of the Asian Control Conference*, Tokyo, pp. 553-556.

Ortega R. and M. W. Spong, 1988. Adaptive Motion Control of Rigid Robots: A Tutorial. *Proceedings of the 27th Conference on Decision and Control*, Austin, Texas.

Owen D. H., 1992. Iterative learning control convergence using high-gain feedback. *In Proceedings of 31th Conference on Decision and Control*, Tucson, AZ, USA, pp. 2545-2546.

Park B. H, Kuc T. Y. and J. S. Lee, 1996. Adaptive learning of uncertain robotic systems. *International Journal of Control* , vol. 65, no.5, pp. 725-744.

Park B. H, J. S. Lee and Kuc T. Y., 1998. Adaptive learning control of rbotic systems and its extension to a class of nonlinear systems. *Chapter 13 of Iterative Learning Control: Analysis, Design, Integration and Application*, Kluwer Academic Publishers, pp. 239-259..

Park K. H, Z. Bien. and D. H. Hwang, 1999. A study on the robustness of a PID-type iterative learning controller against initial state error. *International Journal of Systems Science*, vol. 10, no 1, pp. 239-259.

Park J. S. and Hesketh T., 1993. Model reference learning control for robot manipulators. *In Proceedings of the IFAC 12th Triennial World Congress*, Sydney, Australia, pp. 341-344.

Saab S. S., 1994. On the P-type learning control. *IEEE Transactions on Automatic Control*, vol. 39, pp. 2298-2302.

Sadegh N. and R. Horowitz, 1990. Stability and robustness analysis of a class of adaptive controller for robot manipulators. *International Journal of Robotics Research*, vol. 9, no. 3.

Seo W. G., B. H. Park and J. S. Lee, 1999. Intelligent learning control for a class of nonlinear dynamic systems. *IEEE Proceedings Control Theory Application*, vol. 146, no. 2, pp. 155-170.

Slotine J. J. and W. Li, 1991. *Applied nonlinear control*, Prentice Hall.

Slotine J. J. and W. Li, 1987. On the adaptive control of robot manipulators. *International Journal of Robotics Research*, vol. 6, pp. 49-59.

Spong M. W. and M. Vidyasagar, 1989. *Robot dynamics and control*, John Wiley and Sons, Inc.

Takegaki M. and S. Arimoto, 1981. A new feedback method for dynamic control of manipulators. *Journal of Dynamic Systems, Measurement and Control*, vol. 103, pp. 119-125.

Tayebi A. and M. B. Zaremba, 1999. Exponential convergence of an iterative learning controller for time varying nonlinear systems. *In Proceedings of 38th IEEE Conference on Decision and Control*, Phoenix, AZ, USA, pp. 1593-1598.

Tayebi A. and M. B. Zaremba, 2000. Internal Model-Based Robust Iterative Learning control for Uncertain LTI Systems. *In Proceedings of the 39th IEEE Conference on Decision and Control*, pp. 3439-3444.

Tayebi A. and M. B. Zaremba, 2001. Robust iterative learning control design for uncertain LTI systems via μ-synthesis, *Technical report.*

Tayebi A. and M. B. Zaremba, 2003. Robust Iterative Learning Control Design is Straightforward for Uncertain LTI Systems Satisfying the Robust Performance Condition. *IEEE Transactions on Automatic Control*, vol. 48, pp. 101-106.

Tayebi A., 2004. Adaptive iterative learning control for robot manipulators. *Automatica*, vol. 40, no. 7, pp. 1195-1203.

Tayebi A. and S. Islam, 2004. Adaptive iterative learning control for robot manipulators: Experimental results. *Under Review in Control Engineering Practice.*

Tayebi A. and S. Islam, 2004. Experimental evaluation of an adaptive iterative learning control scheme on a 5-DOF robot manipulator. *Accepted as an invited paper in IEEE Conference on Control Applications*, Taiwan, Sept. 2-4.

Togai M. and Yamano O., 1985. Analysis and design of an optimal learning control scheme for industrial robots: a discrete system approach. *In Proc. of the 24th*

IEEE Conference on Decision and Control, Ft. Lauderdale, Florida, pp. 1399-1404.

Tomei P., 1991. Adaptive PD controller for robot manipulators. *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 565-570.

Uchiyama M., 1978. Formulation of high speed motion pattern of a mechanical arm by trial. *Trans. SICE (Soc. Instrument Control Engineering)*, vol. 14, no. 6, pp. 706-712.

Wang D., 2000. On D-type and P-type ILC designs and anticipatory approach. *International journal of Control*, vol. 73, no 10, pp. 890-901.

Wang D. and C. C. Cheah, 1998. An iterative learning control scheme for impedance control of robotic manipulators. *International journal of Robotics Research*, vol. 19, pp. 1091-1104.

Wen J. and S. Murphy, 1990. PID control for robot manipulators. *Tech. Rep. CIRSSE*, no 54, Rensselaer Polytechnic Institute, Troy, NY.

Xu J-X., 2002. The frontiers of iterative learning control—Part II. To appear in *Journal of Systems, Control and Information*, Vol. 46, no. 5, pp. 233-243.

Xu J-X., V. Badrinath and Z. Qu, 2000. Robust learning control for robotic manipulators with an extension to a class of nonlinear systems. *International Journal of Control* , vol. 73, no. 10, pp. 858-870.

Xu J-X. and Y. Tan, 2001. A suboptimal learning control scheme for nonlinear systems with time-varying parametric uncertainties. *Journal of Optimal Control–Applications and theory*, vol. 22, pp. 111-126.

Ye Y. and D. Wang 2002. Multi-chanel design for ILC with Robot Experiments. *Proceedings of the seventh International Conference on Control, Automation, Robotics and Vision (ICARCV2002)*, Singapore.

Appendix A

MODELLING AND DYNAMICS

In this part of the thesis we give a short description and modeling of the CRS255 robot manipulator that is used in our experimental evaluation.

A.1 Industrial robot system

The robot used in this thesis for real-time implementation of ILC algorithms is a 5-DOF CRS255 (CataLyst5) industrial robot system shown in figure 6.1. This is a 5DOF open-chain articulated industrial robot arm. The articulated links are : the waist q_1, shoulder or upper-arm q_2, elbow or fore-arm q_3, wrist bend or pitch q_4 and wrist roll q_5. The motion equation relates only to the principal structures of the system that performs gross motions (major linkage). In this analysis, the wrist and end-effectors will be considered as a single inertia because wrist joints are usually dominated by inertia, with gravity and inertial coupling effects in comparison with the other links. So, the gross motion links considered for optimization are two main movable links in the upper-arm, forearm and the robot waist.

A.2 Mechanical modelling

The modeling of industrial robots can be divided into kinematic and dynamic modeling. The dynamic part will only be discussed in this section and the materials

partly based on Spong and Vidyasagar (1989). There are a number of procedures for generating the dynamic equations of motion for a robot manipulators:

- Euler-Lagrange(E-L) method
- Newton-Euler (N-E) method
- Recursive Lagrangian method
- Kane's method
- Appel's method
- Generalized D'Alambert principle method

Basically, the models are obtained from known Newtonian physical laws and these methods are equivalent to each other in the sense that they describe the dynamic nature of the same physical robot systems. However, the structure of these equations and the computational efficiency of the equations may differ because they are obtained for various reasons and purposes, such as adequate for simulation, controller design, real-time control etc. Among these technique, the E-L and the N-E formulation have been frequently used to generate the dynamical model of the robot manipulators. These methods have their own advantages and disadvantages. In this thesis, the E-L formulation will be the used for the dynamical model in the design of the ILC.

A.2.1 Dynamics

The Euler-Lagrange equations are a tool from analytical mechanics that can be used to derive the equations of motion for a mechanical system. In this approach the joint variables are considered as generalized coordinates. The kinetic energy of the manipulator can be defined as

$$K(q, \dot{q}) = \frac{1}{2}\dot{q}M(q)\dot{q} \tag{A.1}$$

where inertia matrix $M(q) \in R^{n\times n}$ is symmetric, bounded and positive definite. Let $P(q)$ be a continuously differentiable function, called the potential energy. For a rigid

robot, the potential energy is due to the gravity forces only. The potential energy $P(q)$ is independent of joint velocity vector $\dot{q}$. So, the Lagrangian function can be defined for such system as

$$L(q, \dot{q}) = K(q, \dot{q}) - P(q) \tag{A.2}$$

The Euler-Lagrange equations can be written as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \frac{\partial L}{\partial q_k} = \tau_k, k = 1, 2, 3....n \tag{A.3}$$

where, τ_1,τ_2, τ_3.....τ_n represent the input forces and torques to be applied at each joint. Using the potential and kinetic energy for the Lagrangian L, the above equation leads to

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \tag{A.4}$$

where $M(q)$ is a positive definite inertia matrix, $C(q, \dot{q})\dot{q}$ is a vector containing all inter-axial velocity-dependent coupling terms arising from centripetal and Coriolis forces, $g(q)$ represents gravitational forces term and τ is the vector of joint input containing the torques and forces to be applied at each joint. There are some important properties of the Lagrangian dynamics (A.4) that are helpful in the analysis and design of the manipulator control system. These properties are found in Spong and Vidyasagar (1989):

1. The inertia matrix $M(q)$ is a positive definite and symmetric, and there exist scalars such that

   $$\delta_1(q)I \leq M(q) \leq \delta_2(q)I \tag{A.5}$$

   for revolute joints δ_1 and δ_1 are constants.

2. The matrix $\dot{M}(q) - 2C(q, \dot{q})$ is skew symmetric.

3. The mapping $\tau \rightarrow \dot{q}$ is passive, i.e., there exists $\beta \geq 0$ such that

   $$\int_0^T \dot{q}^T(u)\tau(u)dt \geq -\beta \tag{A.6}$$

4 The equations of motion given in (A.4) are linear in the inertia parameters.

These properties have already been used in several stability and convergence of adaptive, robust and learning controllers analysis for robot manipulators.

A.3 Dynamical equation for the CRS-255 robot manipulator

In this section, we use the Euler-Lagrange approach to obtain the dynamical equations associated with the 5-DOF open-chain articulated robot manipulator shown in figure 6.1. For convenience, and without much loss of generality, the gross motion links are selected for the mathematical modelling of CRS255 robot that includes two main movable links in the upper-arm, forearm and the robot waist. In this case, the orientation links i.e., wrist and end-effectors will be considered as a single inertia. This is due to the fact that, in comparison with oither links, the wrist joints are usually dominated by inertia, gravity forces and coupling effects (Corke 1986). In fact, this assumption comes from the practical consideration because link 4 and 5 are light and move with low velocities. In order to utilize Lagrange's method for modelling, we must first find the kinetic energy of each of the links with the following notations

l_1 : length of the 1st link
l_2 : length of the 2nd link
l_3 : length of links 3, 4 and 5
m_1 : weight of the 1st link
m_2 : weight of the 2nd link
m_3 : weight of the links 3, 4 and 5
I_1 : moment of inertia of link 1
I_2 : moment of inertia of link 2
I_3 : moment of inertia of link 3 including link 4 and link 5

Link 1

Since the center of mass of the first link does not move, it's kinetic energy consists only the angular velocity part. The total kinetic energy of the moving rigid link 1 relative to the motion of it's mass

$$K_1 = \frac{1}{2}\omega_1^T I_1 \omega_1 = \frac{1}{2} I_1 \dot{q}_1^2$$

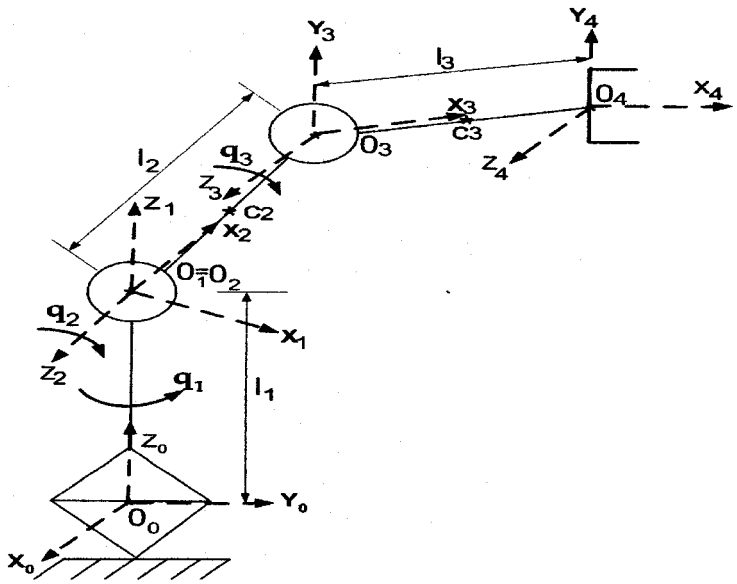

Figure A.1: Co-ordinate frames for CRS Robot Manipulator

Link 2

The rotational part of the kinetic energy of link 2, K_{2rot}, is

$$K_{2rot} = \frac{1}{2}\omega_2^T I_2 \omega_2 = \frac{1}{2}\left[I_2\dot{q}_1^2 \sin^2 q_2 + I_2\dot{q}_1^2 \cos^2 q_2 + I_2\dot{q}_2^2\right]$$

The translational part of kinetic energy for the link 2, K_{2trans}, is

$$K_{2trans} = \frac{1}{2}m_2 v_{c2}^T v_{c2} = \frac{l_2^2}{8}m_2\left[\dot{q}_1^2 \sin^2 q_2 + \dot{q}_2^2\right]$$

The total kinetic energy of the link 2, K_2, is

$$K_2 = \left[(\frac{l_2^2}{8}m_2 \sin^2 q_2 + \frac{1}{2}I_2)\dot{q}_1^2 + (\frac{1}{2}I_2 + \frac{l_2^2}{8}m_2)\dot{q}_2^2\right]$$

Link 3

The translational part of the kinetic energy for link 3 is

$$\begin{aligned} K_{3trans} &= \frac{1}{2} m_3 v_{c3}^T v_{c3} \\ &= \frac{1}{2} m_3 \left[\left(\frac{l_3^2}{4} \cos^2(q_2+q_3) + l_2 l_3 \cos q_2 \cos(q_2+q_3) + l_2^2 \cos^2 q_2 \right) \dot{q}_1^2 + \left(l_2^2 + \frac{l_3^2}{4} \right. \right. \\ &\quad \left. \left. + l_2 l_3 \cos q_3 \right) \dot{q}_2^2 + \frac{l_3^2}{4} \dot{q}_3^2 + \left(\frac{l_3^2}{2} + l_2 l_3 \cos q_3 \right) \dot{q}_2 \dot{q}_3 \right] \end{aligned}$$

The rotational part of kinetic energy is

$$K_{3rot} = \frac{1}{2} \omega_3^T I_3 \omega_3 = \frac{1}{2} \left[\dot{q}_1^2 + (\dot{q}_2 + \dot{q}_3)^2 \right] I_3$$

The total kinetic energy of the link 3, K_3, is

$$\begin{aligned} K_3 &= \frac{1}{2} m_3 \left[\left(\frac{l_3^2}{4} \cos^2(q_2+q_3) + l_2 l_3 \cos q_2 \cos(q_2+q_3) + l_2^2 \cos^2 q_2 \right) \dot{q}_1^2 + \left(l_2^2 + \frac{l_3^2}{4} + \right. \right. \\ &\quad \left. \left. l_2 l_3 \cos q_3 \right) \dot{q}_2^2 + \frac{l_3^2}{4} \dot{q}_3^2 + \left(\frac{l_3^2}{2} + l_2 l_3 \cos q_3 \right) \dot{q}_2 \dot{q}_3 \right] + \frac{1}{2} \left[\dot{q}_1^2 + (\dot{q}_2 + \dot{q}_3)^2 \right] I_3 \end{aligned}$$

The potential energy, P, of the system is given by

$$P = \left[\frac{l_2}{2} m_2 g \cos q_2 + l_2 m_3 g \cos q_2 + \frac{l_3}{2} m_3 g \cos(q_2+q_3) \right]$$

The Lagrangian function, L, is given by

$$\begin{aligned} L &= K_1 + K_2 + K_3 - P \\ &= \left[\frac{1}{2} I_1 + \frac{1}{2} I_2 + \frac{1}{2} I_3 + \frac{l_2}{8} m_2 \sin^2 q_2 + \frac{1}{8} m_3 l_3^2 \cos^2(q_2+q_3) + 0.5 m_3 l_2 l_3 \cos q_2 \cos(q_2+q_3) \right. \\ &\quad \left. + 0.5 m_3 l_2^2 \cos^2 q_1 \right] \dot{q}_1^2 + \left[0.5 I_2 + \frac{1}{8} l_2^2 m_2 + 0.5 m_3 l_2^2 + \frac{1}{8} l_3^2 m_3 + \frac{1}{2} m_3 l_2 l_3 \cos q_3 + \frac{1}{2} I_3 \right] \dot{q}_2^2 \\ &\quad + \left[\frac{1}{8} l_3 m_3 + \frac{1}{2} I_3 \right] \dot{q}_3^2 + \left[I_3 + \frac{1}{4} m_3 l_3^2 + \frac{1}{2} m_3 l_2 l_3 \cos q_3 \right] \dot{q}_2 \dot{q}_3 - \left[\frac{l_2}{2} m_2 g \cos q_2 + l_2 m_3 g \right. \\ &\quad \left. \cos q_2 + \frac{l_3}{2} m_3 g \cos(q_2+q_3) \right] \end{aligned}$$

The dynamical equations for the manipulator are given by,

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) - \frac{\partial L}{\partial q_1} &= \tau_1 \\
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_2}\right) - \frac{\partial L}{\partial q_2} &= \tau_2 \\
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_3}\right) - \frac{\partial L}{\partial q_3} &= \tau_3
\end{aligned} \tag{A.7}$$

Substituting for the various quantities in the above equations and after some mathematical manipulations, the differential equations that describes the dynamical behavior of this manipulator can be derived as

$$\begin{aligned}
&\Big[I_1 + I_3 + I_2 + \frac{1}{4}m_2 l_2 \sin^2 q_2 + \frac{1}{4}m_3 l_3^2 \cos^2(q_2+q_3) + m_3 l_2 l_3 \cos q_2 \cos(q_2+q_3) + m_3 l_2^2 \\
&\cos^2 q_2\Big]\ddot{q}_1 + \Big[\frac{1}{2}l_2 m_2 \sin q_2 \cos q_2 - \frac{1}{2}m_3 l_3^2 \cos(q_2+q_3)\sin(q_2+q_3) - m_3 l_2 l_3 \sin(2q_2+q_3) \\
&-2m_3 l_2^2 \cos q_2 \sin q_2\Big]\dot{q}_1\dot{q}_2 - \Big[\frac{1}{2}m_3 l_3^2 \cos(q_2+q_3)\sin(q_2+q_3) + m_3 l_2 l_3 \cos q_2 \\
&\sin(q_2+q_3)\Big]\dot{q}_1 q_3 = \tau_1
\end{aligned} \tag{A.8}$$

$$\begin{aligned}
&\Big[I_2 + I_3 + \frac{1}{4}l_2^2 m_2 + \frac{1}{4}m_3 l_3^2 + m_3 l_2^2 + m_3 l_2 l_3 \cos q_3\Big]\ddot{q}_2 + \Big[I_3 + \frac{1}{4}m_3 l_3^2 + \frac{1}{2}m_3 l_2 l_3 \cos q_3\Big]\ddot{q}_3 \\
&-m_3 l_2 l_3 \sin q_3 \dot{q}_2 \dot{q}_3 - \frac{1}{2}m_3 l_2 l_3 \sin q_3 \dot{q}_3^2 - \Big[\frac{1}{4}l_2 m_2 \sin q_2 \cos q_3 - \frac{1}{4}m_3 l_3^2 \cos(q_2+q_3) \\
&\sin(q_2+q_3) - \frac{1}{2}m_3 l_2 l_3 \sin(2q_2+q_3) - m_3 l_2^2 \cos q_2 \sin q_2 - \frac{1}{2}m_3 l_2 l_3 \cos q_2 \sin(q_2+q_3\Big]\dot{q}_1^2 \\
&- \Big[\frac{1}{2}m_2 l_2 g \sin q_2 + m_3 l_2 g \sin q_2 + \frac{1}{2}l_3 m_3 g \sin(q_2+q_3)\Big] = \tau_2
\end{aligned} \tag{A.9}$$

$$\begin{aligned}
&\Big[I_3 + \frac{1}{4}m_3 l_3\Big]\ddot{q}_3 + \Big[I_3 + \frac{1}{4}m_3 l_3^2 + \frac{1}{2}m_3 l_2 l_3 \cos q_3\Big]\ddot{q}_2 + \Big[\frac{1}{4}m_3 l_3^2 \cos(q_2+q_3)\sin(q_2+q_3) \\
&+0.5 m_3 l_2 l_3 \cos q_2 \sin(q_2+q_3)\Big]\ddot{q}_1^2 + \frac{1}{2}m_3 l_3 l_2 \sin q_3 \dot{q}_2^2 - \frac{1}{2}l_3 m_3 g \sin(q_2+q_3) = \tau_3
\end{aligned} \tag{A.10}$$

Appendix B

General procedures for Running an ILC Experiment

Suppose you have an ILC scheme that you want to implement on a 5DOF CRS255 robot system in open architectural mode. The following steps are essential to implement it properly:

a) First of all open **Robcomm3** by clicking the green icon from the task bar. To open a new terminal, click on the **Terminal** icon from the Robcomm3 menu bar.

b) Now go to **CRS-C500** controller box and turn on the power. After booting up the CRS-C500 controller, there will be a display **$$ pendant** in the Robcomm3 terminal. To release the teach pendant, just press **ESC** and then press **F1** on the teach pendant keypad. Now release the teach pendant red **Reset** button by rotating it in a clockwise direction.

c) Before applying the power to the robot, please make sure that the work envelope of the robot is clear. If it is, then apply power to the robot by pressing the **ARMPOWER** button on the CRS-C500 control box and promptly the Robcomm3 terminal window will display **$**.

d) Now type **Home** which will start the robot homing. When the robot is finished homing then Robcomm3 terminal display will show **$** again. Now type **ready** in the terminal window which will bring the robot into a ready position.

e) Develop the desired ILC algorithm in Matlab simulink with the switching model and save the simulink model file to this directory: **C : \designername\Quanser–CRS 255 OA**. The switching model is designed by the simulink blocks: stop, switch and clock block. The timer of the clock simulink block is chosen by the designer which will start the program calculations and data collections. To control program calculations and data collections, the output port of the switching model is placed through switching block to the desired data collection and control input ports. For real-time ILC implementations, this switching model is very important to start the program calculations and to store the data simultaneously.

f) Now open the Matlab command window and type **crs_startup** and then type the desired Simulink model **file name**. If a PD controller is used to develop an ILC algorithm in Simulink model then type **Gains** otherwise it is not necessary.

g) In order to build the desired simulink ILC model, select the **WinCon** menu bar from simulink model and click **Build**. If the model is successfully built then a new WinCon window will appear which means that the desired ILC model is ready to run in real-time.

h) To start an experiment, click the **start** button of the Wincon window. To take control of the robot manipulator, change the **Mode** bit from **1 to 0** in the simulink model. Now promptly go to Robocomm3 window, where Robocomm3 terminal is already open, and type **limp**. Then there will be a question of whether you want to limp all robot axes or not. Simply type **y** to run the experiment and collect the data for the upcoming operation. After finishing the 1st operation, go back to the Robcomm3 terminal window and type **nolimp**.

i) In the next operation, collected data is retrieved by typing **load filename** in the matlab command window. Then go back to the Robcomm3 terminal window and type **ready**, and follow the steps from **f** as explained.

j) To end the experiment, reapply the brakes of the robot arm by typing **nolimp** in the Robcomm3 terminal window and then click the stop button on the WinCon window. Now press **Reset** button of the teach pendant to turn off the robot power.

k) To turn off the CRS-C500 controller, just type **shutdown now** in the Robcomm3 terminal window. Now you can turn off the main power supply to the robot arm.