POSITION CONTROL OF LINEAR ULTRASONIC MOTOR

KRISHNA MAINALI
(B. E., KREC Surathkal, India)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

Acknowledgments

I would like to thank all the people who have helped me during my study at National University of Singapore. First and foremost, I would like to thank my supervisors Assoc. Prof. Sanjib Kumar Panda and Assoc. Prof. Jian-Xin Xu. They have been extremely enthusiastic and supportive regarding this research. I have learnt many aspects of electric drives and control through our frequent interactions. I appreciate Prof. Panda and Prof. Xu's innovative ideas and profound knowledge in electrical drives and control theory. Without their encouragement and support this study would have not been possible.

I had the pleasure of interacting with many research students from Electrical Machines and Drives Laboratory, and Power Electronics Laboratory. My sincere thanks to all of them. It was wonderful time working with them in the laboratory helping each other.

My warmest thanks to Electrical Machines and Drives Laboratory officers, Mr. Y. C. Woo and Mr. Mukaya Chandra. I appreciate their helpful nature and dedication in making laboratory such a nice place to work.

My study at National University of Singapore was made possible through graduate research scholarships. I am extremely thankful to National University of Singapore for the financial support.

And finally, there are no words suffice to express my heartfelt gratitude to my parents and brothers. I would have never reached so far without their constant love and support.

Contents

Summary

In recent years ultrasonic motors are finding wide applications in precision positioning. These motors have simple construction, large power to weight ratio, high torque/force at low speed, no effect and generation of electromagnetic field. Ultrasonic motors are used as actuators for robots, wafer metrology, photolithography, optical mounting, fiber alignment, fabrication and alignment of high density data storage devices.

This thesis is focussed on the position control of the linear ultrasonic motor HR8 manufactured by Nanomotiom Ltd. HR8 ultrasonic motor uses two orthogonal vibration modes of a piezoelectric plate to generate elliptical motion at the spacer, which is transferred to moveable stage through friction. Since non-linear phenomenon of inverse piezoelectric effect and friction are involved, it is very difficult to develop a detailed and accurate mathematical model of the motor. In the absence of an accurate motor model, intelligent control techniques based on neural networks and fuzzy logic have been reported in the literature. In spite of using sophisticated control algorithms with significant implementation overhead, the control performances achieved with these controllers are not enticing. In this research, first the open-loop characteristic of the motor is obtained. The motor characteristic is non-linear. It has a deadzone in its characteristic; the stage does not move till the control voltage exceeds a certain minimum value. This deadzone in motor characteristic is compensated by providing a feedfoward signal in motor control voltage depending on the velocity and position error profile. With the deadzone compensation, first a conventional linear proportional-integral (PI) controller is used for position control of the motor. A repeatable performance with a resolution of $100\,nm$ is achieved for set point position tracking. The performance of linear PI controller is not so satisfactory when tracking time varying reference trajectory and under load variations. In order to improve the tracking performance for time varying reference trajectories a linearized second order model of the motor is used to design a sliding mode position controller. The performance of the variable structure controller with sliding mode is found to be better than linear PI controller for tracking time varying position reference trajectory and under load variations.

The conventional sliding mode controller suffers from intense chattering, which leads to wear and tear in the motor. In sliding mode control, a high switching gain is desirable to ensure global attractiveness of the switching surface. Once in sliding mode, it is necessary to have equivalent control component to ensure good tracking performance. But due to presence of system modelling error and existence of disturbances, it is not possible to directly acquire the equivalent control. Thus, a low pass filter is used to extract equivalent control from the switching control signal and concurrently a second low pass filter is used to scale down the switching gain. These two filters are activated simultaneously and they work concurrently when the system enters the sliding phase. With this filtering scheme, chattering is significantly reduced. This scheme can also be used to estimate the disturbance in the system and provide compensation for it.

The detailed knowledge of the system is a prerequisite for designing a suitable control law to achieve high performance control. In the absence of the good system knowledge and when system is performing a repetitive task, the control performance of the system can be improved using an iterative learning process. The system knowledge available is made use of to design the controller and further system

knowledge is acquired when the system is running using iterative learning control. In iterative learning control (ILC), a feed-forward control signal is determined in an iterative manner to achieve the desired result by compensating the periodic errors. The error measured during the current cycle is converted into an improved feed-forward signal for the next cycle. During the next cycle the computed feed-forward signal is added to the control signal provided by the feedback controller to generate final control signal. This process is continued until the control signal is modified to the point such that when applied to the system produces the desired output. Using iterative learning control for repetitive position tracking of the ultrasonic motor, a significant improvement in position tracking performance is observed.

The iterative learning control is effective in improving the tracking performance only if the trajectory is strictly repetitive. However, for non-repetitive position tracking iterative learning control cannot be used. For such tasks a direct learning control technique has been proposed and implemented. A system may have plenty of prior control knowledge obtained through all past actions corresponding to different but highly correlated tasks. The knowledge of such prior control profiles is utilized for directly generating the control signal for a new but similar task. Such direct learning control technique is used for spatially identical but different time scales reference position trajectory tracking for the ultrasonic motor. This control technique is effective to limit the tracking error to a lower level. In case, if the new reference trajectory is repetitive, iterative learning controller is used in coordination with direct learning controller. Such a hybrid scheme limits the tracking error to a lower value and reduce the learning convergence time.

The efficacy of the position control schemes are substantiated by extensive experimental validations. The PI controller has good set point position tracking with repeatable resolution of $100\,nm$. Its performance deteriorates for tracking time varying trajectory and under load variations. Robust sliding mode control is

used to overcome this drawback of linear PI controller. Experimental results show that sliding mode controller gives satisfactory performance for set point and time varying position tracking even under load disturbances. For time varying reference tracking, the sliding mode controller improves the tracking performance by a factor of two as compared to linear PI controller. The chattering in sliding mode control is reduced by addition of two low pass filters. One low pass filter estimates equivalent control signal from switching signal and other reduces the switching gain concurrently once the operating point is in sliding mode. For repetitive position tracking, iterative learning control is introduced to further improve the performance of sliding mode control. With the addition of iterative learning control, the tracking error is approximately reduced by a factor of ten. The application of iterative learning controller is limited to strictly repeatable tasks. For non-repetitive position tracking direct learning control technique is used. Using direct and iterative learning control schemes in coordination, a good position tracking performance is achieved for both repetitive and non-repetitive position tracking.

List of Figures

List of Symbols

x_d	reference position (mm)
x	actual motor position (mm)
e	position tracking error (mm)
v_{ref}	reference velocity (mm/s)
U_{pi}	control voltage from PI position controller (V)
U_{ff}	feedforward control voltage (V)
K_P	proportional gain of PI controller (V/mm)
K_I	integral gain of PI controller (V/(mm.s))
V_c	net control voltage (V)
T_s	sampling time (s)
σ	sliding surface (mm/s)
λ	a positive constant (s^{-1})
u_{eq}	equivalent control (V)
u_c	nominal equivalent control (V)
u_s	switching control (V)
β	switching gain (V)
d	disturbance (V)
ϕ	thickness of boundary layer
M	mass of the motor stage (Kg)
B_v	coefficient of viscous friction (N.s/m)
F	force generated by piezoelectric vibration (N)

F_d	force due to disturbance (N)
F_{fric}	force due to friction (N)
K_f	force constant (N/V)
R	set of real numbers
β_0	upper bound of disturbance (V)
β_d	upper bound of disturbance derivative (V/s)
t_r	reaching time (s)
T_P	reference time period (s)
L	learning gain (V/mm)
δ	forgetting factor
$u_k^{ilc}(r)$	ILC control output at r^{th} instant of $(k)^{th}$ iteration (V)
y_d	desired position (mm)
u_d	control voltage to track desired position (V)
G_p	plant transfer function
G_{fb}	feedback controller transfer function
G_c	closed-loop transfer function
U_{smc}	sliding mode control voltage (V)
U_{dlc}	direct learning control voltage (V)

Chapter 1

Introduction

In this chapter a brief review on different types of electromagnetic actuators used for positioning applications are given. Ultrasonic motors are presented as an alternative actuators to electromagnetic motors for precision positioning. Survey on the present state of art in ultrasonic motors, motivation for the work done and structure of this thesis are summarized.

1.1 Background

Electric drives are the efficient medium of electromechanical energy converter to generate the mechanical power required for numeral applications ranging from rudimentary motion control to high precision machines tools. Many industrial applications like precision machining and processing of semiconductor, optoelectronic elements, and high density magnetic memory devices have increasing demand for high precision position control. These applications require actuators to be precisely controlled. Electromagnetic motors and piezoelectric actuators are main candidates for such applications.

1.1.1 Conventional Electromagnetic Motors

All electromagnetic motors (dc and ac) depend on the basic principle that a conductor carrying current in a magnetic field will have a force induced onto it. This force induced on the current carrying conductor is the source of motion in electromagnetic motors. Most often for industrial positioning applications rotary electromagnetic motors are used. The leadscrews are used to obtain the linear motion from the motor's rotary motion. Commonly used electromagnetic motors for positioning applications are DC servomotors, stepper motors and brushless DC/AC motors. A brief discussion on different types of electromagnetic motors used for servo applications is given in this section.

The DC motors have been popular choice for servo applications in the 1960s due to simplicity of their power converter requirement and ease of their control. The DC motor operates on the principle of a rotating armature and a stationary field system. The field flux is produced by the permanent magnet or electromagnet placed on the stator. Current is supplied to the armature via carbon brushes that are placed along the magnetic axis of the conductor. The magnetomotive force produced by armature excitation and the field are in quadrature to each other, and this generates the motor torque that causes the armature to rotate. The DC motor has linear torque/speed characteristics. The motor can be easily controlled electronically adjusting the applied voltage. With the help of shaft mounted rotary encoders, good speed and position control can be achieved. DC motors are becoming unpopular due to the operational problems associated with brushgear. The brushes need to be replaced periodically, depending on the load, speed and duty cycle. The commutation limits the top speed that can be reached before arcing over the commutator segments occurs. The space required for the commutator and brushgear is considerable. The heat generated in the rotor winding has to travel across the air gap and through the stator to be dissipated. This long thermal path results in a motor that is less thermally efficient and therefore a DC motor is larger than a brushless motor with an equivalent power rating. The windings also add inertia to the rotor, it results in lower peak acceleration than a similar brushless motor.

Stepper motors are inherently brushless and digital in operation. Stepper motor moves in discrete steps and hold the position at rest without the need for feedback device. In stepper motors, electromagnets in the stator are energized to cause teeth in the rotor to line up with teeth in the stator. Properly sequencing the current in the windings precise positioning can be achieved. Stepper motor generates full torque at low speeds. These motors can provide good positioning performance even in open-loop with excellent repeatability. The stepper motor rotation is in fixed increments and maximum speed depends upon the time constant of the control winding. The ultimate resolution of stepper motor system is determined by the minimum motor step angle. In stepper motor, the fine tooth structure requires a small air gap which adds to the manufacturing cost. For better performance, the motor requires expensive laminated steel.

Brushless motors are constructed with the windings on the stator and permanent magnets attached on the rotor. Generally, the brushless motors are of two types: a) Brushless DC permanent magnet motor and b) Brushless permanent magnet synchronous motor. Brushless DC permanent magnet motor are inside out machines *i. e.* the armature winding is on the stator whereas the permanent magnet is on the rotor. The brushless DC permanent magnet motor operates without brushes; the armature current commutation that the brushes provided are done electronically. Electronic commutation is provided by switching power switches (*e. g.* transistors) on and off at appropriate times based on rotor permanent magnet position information from a feedback device sensing the position of the permanent magnet on the rotor. The armature of three coil brushless DC permanent magnet motor is similar to the stator of a three phase AC machine. The rotor position information is used to energize the stator winding. At a given instant only two of the phases carry current and as the rotor rotates, different pairs of phase windings carry current. These motors require less maintenance due to absence of brushes and they have higher speed and acceleration capabilities. The brushless DC permanent magnet motors are easier to control and the control algorithm can be readily implemented in commercial micro-controllers [1]. Brushless permanent magnet synchronous motor also has the armature as a distributed AC winding on the stator and permanent magnet on the rotor. The magnets can be mounted on the rotor surface or they can be internal to the rotor. The interior construction simplifies assembly and relieves the problem of retaining the magnets against centrifugal force. The mechanical commutator is replaced by electronic commutator on the stator. The motors can be run directly from the AC supply without any electronic commutator too. There are basically two types of brushless permanent magnet synchronous motors: a) sinusoidally excited permanent magnet synchronous motor and b) trapezoidally excited permanent magnet synchronous motor. The sinusoidally excited permanent magnet synchronous motor are more popular brushless permanent magnet motors. Linear brushless motors are also available. Using the linear brushless motors, the need for rotary to linear motion conversion is eliminated. This eliminates the backlash, lead error and other mechanical system inaccuracies. All the electromagnetic force is utilized to produce the thrust directly. This type of motor comprises a number of base-mounted permanent magnets forming the stator and a translator (as counterpart of the rotor in a rotating motor) formed by a number of iron-core coils. By applying a three-phase current to three adjoining coils of the translator, a sequence of attracting and repelling forces between the poles and the permanent magnets will be generated. This results in a thrust force being experienced by the translator. The drawback of the brushless motors are higher cost compared to DC and stepper motors [2].

Another important linear motor used for positioning is voice-coil motor. The voice-coil motor's working mechanism is similar to that of an audio speaker. The motion of the coil carrying current in a magnetic field is used for different positioning applications where high-speed positioning is demanded [3]. Voice-coil stages are well suited for applications ranging from lens grinding to mirror positioning in laser applications where rapid response and rapid settling are critical. The principal drawback of voice-coil motors is that they generate considerable heat.

Induction motor has been the workhorse of the industry for servo applications. Induction machines are sturdy and are very suitable for harsh environments. The induction motor is indeed a brushless motor. The distributed windings in the stator when supplied with voltage, produce the rotating magnetic field. The change in flux linking with rotor conductors induces emf and forces the current through the short circuited rotor. These induce current in turn produce magnetic field which tries to catch the stator field by producing the torque in rotor. But the rotor field can never catch stator field. Slip is essential for the torque production and it is impossible, even in theory, to achieve zero rotor losses. This is one of the chief limitations of the induction motor, since rotor losses are more difficult to remove than stator losses. The efficiency and power factor of induction motors falls off in small sizes. Moreover, for induction motors the controllers are complex and expensive and cannot be justified in very small drives [1].

Another brushless electromagnetic motor having simple, rugged construction, high speed capability and low cost is switch reluctance motor. Switch reluctance motors are doubly salient, singly excited electric motors with windingless or permanent-magnetless rotors. Their concentrated stator coils are turned on and off sequentially through DC voltage pulses, which results in unipolar controlled current that produces the motor torque. In these motors, the knowledge of the rotor position is essential for phase current commutation. Switch reluctance motors have inherent torque ripples. Presence of significantly large torque ripples hinders their use in high performance servo applications.

Though widely used in the past century, the conventional electromagnetic motors in general have drawbacks as listed below

- Low power per unit weight,especially for small electromagnetic motors, heavy weight due to winding
- Difficult to miniaturize due to magnetic saturation
- Low efficiency at low speed
- Cause electromagnetic interference and susceptible to external electromagnetic fields

Due to these drawbacks of electromagnetic motors, for precision positioning applications ultrasonic motors are used as alternative source of actuation.

1.1.2 Ultrasonic Motors

In the 1980s, the semiconductor, photonics, biomedical industries began requesting for much more precise and sophisticated positioners which do not generate magnetic field noise. This high demand surged the developments in ultrasonic motors. Due to reasonable costs of piezoelectric ceramic, ultrasonic motors are already in use in consumer products like Canon camera lens control [4] and industrial applications like wafer testing, optical fiber alignment etc. [5].

An ultrasonic motor is a type of actuator that uses mechanical vibrations in ultrasonic range as its drive source [6]. An AC field is applied to piezoelectric ceramic element to generate the alternating expansions and contractions at or near mechanical resonant frequency. The oscillations of piezoelectric ceramic are mechanically rectified in the motor to obtain the unidirectional movement. The rotor/slider pressed against the vibrating stator receives motive power through friction. Fig. 1.1 shows the basic construction of ultrasonic motors. It consists of a high frequency power supply, a vibrator and a slider. The vibrator is composed of piezoelectric driving component and an elastic vibrator part. The slider is the moving part with a friction coat.

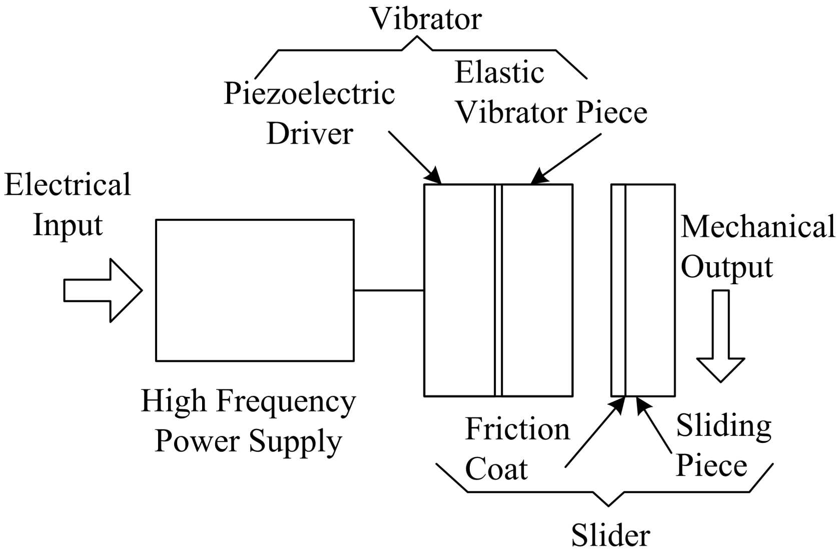


Figure 1.1: Fundamental construction of ultrasonic motors.

The first practical ultrasonic motor was proposed by H.V. Barth of IBM in 1973 [7]. In last 30 years tremendous research and development in this field has been done in Japan. Canon has utilized ultrasonic motors for automatic lens focusing systems and the company is developing smaller inexpensive ultrasonic motor for automatic film winding. Seiko Instruments has used miniaturized motors for their watches. Toyota has used ultrasonic motors for head-rest control in cars. The floppy drive, CD/laser disk drive etc. are possible future applications where ultrasonic motors can be used [8]. The tremendous rise in applications of ultrasonic motors comes from its advantages over conventional electromagnetic motors, as briefly stated below [9].

- Silent drive
- High power/weight ratio and high efficiency
- Compact size and light weight

- High troque/force at low speed, holding torque at zero speed
- Precise positioning is possible as error due to backlash caused by speed reduction gears are eliminated
- Simple construction and easy production process
- No effect and generation of electromagnetic field
- Flexibility in designing the shape of motor to suit particular applications

Despite these attractive features, ultrasonic motors have few drawbacks listed as below:

- High frequency power supply is needed.
- Less durable due to friction drive; special wear-resistant material for stator-rotor interface is needed in order to reduce friction losses and increase motor life.
- Properties of the piezoelectric ceramic elements may change with temperature causing the resonant frequency to change. This may lead to lower motor efficiency.

Ultrasonic motors are used in applications where conventional electromagnetic motors are inadequate. Some of these special applications are discussed here.

Actuators for robots: Ring or cylinder shaped motors could be used in the joints of the robots. This will result in much lighter robots than using electromagnetic motors.

Actuators for consumer goods: Consumer goods like camera where ring type of motor is used for autofocus lenses. Application of ultrasonic motors to consumer goods where frequent operation is not needed justifies its use.

Actuators for precise positioning: Since ultrasonic motors are direct drive without backlash, they are used as rapid positioning devices with positioning accuracies of nanometers in production of semiconductors [10].

Actuators for miniaturized machines: Electromagnetic motors smaller than 1 cm^3 are difficult to produce due to magnetic saturation. Ultrasonic motors efficiency is insensitive to size and are superior in mini/micro motor area. Research are still being conducted for biomedical applications.

Actuators for space machines: Ultrasonic motors can be used for space actuators where low speed operation in absence of lubrication is required.

There are many types of ultrasonic motors developed to date. These motors can be categorized according to their operating principle, construction or motion generated. According to the operating principle, they can be categorized either as standing-wave type or travelling-wave type. As per the construction, they can be either π-shaped, rod, ring and cylinderical types. The motors can also be classified as rotary, linear and multi-DOF ultrasonic motors. Since piezoelectric material can be easily machined into different shapes, so each type of motor has advantages for specific applications.

In this thesis , we will be concentrating on the Linear Standing-wave Ultrasonic Motor (LUSM) manufactured by Nanomotion Ltd.

1.2 Literature Survey

The history of the ultrasonic motors is much shorter than that of the electromagnetic motors, nevertheless many types of the motors have been developed and applied for several applications. Many publications on design and control aspects of these actuators have appeared during the last three decades. A brief summary of the works reported so far is presented in this section.

The most of the works are focussed on the design and control of ring type of ultrasonic motors. The construction and operating principle of the ring type ultrasonic motor invented by Sashida is described in detail in [6]. The servo control of ring type of ultrasonic motor has become a major topic in this area in last decade. Efforts have been made to develop different control algorithm for better control of the motor. Finding an accurate model of the ultrasonic motor suitable for control purpose is extremely difficult as motion is generated by piezoelectric effect and force is transferred through friction. These phenomenon are mathematically difficult to model. A simple model of the motor is found in [6], where it is claimed that the linear equivalent circuit model matches the actual response very well making it useful for designing a control system. A complicated averaged model has been derived and servo control design based on derived model is reported in [11]. A hybrid model combining the equivalent circuit model and analytical model has been proposed in [12]. Author claims that the model captures the nonlinear behavior of the motor and therefore is better model for control community than the existing models. Variable-structure adaptive control have been implemented successfully for position control in [13]. Many other servo control schemes that do not require detailed models such as fuzzy logic [14], neural network [15] and adaptive control [16] have also been proposed. For ring type ultrasonic motors, the three parameters of the two phase voltages can be used for servo control. The amplitude, phase difference and the frequency of voltages can be adjusted. Dual mode control technique with phase and frequency control has been used for precise position control in [17]. For quick and precise position control of rotary type of ultrasonic motors, the phase difference and frequency control methods are proposed in [18], [19]. In [18], the phase difference control system based on adaptive control, and driving frequency control system that takes friction force control to avoid the starting failure caused by hysteresis is discussed. The phase difference and frequency of the applied voltages are controlled based on sliding mode control in [19]. The ultrasonic motor has speed ripples caused by its driving principle and structure. These speed ripples are larger than those of other small motors and the suppression of speed ripples is an important problem for speed/position control of ultrasonic motor. Repetitive control scheme is proposed in [20] to reduce the speed ripple. The speed ripples with PI controller is reduced by addition of the repetitive compensator. Similar repetitive controller for speed ripple reduction for cylindrical ultrasonic motor is reported in [21]. The paper [22] proposes a position control scheme for ultrasonic motor employing backstepping control with fuzzy inference. The fuzzy inference is used to eliminate the dead-zone from motor characteristics. The adaptive backstepping control performs accurate positioning of the drive system. The paper [23] proposes a position control scheme for ultrasonic motor with adaptive dead-zone compensation. Neural network is adopted to determine a dead-zone compensation and sliding mode control is used for position control. C. Y. Yen *et.al.* [24] have designed the driving circuit and controller to deal with non-linearities ultrasonic motor. The comparative study on the performance of proportional-integral-derivative (PID) and sliding mode control is given. It is shown that sliding mode control has better positioning performance.

Though the linear ultrasonic motors work in same inverse piezoelectric principle as ring type of motor, there are less works reported in the design and control of them. A survey on the existing linear ultrasonic motors is found in [25]. Five different types of linear ultrasonic motors are described in the paper. Valentinas Snitka [26] has described theory of the linear ultrasonic motor using two vibration modes to generate the motion. The motor drive model based on system identification and prediction is developed and used for control. It has been shown that the developed model is accurate only for a small range of input signal. Different models are used for different input signal ranges. Kümmel *et. al.* [27] have detailed the working principle of the linear ultrasonic motor using superposition of longitudinal and bending vibrations. The oscillator is modelled taking the non-linear material behavior into consideration. Important aspects of the linear ultrasonic motor are discussed. In [28] a non-linear PID controller has been used for the position control of the linear ultrasonic motor. The non-linear PID controller is composed of two tracking differentiators which can yield high quality differential signal in the presence of disturbance and measurement noise. It is also shown that the tracking error for the non-linear PID controller is 50% less than that from the linear PID controller. Further the performance of the non-linear PID controller is enhanced by augmenting a feed-forward learning controller. Faa-Jeng *et al.* in [29] have used recurrent fuzzy neural networks with varied learning rates for periodic reference position tracking of linear ultrasonic motor. The recurrent neural network has been used to take benefit of the internal feedback present to enhance the dynamic performance and to learn the process. Further the fuzzy concept in the control is added as there are uncertainties in motor information. R. J. Wai *et al.* [30] have proposed the recurrent-fuzzy neural network control based upon the backstepping design technique for the position control of linear ceramic motor. The concept of the backstepping is to select recursively an appropriate function of state variables as pseudo-control inputs for the lower dimension subsystems of the overall system. The reference [31] presents a robust cerebellar-model-articulation-computer (CMAC) neural network control system for linear piezoelectric motor driven by a two-inductance two-capacitance resonant inverter. In [32] the application of the wavelet neural network (WNN) for position control has been proposed. The WNN has been designed using adaptive sliding mode control. The WNN is used to learn the ideal equivalent control law and a robust controller is designed to meet the sliding condition. In [33] a robust control system based on hypothetical dynamic model is proposed. The robust controller has three parts: state feedback controller, feedforward controller and uncertainty controller. The adaptation laws are used to control the gains of the state feedback controller. Author claims it to be more simple than intelligent control and having the learning ability. Effectiveness of the proposed controller is shown by comparing the tracking performance with that of integral-proportional (IP) controller.

1.3 Motivation

The ultrasonic motors have short history. Due to attractive features and potential applications many research works have been done. Most works have been focussed in designing new type of piezo-actuators. Comparatively less works have been focussed on control of such actuators for different applications. Though the ultrasonic motors have high potential for applications in high precision motion control systems, the inherent non-linear features associated with the dynamics of these actuators are challenges to how efficiently these potentials can be realized. Friction has been identified as one of the main problems to be addressed in ultrasonic motor. In ultrasonic motor, the friction is the primary medium of force transfer from the piezoelectric vibrator to the rotor/slider. At the same time, it also introduces nonlinearities in the motor characteristics. For high precision position control the frictional force needs to be adequately compensated.

Most controllers reported so far for the position control of linear ultrasonic motors are complex advanced non-linear controllers. These controllers are feared to be less favorable for practical applications as high costs are associated with their implementation. It is also difficult for the operators, often unfamiliar with advanced control algorithms, to adjust the control parameters. Although simple control structure is desired, achieving stringent precise positioning with conventional linear feedback controller is challenging. The main objective of this thesis is to address this problem of precision positioning with simpler control structure so that it can be easily realizable for practical applications.

The thesis has focussed on eliminating the deadzone present in motor characteristics based on a simple algorithm. With the elimination of hard non-linearity, *i. e.* deadzone, performance of conventional linear PI position controller is evaluated. It shows satisfactory response for set point position tracking. Its performance deteriorates when tracking time varying trajectory and under load disturbances. Thus a robust sliding mode controller is recommended to over drawback of linear PI position controller. The performance of sliding mode position controller is enhanced by extracting equivalent control from the switching signal using a low pass filter and reducing the switching gain concurrently by another low pass filter. The filters are simple first order low pass filters and can be easily implemented. For periodic position tracking, the performance of the sliding mode controller is improved by adding a plug-in iterative learning controller. The iterative learning controller is simple to implement and it reduces the periodic tracking error significantly. For non-periodic position tracking direct learning control is proposed and implemented. This controller is also easy to implement and use. It involves learning few control trajectories and use the mapping function to generate the control signal for similar but new trajectory. Finally, the position control techniques *i. e.* sliding mode control with iterative and direct learning control algorithms are integrated to provide a hybrid controller which provides improved tracking performance for various kinds of reference trajectory tracking.

1.4 Structure of the thesis

The thesis is organized into six chapters. In Chapter 2 the structure, working principle of the HR8 linear ultrasonic motor are presented. The drive system is explained. The motor open-loop characteristics is discussed.

Chapter 3 presents the closed-loop position control of the motor. First the performance of linear proportional-integral (PI) control is investigated. To have a better tracking performance when tracking a time varying trajectory and under load variations sliding mode control is proposed.

In sliding mode control, a high switching gain is desirable to ensure global attractiveness of the switching surface. For good tracking performance, it is necessary to have equivalent control component. Due to presence of system modelling error and existence of disturbance, it is not possible to directly acquire the equivalent control. Thus in Chapter 4, a sliding mode control with close-loop filtering is proposed. A low pass filter is used to extract equivalent control from the switching control signal and at the same time a second low pass filter is used to scale down the switching gain. The scheme has reduced chattering. This scheme can faithfully estimate the disturbance in the system and provide compensation for it.

In Chapter 5, for repetitive position tracking the repeatable error is reduced by using iterative learning control. The principle behind the iterative learning control is discussed. This plug-in controller is used to improve the tracking performance of the sliding mode controller.

The iterative learning control scheme proposed in the Chapter 5 is limited to strictly repeatable position tracking tasks only. In Chapter 6, a direct control estimation termed as direct learning control is proposed for tracking of non-repeatable position trajectories. Its usefulness in position tracking is shown by the experimental results.

Finally, Chapter 7 concludes this thesis highlighting the major contributions of this research. A brief possible future research directive is also included in it.

The appendices include the detail information of the linear ultrasonic motor, the servo driver, architecture of the controller used and a C program that has implemented sliding mode controller with iterative learning and direct learning control schemes.

Chapter 2

Linear Ultrasonic Motor

In this chapter, an introduction to the piezoelectric phenomenon is given. The structure and operating principle of the linear ultrasonic motor are explained. A detailed description of the experiment setup is presented. The open-loop characteristics of the motor is discussed.

2.1 The Piezoelectric Concept

The history of piezoelectric materials dates back to 1880, when Pierre and Jacques Curie published the first experimental demonstration of piezoelectricity in various materials such as rochelle salt, quartz, and tourmaline [34]. When these crystalline materials are subjected to tensile or compressive forces, they become electrically polarized. Conversely, the crystals deform when exposed to an electric field. Together, these two effects are known as piezoelectric effect. These two aspects are distinguished as *positive* and *inverse* piezoelectric effects.

The crystals like quartz and tourmaline generate minute force and thus such piezoelectric actuators were not as powerful as conventional actuators. In the 20th century, metal oxide-based piezoelectric ceramics and other manmade materials enabled designers to employ the piezoelectric effect and the inverse piezoelectric effect in many new applications. Generally these materials are physically strong and chemically inert, and they are relatively inexpensive to manufacture. Ceramics manufactured from a compound of lead zirconate and lead titanate exhibit greater sensitivity and higher operating temperatures, relative to ceramics of other compositions. These ceramic materials are named as "PZT" and are currently the most widely used piezoelectric ceramics [35], [36]. To make PZT piezoelectric, it has to

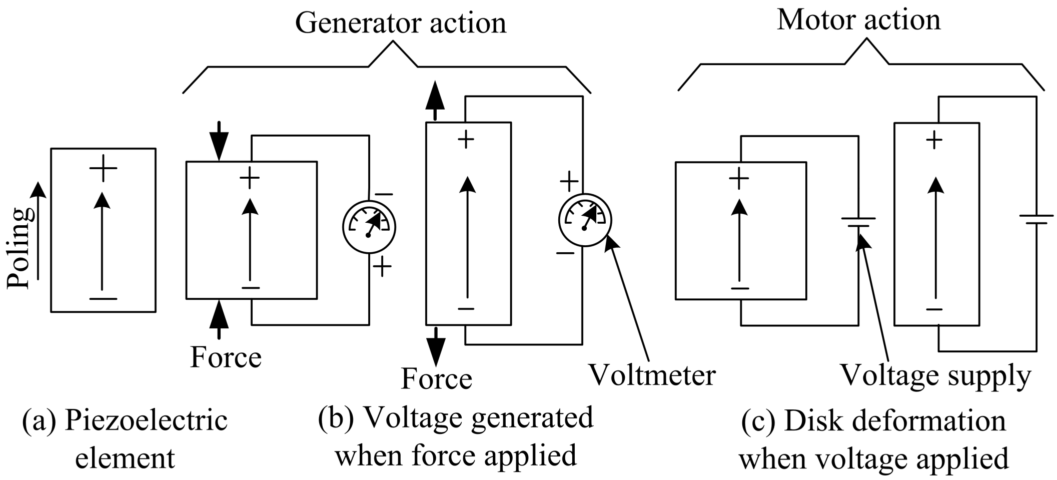

Generator action
Motor action
Poling
+
−
Force
Force
Voltmeter
Voltage supply
(a) Piezoelectric element
(b) Voltage generated when force applied
(c) Disk deformation when voltage applied

Figure 2.1: Piezoelectric Phenomenon.

be polarized. This involves exposing the material to very strong electric field. Under the action of such strong electric field, the electric dipoles become oriented in the direction of the field. A remanent polarization and deformation remains in the material when the field is removed. If an external compressive or tensile force is applied to the polarized material, the resulting change in dipole moment causes a voltage to appear between the electrodes on two opposite faces of the material. If the piezoelectric material is compressed, a voltage of the same polarity as the poling voltage will appear between the electrodes. If it is stretched, the voltage across the electrodes will be of opposite polarity to the poling voltage. This is generator action of piezoelectric ceramics, conversion of mechanical energy to electrical energy and it is shown in Fig. 2.1 (b). There exists the transverse effect also, if a voltage of opposite polarity to the poling voltage in applied to the electrodes, the piezoelectric material will contract. If the applied voltage has the same polarity as the poling voltage, the material will elongate. This is motor action of piezoelectric ceramics, conversion of electrical energy to mechanical energy as shown in Fig. 2.1 (c). If an alternating voltage is applied to the electrodes, the piezoelectric material will elongate and compress at the same frequency as that of the applied voltage. This is the principle used in ultrasonic motors. The contraction and elongation of the piezoelectric material is used for the generation of motion. Since the change in shape is very small, one may not consider that piezoelectric materials useful for larger displacement in motion control. However, by using piezoelectric elements to precisely generate a micro ellipse and create a continuous motion along the length of travel, ultrasonic motor can give larger displacement.

2.2 Linear Ultrasonic Motor

In this section, the structure of the linear ultrasonic motor HR8 from Nanomotion Ltd., which is used for the experiment, is described. The working principle of the motor is illustrated. The detail information of the motor is given in Appendix A.

2.2.1 Structure of the motor

The construction of the linear standing wave motor HR8 from Nanomotion Ltd. is as shown in Fig. 2.2. It consists of a thin cuboid shaped piezoelectric ceramic with two electrodes bounded on the large front face. Each electrode consists of two parts (A and A', B and B') placed diagonally to form a checkerboard pattern. The third electrode C is deposited at the bottom face of the ceramic plate fully covering it. The electrode C is grounded through a tuning inductor. The movement of the ceramic plate is constrained by a pair of fixed support and two high stiffness springs supporting along the long edge where the movement in the x-direction is zero. These supports are designed so as to allow the ceramic plate to slide in the y-direction. This arrangement results in less energy loss in the supports. A small finger like hard ceramic spacer is attached at the center of the short edge.

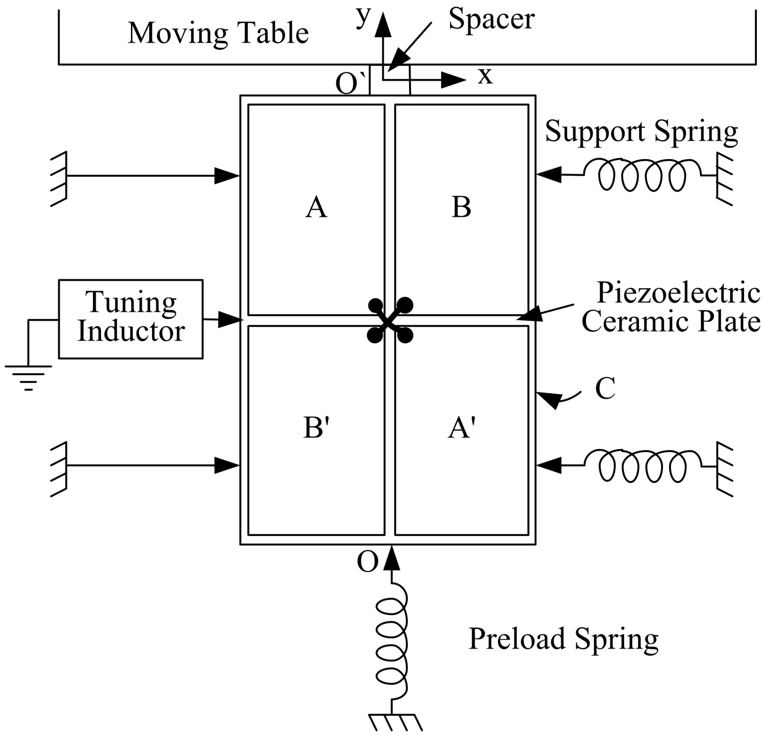
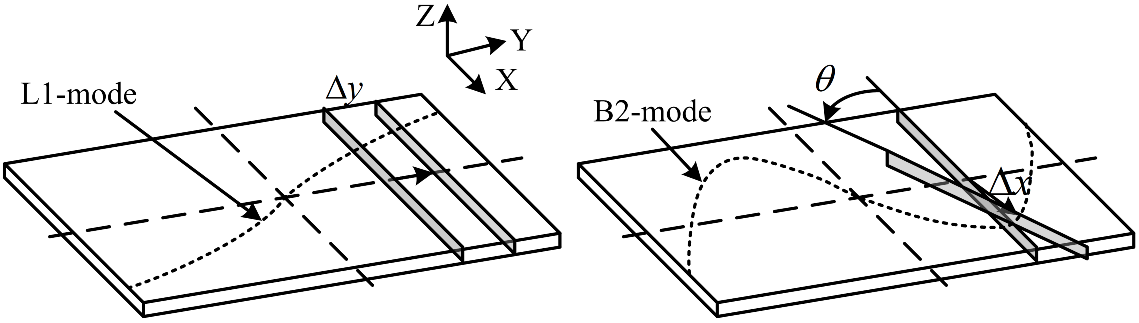


Figure 2.2: Linear Ultrasonic Motor Structure

A pre-load force provided by a spring from opposite short edge keep the spacer always in tact with the moving stage. The motion of the spacer is transmitted to the moving stage through dry friction. The HR8 motor consists of eight such piezoelectric ceramics configured in a 2×4 unit tandem/parallel configuration. All the piezoelectric ceramics are driven by a common control system. Using eight synchronously working piezoelectric ceramics, large driving force and velocity is achieved [37]. The moving stage is mounted on the motor base with V-flat way. The friction in this V-flat way is minimized using roller bearings.

2.2.2 Operating Principle of Linear Ultrasonic Motor

This linear ultrasonic motor uses two orthogonal vibration modes simultaneously to generate the driving force. It uses first longitudinal mode ($L1$-mode) in the y-direction and the second bending mode ($B2$-mode) in the x-direction as shown in Fig. 2.3 [27]. These modes of vibration results in elliptical motion of surface points in the x-y plane. The Fig. 2.3 (a) depicts the deformation of the piezoelectric ceramic plate for longitudinal mode. A cross section area in x-z plane is displaced in the y-direction by Δy. The longitudinal mode has vibrating node in the middle of the plate. Fig. 2.3 (b) shows the deformation of the ceramic plate for bending mode. A cross section area in x-z plane is displaced in x-direction and rotated about z-axis. The displacement in x-direction is denoted as Δx and angle of rotation as θ. This vibrating mode has nodes in middle, up and down parts of the ceramic plate. When the frequency of the exciting AC voltage is equal to the natural resonant frequency of the piezoelectric ceramic, the amplitude of the displacement will be maximum. The piezoelectric ceramic behaves like capacitor, thus the inductor between the electrode C and ground is used to tune the electrical resonance frequency of the ceramic plate to its mechanical resonance frequency. Elliptical motion is obtained only if the resonance frequencies of the two modes are equal or close to each other. Thus the length and width of the piezoelectric ceramic plate are chosen such that the resonance frequencies are as close as possible.



(a) Longitudinal vibration mode

(b) Bending vibration mode

Figure 2.3: Vibration Modes in the Piezoelectric Plate

The dimensions of the piezoelectric ceramic are such that the resonance frequencies ω_x and ω_y are closely spaced and have overlapping resonance curves as shown in Fig. 2.4 (a). The frequencies at the maximum displacements of $\bar{x}$ and $\bar{y}$ are denoted as resonance frequencies ω_x and ω_y respectively. The region between

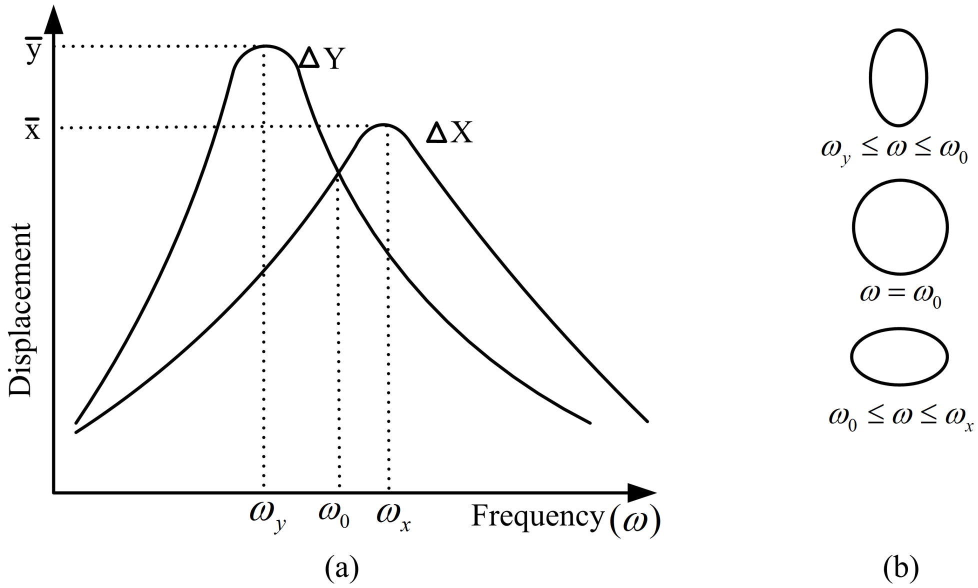


Figure 2.4: (a) Resonance curves of Piezoelectric Plate (b) Spacer motion for different excitation frequencies

ω_x and ω_y is defined as the *working zone*. The frequency of the exciting AC voltage is maintained in the working zone. As shown in Fig. 2.4 (b), when the frequency of the exciting voltage is equal to ω_0, the trajectory of the spacer will be an exact circle. If the excitation frequency is more than ω_0, the spacer trajectory will be a flat ellipse. This leads to larger stage movement step size. In cases, when the voltage frequency is less than ω_0, the spacer trajectory will be an acute ellipse giving smaller step size. When the motor is excited with AC voltage with frequency within the working zone, both Δx and Δy resonance will be excited. Fig. 2.5 (a) shows resonance of Δx and Δy when the electrodes A and A' are excited. The Δx and Δy at the spacer are positive causing the stage to move in the rightward direction. When the electrodes B and B' are excited the Δx is reversed in phase and Δy is positive as shown in Fig. 2.5 (b) causing the stage to move in leftward direction [37]. Thus the motor can be moved in forward and backward in x-direction depending on the pair of electrodes excited. The applied voltage on the motor determines the oscillation amplitude, which in turn determines speed of the motor.

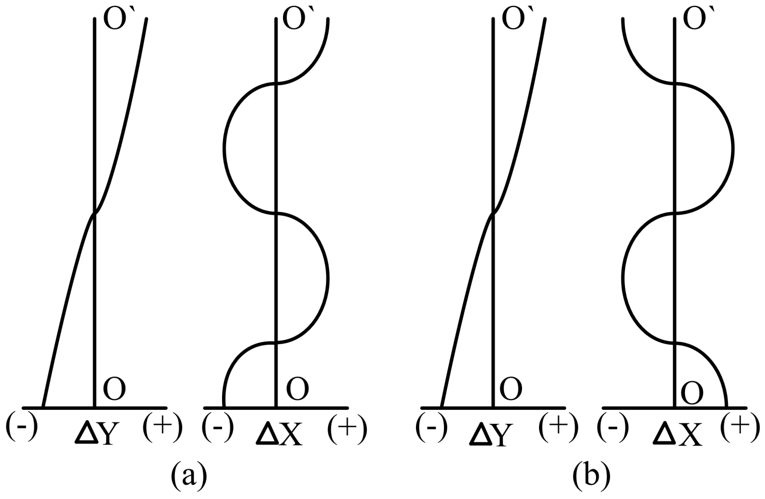


Figure 2.5: Displacement curves for (a) forward motion (b) backward motion

2.3 Experiment Setup

Experimental setup for the position control of the linear ultrasonic motor has been built using dSPACE1104 control card. Fig. 2.6 shows the configuration of the experimental setup. The setup comprises :

1. A PC for programming, debugging and real-time control.

2. A DS1104 controller board for executing control programs, performing D/A conversions, generating control signals, reading incremental encoder channel and communicating with the PC.

3. A surface-mounted HR8 ultrasonic motor with built-in incremental encoder for position sensing.

4. AB1A motor driver, manufactured by Nanomotion Ltd, converts the control voltage to high frequency AC voltage to drive the motor.

2.3.1 System Hardware

The drive system hardware essentially comprises the HR8 linear ultrasonic motor, optical incremental position encoder, AB1A motor driver, rectifier, control PC and

DS1104 controller board. The picture of the setup is shown in Fig. 2.7.

The PC used in the experiment is Pentium III running at 866 MHz. The DS1104 controller board is inserted into the PCI slot of the PC. The dSPACE DS1104 board is specially designed for research and development having features for simulation and real time implementation of digital controllers in various fields. The DS1104 board consists of PowerPC 603e microprocessor as main processor and TMS320F240 as slave DSP. The main processor runs at 250 MHz and controls ADC unit, DAC unit and 20-bit I/O, incremental encoder and serial interface. The slave DSP runs at 20 MHz and controls 14-bit digital I/O unit, PWM and serial peripheral interface. The detailed architecture of DS1104 board is given in Appendix C. The control voltage generated by the motor control algorithm is converted to analog voltage ($\leq \pm 10V$) by D/A converter. This control voltage is fed to the motor driver.

The structure and working principle of the HR8 linear ultrasonic motor has been given in previous Section 2.2. The parameters of the motor and the physical structure of the motor are given in Appendix A. The moving stage has the stroke length of 58 mm. The stage moves on V-guideways provided on stationary base plate of the motor platform. The friction in guideways is minimized by using ball bearings. An optical incremental encoder M2110, manufactured by MicroE systems, is used to sense the position of the motor stage. The stationary base of the HR8 ultrasonic motor is mounted with linear incremental encoder read head. The linear optical scale is mounted on the lower side of the moving stage. The encoder read head can read the gratings on the scale through a small aperture on the stationary base. The encoder is interfaced to the DS1104 incremental encoder channel through the buffer SN74HCT540N. The DS1104 incremental encoder interface internally performs a four fold subdivision of each encoder line giving the ultimate encoder resolution of 0.1 μm. The mounting of the position encoder on motor structure is shown in Appendix A, Fig. A.3.

The control voltage from the DS1104 is converted to high voltage sinewave required for piezoelectric vibration by the motor driver AB1A, manufactured by Nanomotion Ltd. The AB1A servo amplifier had three connection terminals on its front panel: 1) control terminal 2) I/O port 3) motor terminal. The control terminal is a five pin connector. This terminal has connection for +48 VDC power supply, analog control voltage from DS1104 and motor ENABLE signal. A female D-type 25 pin I/O port connector is used to interface control sources like joystick. The motor out terminal is a male D-type 9 pin connector, which interfaces the motor. AB1A generates a constant frequency sine wave 39.6 kHz with the magnitude dependent on the control voltage it receives. Depending upon polarity of control voltage, one of the two pairs of electrodes will be excited. For positive control voltage, one of the two pairs of electrodes will be excited so as to move the

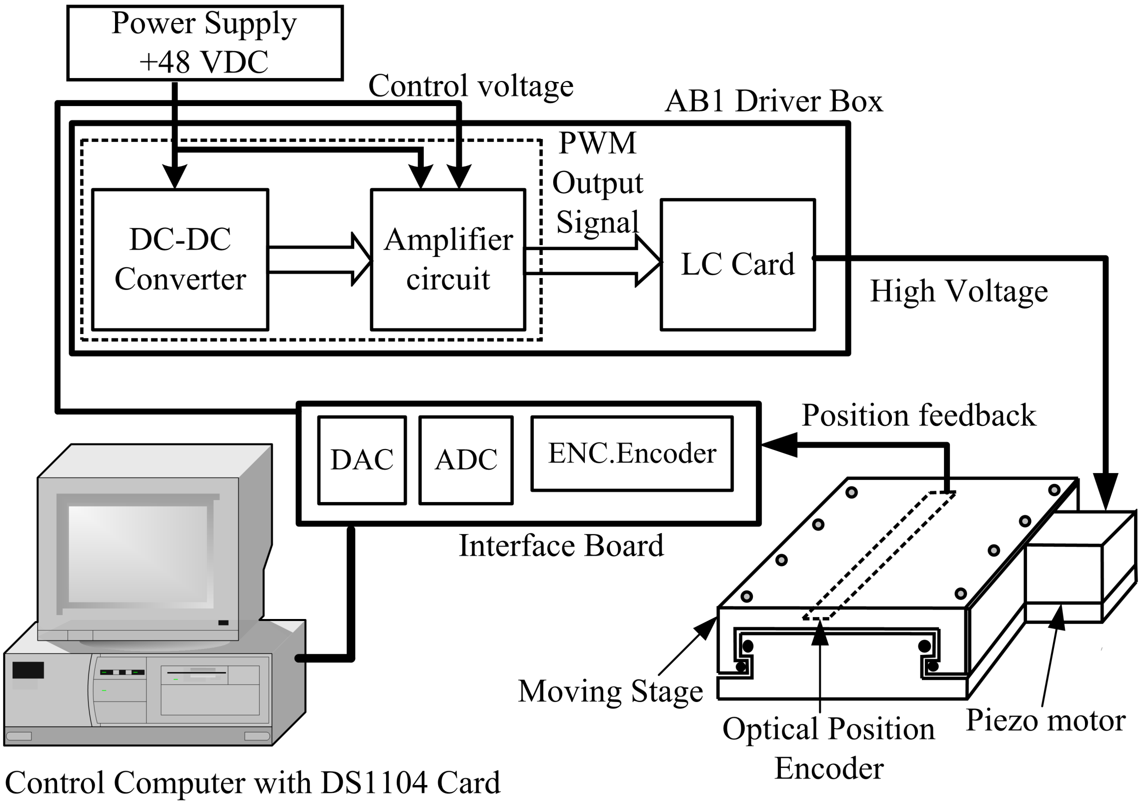


Figure 2.6: The Linear Ultrasonic Motor Drive System

motor in positive direction while for the negative control voltage the other pair of electrodes will be excited. The selection of the pair of electrodes to be excited is done internally in AB1A by activating either UP or DOWN lines in the motor out terminal. The generated sine voltage is fed to the motor via motor terminal. A rectifier is used to supply the DC power to the AB1A motor driver. The rectifier is supplied with $220\,V$, $50\,Hz$ and provides $+48\,VDC$ [38]. An insight into the internal circuit schematic of AB1A motor driver is given in Appendix B.

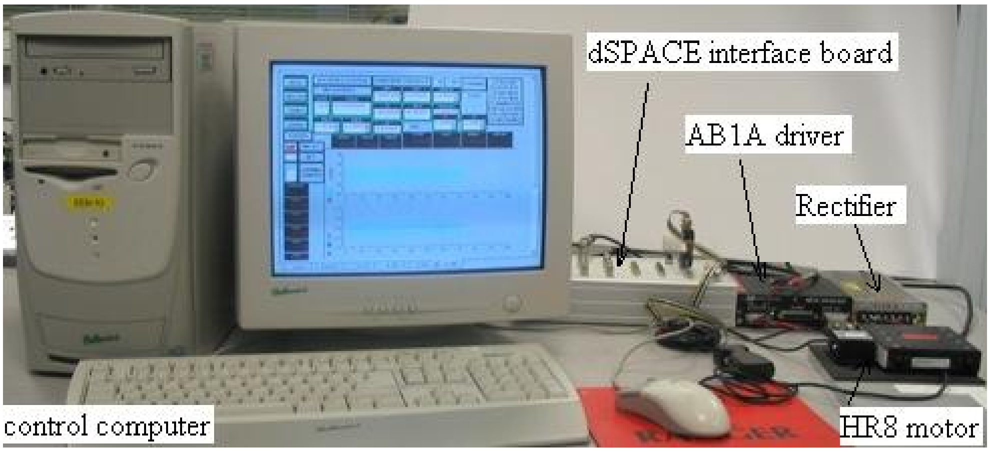


Figure 2.7: Photograph of Experiment Setup

2.3.2 Software Implementation

The entire control program in this project has been written in C language. dSPACE provides the flexibility to implement the control algorithm either in Simulink using their Real-Time Interface (RTI) library or directly in C using RTlib's functions. Programming in C offers greater flexibility and it is easier to implement many mathematical functions. Further the user will get the freedom to optimize the source code so as to achieve the lowest execution time. Fig. 2.8 shows the steps involved in building up the real-time executable code. When the C coded control program is downloaded to the processor, first the Microtec PowerPC C compiler converts the C source code to assembly language source codes by compiling, assembling and linking C source code modules. The assembler then translates the

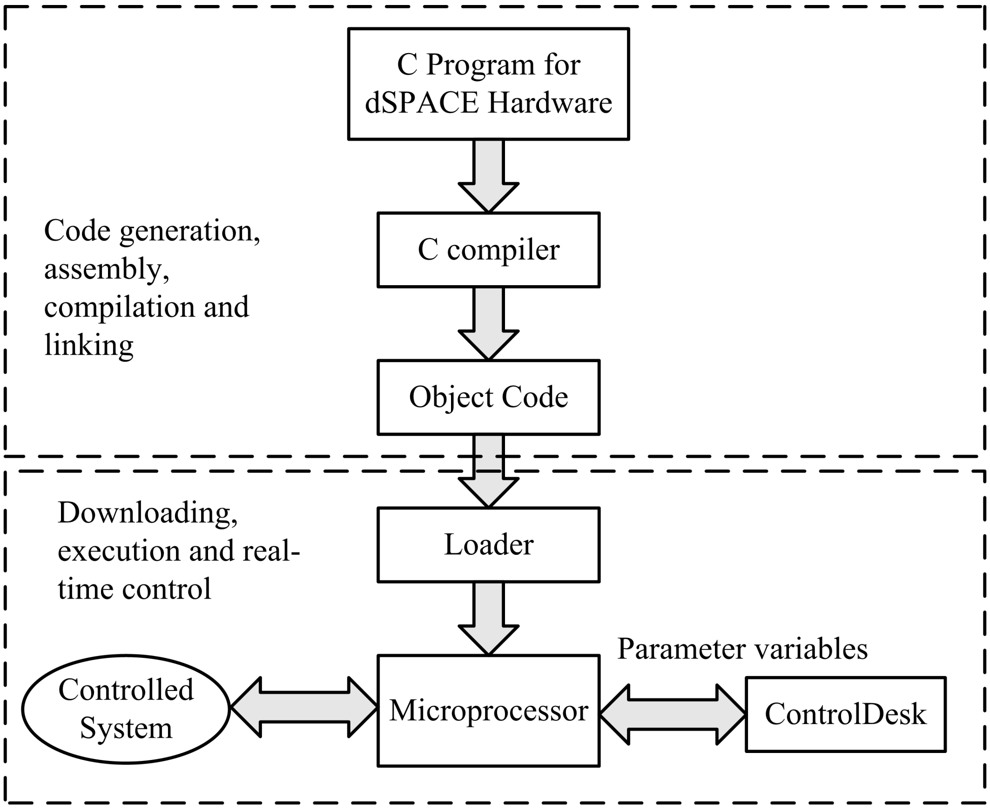

C Program for dSPACE Hardware
Code generation, assembly, compilation and linking
C compiler
Object Code
Downloading, execution and real-time control
Loader
Parameter variables
Controlled System
Microprocessor
ControlDesk

Figure 2.8: Real-Time Executable Code Generation

assembly code to machine language object files. The object files are loaded into the processor and the program execution begins. As soon as the real-time application is running, we can use ControlDesk to change and observe parameters of running real-time application.

Fig. 2.9 shows the flowchart of the main control program. All the control programs are implemented within the interrupt service routine (ISRT). The program code executes once in one timer interrupt and it will be repeated at the rate the timer generates the interrupt. A full control program in C language is in Appendix D.

2.4 Characteristics of Linear Ultrasonic Motor

Fig. 2.10 illustrates the motor control voltage versus velocity characteristic. The motor can be moved in positive or negative direction depending upon the polarity

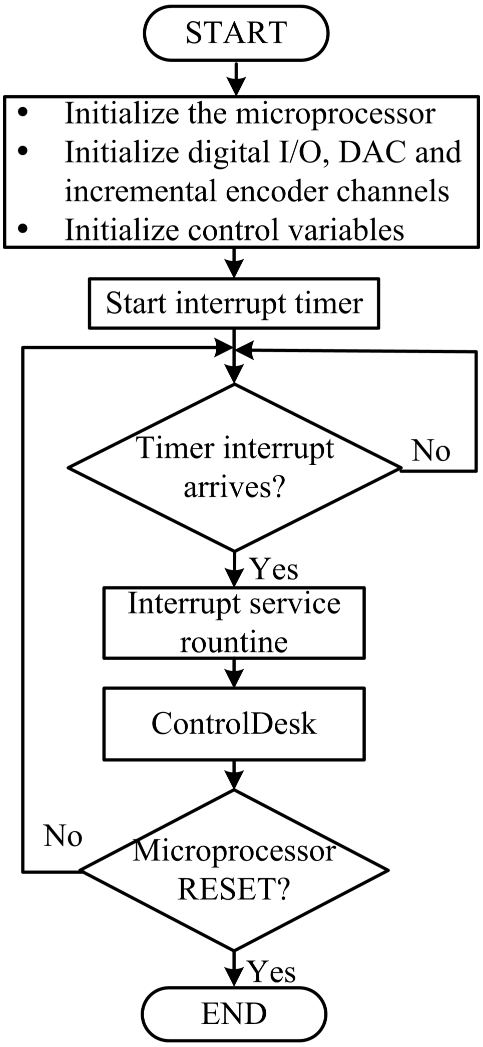

START
Initialize the microprocessor
Initialize digital I/O, DAC and incremental encoder channels
Initialize control variables
Start interrupt timer
Timer interrupt arrives?
No
Yes
Interrupt service rountine
ControlDesk
Microprocessor RESET?
No
Yes
END

Figure 2.9: Flow chart of control program

of the applied control voltage V_c. When a positive control voltage is applied, electrodes A and A' will be excited and the motor moves to the positive direction. Similarly for a negative control voltage V_c, electrodes B and B' will be excited and the motor moves in negative direction. The applied voltage amplitude determines the oscillation amplitude and hence the range of speed and force that the motor can produce. Due to the presence of deadzone in the motor characteristics, the motor does not move till the control voltage exceeds the offset value. This is due to static friction between the spacer and moving stage. The deadzone is almost invariant with the load on the stage since the load on the stage is acting vertically downward while the driving force is acting tangentially sideways on the stage. The friction in the V-flat way is negligible as roller bearing are used. It can also be seen that the deadzone is not symmetric, it is more for positive direction $(0.9\,V)$ than for the negative direction $(\text{-}\,0.8\,V)$ of motion.

Though the static friction gives position holding feature to the ultrasonic motor, it introduces non-linearity in motor the characteristics. Compensation for the deadzone has to be provided at the control level to generate high-performance motion profiles.

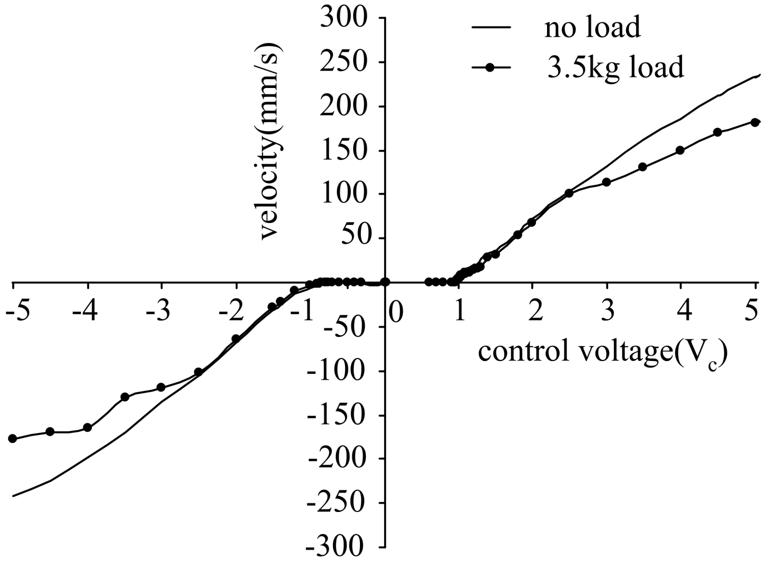


Figure 2.10: Motor Control Voltage-velocity Characteristics

2.5 Summary

The piezoelectric principle is first introduced. The structure and working principle of the linear ultrasonic motor are then discussed. The movement in linear ultrasonic motor is generated by simultaneous excitation of the two orthogonal vibrating modes of the piezoelectric ceramic plate. The complete experiment setup based on DS1104 control board is described. The computer based control setup allows the user to monitor the performance and control the system. The software environment for the implementation of the control schemes is presented. The control algorithm coded in C language is compiled, assembled and loaded into the microprocessor of the DS1104 control board. The open-loop control voltage-velocity characteristics of the motor is obtained experimentally. It has been shown that the motor has deadzone in its characteristics.

The following chapters will present the experimental results for the position control of the ultrasonic motor using the setup shown in this chapter.

Chapter 3

Closed-loop Position Control

This chapter is focussed on the position control of the linear ultrasonic motor discussed in Chapter 2. In the first section, the technique used for compensating the deadzone is presented. Following that, the position control of the linear ultrasonic motor is carried out using the linear proportional-integral (PI) controller. Next a robust position controller, namely the sliding mode controller, is proposed for better position control. Experimental results are shown and discussed for both the controllers. The last section presents a brief summary of this chapter.

3.1 Motor Deadzone Compensation

As discussed in Section 2.4 of Chapter 2, the motor has deadzone in its control-voltage velocity characteristics. In any servo system that has stiction, there should be some compensation for it at the control level. With no compensation for the stiction, the control parameters will not be appropriate. For normally used PID controller, the proportional gain K_p will be raised to high level to overcome the friction. The large K_p may lead to a point of instability and the motor would never be able to settle. For position control of ring type of motor, the deadzone compensation techniques based on fuzzy inference and adaptive techniques have been reported in [22],[23], [39]. In this research, we propose the deadzone compensation scheme based on velocity feedforward and position error signal. If the compensation for the deadzone is provided based on the reference velocity of the trajectory alone, the scheme works well only for continuously changing reference trajectory. For set point tracking, at the set point the reference velocity is zero. There will not be any compensation for the stiction and it takes considerable time for integrator to accumulate the error so that enough control effort is generated to overcome it. Thus this scheme shown in Fig. 3.1 is used for the deadzone compensation. When the reference velocity is zero, the compensation is provided based upon the position error signal. When the position error is within the tolerable limit, no control effort is given. The tolerance for the position error is set to the resolution of the encoder. This scheme works for both the time varying trajectory and set point tracking. The compensation voltages are $u_{ffp} = 0.9\,V$ and $u_{ffn} = -0.8\,V$. With the feedforward deadzone compensation scheme as shown in Fig. 3.1, the

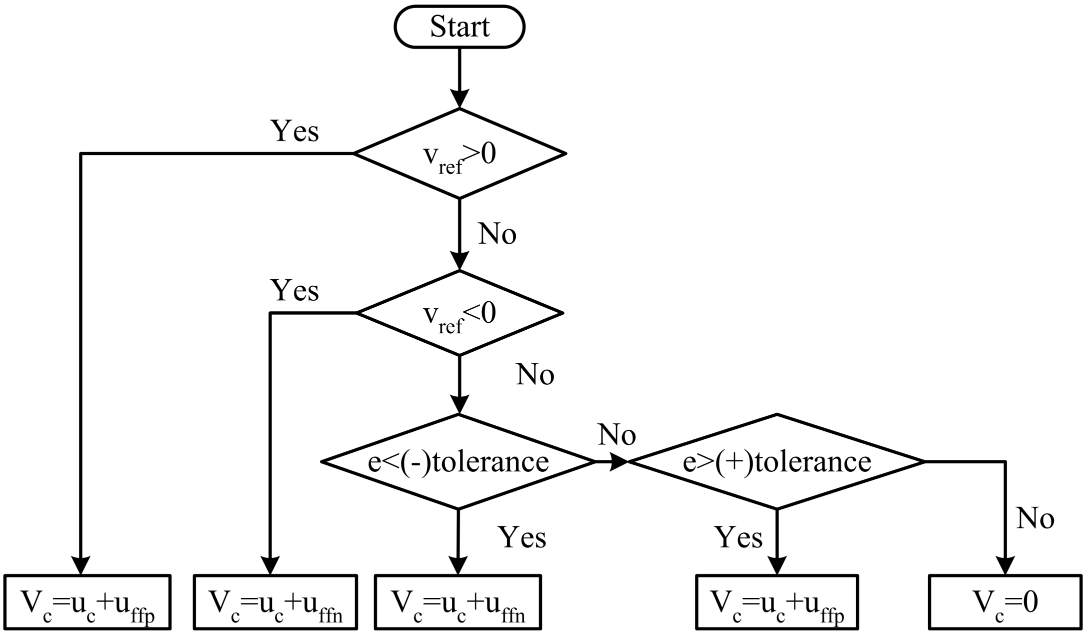


Figure 3.1: Deadzone Compensation Scheme

motor characteristics is somewhat linear as shown in Fig. 3.2. With the deadzone compensation, the velocity with which the motor can move for a control voltage is lesser for loaded condition than on no-load. This is due to the added inertia of the load to the stage.

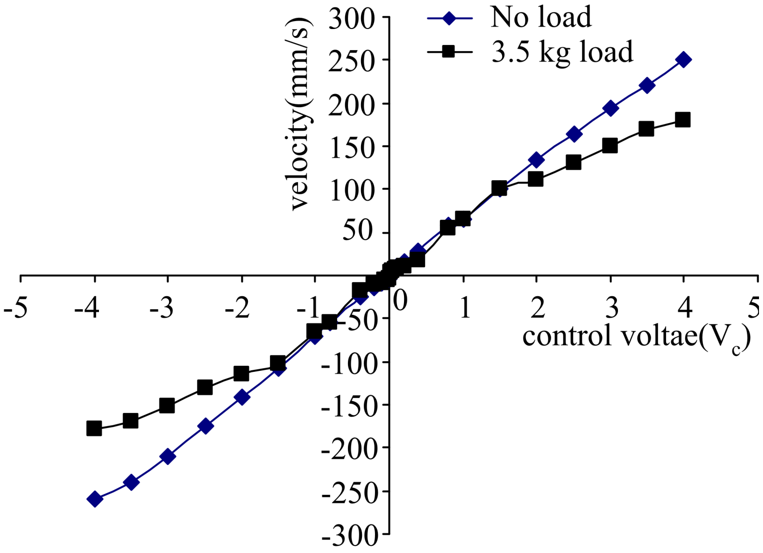


Figure 3.2: Motor control voltage-velocity characteristics with deadzone compensation

3.2 Linear Feedback Position Controller

With the proposed deadzone compensation scheme, the control voltage to velocity relation becomes linear as shown in Fig. 3.2. A linear proportional integral (PI) controller is used for position control of the motor. Due to the lack of the accurate mathematical model of the motor, an experimental approach is used to design the controller. The closed-loop position control scheme is shown in Fig. 3.3 The

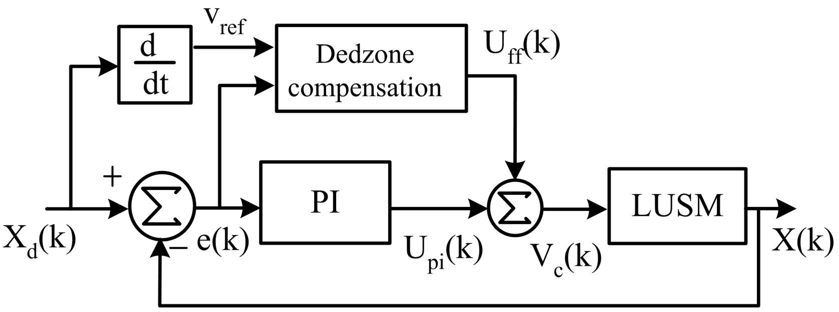


Figure 3.3: PI position control scheme with Deadzone Compensation

PI controller computes the control voltage U_{pi} depending upon the position error e. The PI computed control voltage and the deadzone compensation feedforward signals are added together to form the total control voltage. The magnitude of the control voltage controls the amplitude of the sinusoidal voltage generated by the driver circuitry AB1. The polarity of the control voltage determines the whether electrode A or B is to be excited. The control input to the motor driver from the PI position controller is given as

$$U_{pi}(k) = K_P e(k) + K_I T_s \sum_{i=0}^{k} e(i) \tag{3.1}$$

The gains K_P and K_I of PI position controller are adjusted by trial and error, the values used in the experiment are 35 and 10.5 respectively. T_s is the sampling time and its value is $50\,\mu s$. The values are tuned such that a good set point tracking is obtained with fast rise time and reduced settling time and minimum overshoot.

From the motor characteristics, the velocity of the motor is around $250\,mm/s$ when the control voltage is approximately $4\,V$. This is the maximum velocity rating of the motor and thus the control voltage V_c limited to $\pm 5\,V$. In such cases, the integrator windup problem may occur in the PI position controller. Suppose a large control signal due to a large position reference step causes the control voltage to saturate, then the integrator in the PI position controller keeps integrating the error signal, and the control signal of the controller keeps growing. However, the output of the driver is already at its maximum value, so the error remains large. The increase in the control signal is not helpful since the input to the plant is not changing. The integrator output may become quite large if the saturation lasts for a long time. It will take a considerable negative error signal to bring the integrator output back within the proportional band where the control is not saturated and is ready for subsequent operation. In position control, this may result in large overshoot in the position output. Thus, a typical integrator anti-windup scheme is adopted. The integral action is turned off as soon as the control voltage saturates at $\pm\,5\,V$.

The experiment results for set point position tracking with linear PI position controller is shown in Fig. 3.4. The set point reference is S profiled. The steady state error is $0.1\mu m$, which is the resolution of the encoder used. The motor settles to the reference set point within $0.6\,s$. The steady state error is zoomed and shown. The same repeatable performance is achieved with different set point reference trackings. Using the same PI parameters, a time varying reference position trajectory is tracked. Sinusoidal reference positions are considered for the experiment. Fig. 3.5 shows the tracking performance under no-load. Fig. 3.5(a) shows the sinusoid reference of $0.24\,Hz$ and the actual motor position. The peak tracking error is $18\,\mu m$. The tracking results for the same sinusoidal reference trajectory under $3\,kg$ load is shown in Fig. 3.6. The performance has deteriorated after the introduction of the load. The peak tracking error has increased to $25\,\mu m$. With the load variation, the PI controller requires the control parameters to be tuned to obtain optimal performance. In cases when the motor is run for long time, the temperature of the contact surface increases. It will change the motor friction characteristics and thus controller parameters have to be tuned to get good tracking

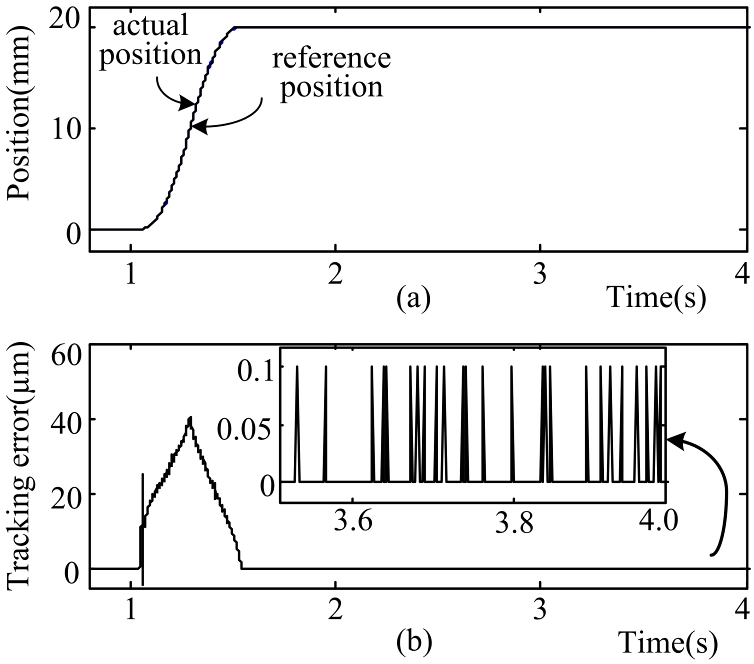


Figure 3.4: Set point position tracking with PI controller: (a) Slider reference and actual position (b) Tracking error

performance. Moreover, the error of 18-25 μm is not good enough for many high performance positioning applications. Thus there is a need for using a better controller for improving the performance. In the next section such a non-linear robust controller *i. e.* sliding mode control has been proposed to improve the tracking performance.

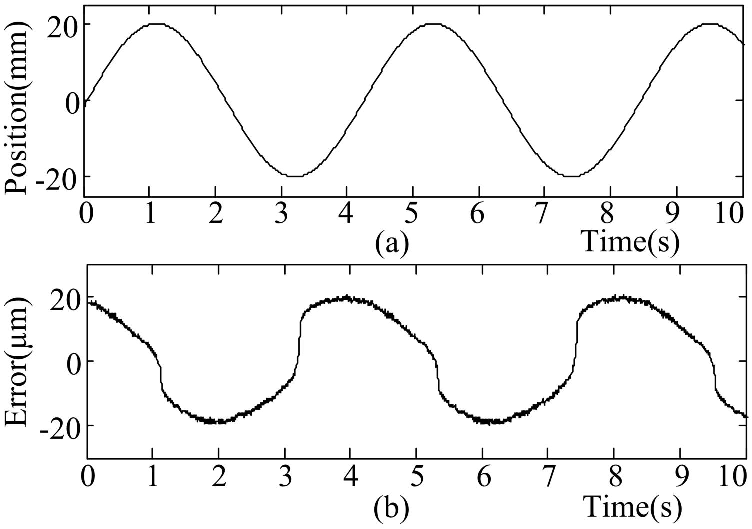


Figure 3.5: Periodic sinusoidal position tracking with PI controller: (a)Actual and Reference position (b) Tracking error under no-load

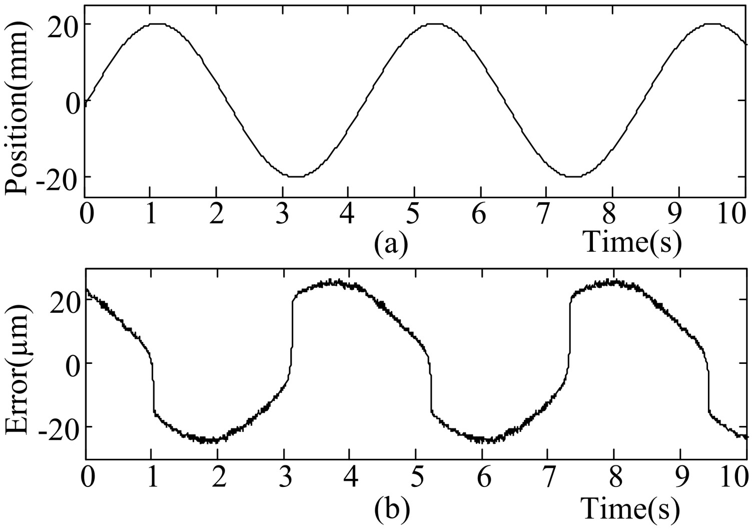


Figure 3.6: Periodic sinusoidal position tracking with PI controller: (a)Actual and Reference position (b) Tracking error under 3kg load

3.3 Sliding Mode Control

During last few decades there has been significant work in sliding mode control (SMC) techniques. The SMC has been successfully applied in many applications such as electric drives, power electronics, power systems etc [40], [41]. Due to order reduction property and its low sensitivity to disturbances and plant parameter variations, sliding mode control is an efficient tool to control complex high-order dynamic plants operating with uncertainty conditions which are common for many processes of modern technology.

The sliding mode control is robust control technique and it requires only the upper bounds of the parametric uncertainties and disturbances. The concept of the sliding mode control is to formulate a variable structure control law such that the system state is forced to certain pre-defined surface, called sliding surface, and it is forced to stay there by appropriate switching of the control structure [42].

Consider the SISO nonlinear system

$$x^n(t) = f(\mathbf{x}, t) + u + d(t) \tag{3.2}$$

where $u \in R$ is the control input, $x(t) \in R$ is the output, $\mathbf{x} = (x, \dot{x}, ..., x^{n-1})^T \in R^n$ is the state vector and $d(t)$ is the disturbance. Assume the nonlinear function $f(\mathbf{x}, t)$ is not exactly known, but

$$f(\mathbf{x}, t) = g(\mathbf{x}, t) + \Delta f(\mathbf{x}, t) \tag{3.3}$$

where $g(\mathbf{x}, t)$ is known function, $\Delta f(\mathbf{x}, t)$ is unknown but bounded by known function F, *i.e.* $|\Delta f(\mathbf{x}, t)| \leq F$ and disturbance $d(t)$ is bounded by β_0. The control objective is to determine a control signal u such that the state $\mathbf{x}$ of the closed-loop system will follow the desired state $x_d = (x_d, \dot{x}_d, ..., x_d^{n-1})^T$ in the presence of modelling uncertainties and disturbance. The sliding surface takes the general form as following

$$\sigma = \left(\frac{d}{dt} + \lambda\right)^{n-1} e \tag{3.4}$$

where $e = x_d - x$ and λ is a positive constant. The dynamics while in sliding mode of the sliding surface $\sigma = 0$ can be written as $\dot{\sigma} = 0$. By solving the above equation formally for the control input from the system dynamics, one obtains an expression for u called equivalent control, u_{eq}, which can be interpreted as the continuous control law that maintains $\dot{\sigma} = 0$ and $\mathbf{x} = \mathbf{x_d}$, if the dynamics were exactly known. The method of equivalent control is a means of determining the system motion restricted to the sliding surface $\sigma = 0$. In sliding mode control, the goal is to determine a switched feedback control law which drives the plant state trajectory to $\sigma = 0$ and keep the system on $\sigma = 0$ for all the subsequent time. The equivalent control is only necessary control signal to maintain the system states in sliding mode, it is not sufficient to drive the system states to the switching surface. A high gain switching control signal is used to drive the system into the sliding mode. Thus the total control law takes the form

$$\begin{aligned} u &= u_{eq} + u_s \\ u_s &= \beta \text{sgn}(\sigma) \end{aligned} \tag{3.5}$$

where

$$sgn(\sigma) = \begin{cases} 1 & \text{if } \sigma > 0 \\ -1 & \text{if } \sigma < 0 \end{cases} \tag{3.6}$$

u_{eq} is a compensation part used to cancel the known dynamics and u_s is the discontinuous or switched part. β is positive value termed as switching gain. In case of no disturbances and uncertainties, it is possible to directly get u_{eq} from the dynamics of the derivative of the switching surface. In the presence of disturbances and uncertainties, it is not possible to get u_{eq} directly, thus the nominal equivalent control u_c is computed from $\dot{\sigma} = 0$ neglecting the disturbance and uncertainties. The control law is

$$u = u_c + u_s \tag{3.7}$$

A variety of control strategies are available for designing sliding mode control. Usually, Lyapunov function candidate is used to design the controller. It is a positive energy like function, which is mathematically defined as a positive-definite function. The controller is designed such that the defined energy keeps dissipating, which is mathematically reformulated as the negative or negative semidefinite property of the time derivative of the energy-like function. Consider the Lyapunov function candidate as

$$V = \frac{1}{2}\sigma^2 \tag{3.8}$$

For sliding mode to occur

$$\dot{V}(t) = \sigma(t)\dot{\sigma}(t) \leq 0 \tag{3.9}$$

For simplicity, consider $n = 2$. Solving equation (3.4) for $\dot{\sigma} = 0$, we get

$$u_c = -g(\mathbf{x}, t) + \ddot{x}_d + \lambda\dot{e} \tag{3.10}$$

Substituting u from equation (3.7) and u_c from equation (3.10) in equation (3.9) we get

$$\begin{aligned}
\dot{V}(t) &= \sigma(t)\dot{\sigma}(t) \\
&= \sigma(t)(\lambda\dot{e} + \ddot{x}_d - \ddot{x}) \\
&= \sigma(t)(\lambda\dot{e} + \ddot{x}_d - f(\mathbf{x}, t) - u - d(t)) \\
&= \sigma(t)(\lambda\dot{e} + \ddot{x}_d - g(\mathbf{x}, t) - \Delta f(\mathbf{x}, t) - u_c - u_s - d(t)) \\
&= \sigma(t)(-\Delta f(\mathbf{x}, t) - u_s - d(t)) \\
&= \sigma(t)(-\Delta f(\mathbf{x}, t) - \beta \text{sgn}(\sigma) - d(t)) \\
&= -\sigma(t)(\Delta f(\mathbf{x}, t) + d(t)) - \beta\sigma(t)\text{sgn}(\sigma) \\
&= -\sigma(t)(\Delta f(\mathbf{x}, t) + d(t)) - \beta|\sigma(t)| \\
&\leq -|\sigma(t)|(\beta - |(\Delta f(\mathbf{x}, t) + d(t))| \qquad (3.11)
\end{aligned}$$

For $\dot{V}(t) \leq 0$, the switching gain β should be

$$\beta \geq |(\Delta f(\mathbf{x}, t) + d(t))| \tag{3.12}$$

Designing the switching and nominal equivalent control as above will satisfy the sliding condition.

Typical sliding mode control is switching ideally at an infinite frequency with a sufficiently high gain to suppress norm-bound disturbance and uncertainties. Combined with the high gain applied, the switching nonidealities relevant to the sampling delay of any digital implementation will cause a chattering phenomenon, which is generally perceived as an oscillation about the sliding manifold. The chattering will be severe with the larger value of β. Thus a relatively small value of β satisfying equation (3.12) is chosen. The discontinuities of the control law may excite the unmodeled high frequency dynamics of the plant which is harmful to actuation mechanism. To avoid this problem, a smoothing scheme, saturation function, is used in place of the signum function.

$$u = u_c + \beta \mathrm{sat}\left(\frac{\sigma}{\phi}\right) \tag{3.13}$$

where

$$sat(\sigma) = \begin{cases} \frac{\sigma}{\phi} & \text{if } |\frac{\sigma}{\phi}| < 1 \\ \mathrm{sgn}(\frac{\sigma}{\phi}) & \text{if } |\frac{\sigma}{\phi}| \geq 1 \end{cases} \tag{3.14}$$

where ϕ is the thickness of the boundary layer introduced to smooth the switching function. Clearly, outside the boundary layer the control law equation (3.13) is equivalent to equation (3.7), which guarantees the sliding condition equation (3.9). While inside the boundary layer the control law equation (3.13) becomes the smooth varying function and sliding condition may still hold good depending upon the value of the β and the system uncertainties. The drawback of introducing the boundary layer to smooth the control action will be reduction in feedback effect and hence reduction in control precision. Thus there is trade-off between the control precision and reduction of chattering.

3.3.1 LUSM Position Control with Sliding Mode Control

The motion equation of the motor is

$$F(t) = M\frac{d^2x(t)}{dt^2} + B_v\frac{dx(t)}{dt} + F_d(t) \tag{3.15}$$

F being the force generated by the piezoelectric vibration, M is the mass of the stage, B_v is the coefficient of viscous friction and F_d is the disturbance due to modelling error and uncertainties. The $F = K_fV_c$, V_c being the control voltage. The motion equation (3.15) can be written as

$$\begin{aligned} \ddot{x}(t) &= -a\dot{x}(t) + b(u(t) + d(t)) \\ d(t) &= -\frac{F_d(t)}{bM} \\ F_d(t) &= F_{fric} + \Delta M\ddot{x}(t) + M\Delta a\dot{\theta}(t) - M\Delta bu(t) \end{aligned} \tag{3.16}$$

where $a = \frac{B_v}{M}$ and $b = \frac{K_f}{M}$. Δa, Δb and ΔM are corresponding variations in system parameters a, b and M respectively. All the system uncertainty $d(t)$ is assumed to be bounded as following

$$|d(t)| \leq \beta_0 \tag{3.17}$$

Consider the state of the system $\mathbf{x}(t) = (x_1(t), x_2(t))^T$ and the desired states as $\mathbf{x_d}(t) = (x_{1d}(t), x_{2d}(t))^T$. The tracking error is defined as

$$\mathbf{e} = \mathbf{x}_d - \mathbf{x} \tag{3.18}$$

The sliding surface is defined as the sliding surface σ as

$$\sigma = \alpha\mathbf{e} \tag{3.19}$$

where $\alpha = [\lambda \quad 1]$.

The nominal equivalent control required to maintain $\dot{\sigma} = 0$, is obtained solving $\dot{\sigma} = 0$ for the system, neglecting $d(t)$.

$$u_c = \frac{M(\lambda(x_{2d} - x_2) + \dot{x}_{2d}) + B_vx_2}{K_f} \tag{3.20}$$

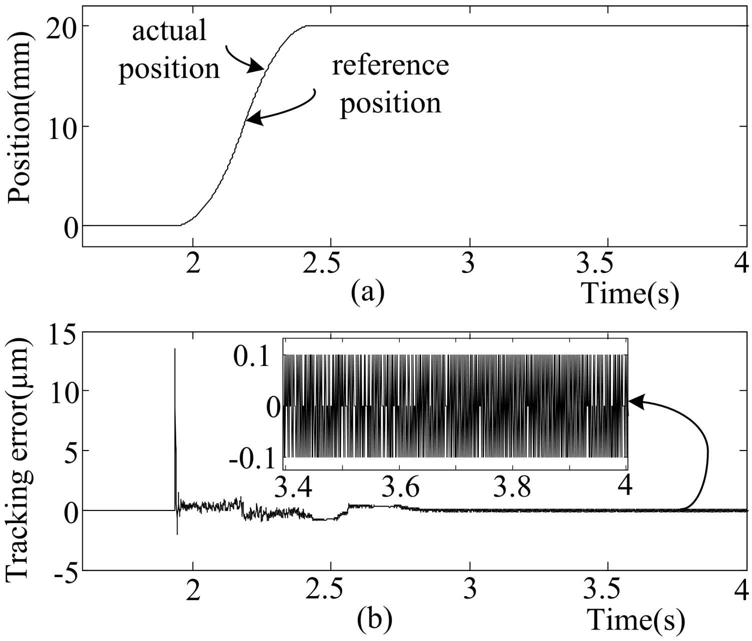


Figure 3.7: Set point position tracking with sliding mode controller: (a) Slider reference and actual position (b) Tracking error

The switching gain β is chosen such that condition equation (3.12) is satisfied and sliding condition can be guaranteed. Thus

$$\beta \geq \beta_0 \tag{3.21}$$

The performance of the controller can be further improved by augmenting a proportional part to the control law (3.22)[43]. The additional term $\alpha\sigma$ forces the motor to approach the switching surface faster when σ is large. Thus the final control law becomes

$$u = u_c + \alpha\sigma + \beta sat\left(\frac{\sigma}{\phi}\right) \tag{3.22}$$

The control law (3.22) has reduced chattering effect and fast sliding trajectory reaching action. The larger value of β and smaller boundary layer is desired to facilitate global attractiveness of the sliding surface and achieve smaller steady state error. But using higher switching gain and small boundary layer results in intense chattering, which is not preferable for the safe operation of the motor. The largest value of the $d(t)$ is estimated to be 0.4. The parameters are chosen such that a compromise is met between tracking accuracy and chattering. The parameters of the sliding mode controller used in experiment are $\lambda = 10000, \beta = 0.4, \alpha = 0.02$ and $\phi = 100$. The parameters of the motor are $M = 0.8kg$, $B_v = 132\, N.sec/m$ and $K_f = 6\, N/V$. Fig. 3.7 shows the set point position tracking results. The steady state error is $0.1\,\mu$m. Though the chattering is reduced by using the *sat* function, it is still there at smaller scale. Same repeatable performances are achieved for different set point position tracking. The results for tracking the same sinusoidal reference trajectory as in Fig. 3.5 with sliding mode controller are shown in Fig. 3.8. The peak error is approximately $10\,\mu$m, which is half of that achieved with PI controller. For the loaded condition Fig. 3.9, the peak error is still of the same magnitude. As the sliding mode controller is robust to disturbances, the tracking error does not deteriorate with load variations. Same performance is maintained when the motor is run for long time causing the motor parameters to change. The lager boundary layer will reduce the feedback gain and result in larger tracking error, while thin boundary layer will result in chattering. The effect of using small boundary layer is shown in Fig. 3.10. With $\phi = 60$, it is seen that chattering occurs as shown in Fig. 3.10 (b) and (c). The occurrence of chattering is easily noticed while doing the experiment from the high pitch noise the motor makes while moving. Thus boundary layer thickness ϕ is chosen slightly higher so smoother operation is achieved. The choice of $\phi = 100$ in above experiment is thus justifiable.

3.4 Summary

In this chapter, a technique to compensate the deadzone in the motor characteristics has been discussed. The inherent deadzone in the motor characteristics leads to nonlinearity in the system. With deadzone compensation, a simple PI control has been used for position control. The PI position controller provides satisfactory performance for set point position tracking but its performance deteriorates for

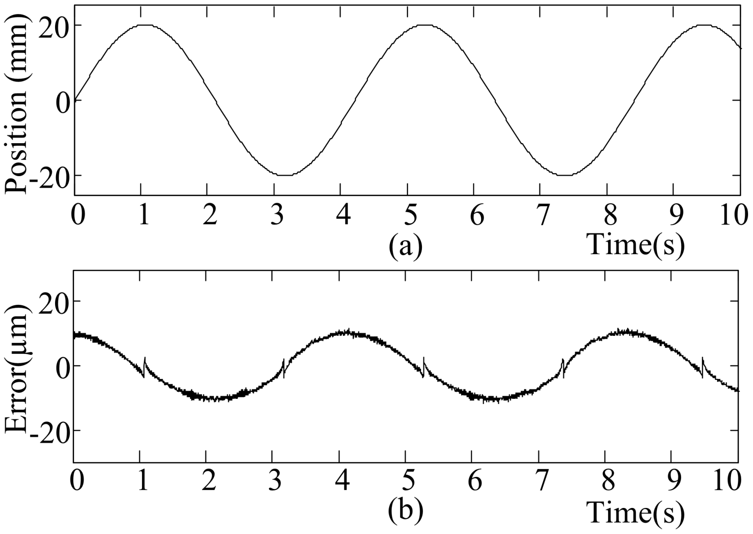


Figure 3.8: Periodic sinusoidal position tracking with sliding mode controller: (a) Reference and actual position (b) Tracking error under no-load

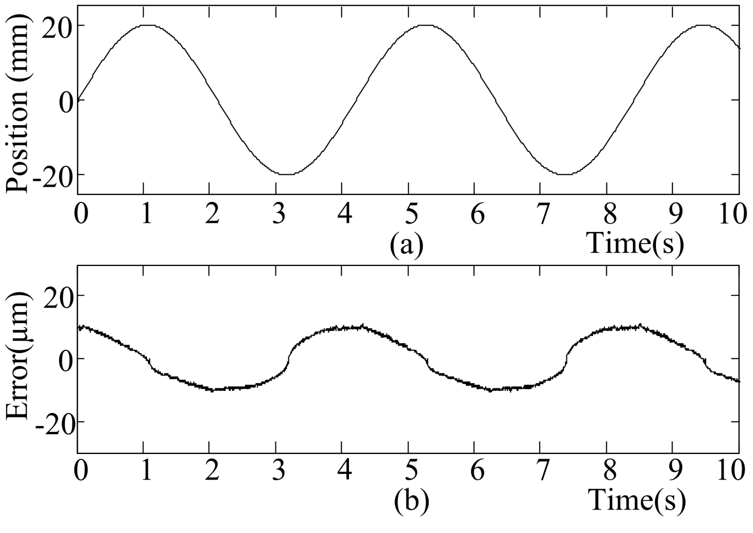


Figure 3.9: Periodic sinusoidal position tracking with sliding mode controller: (a) Reference and actual position (b) Tracking error under 3kg load

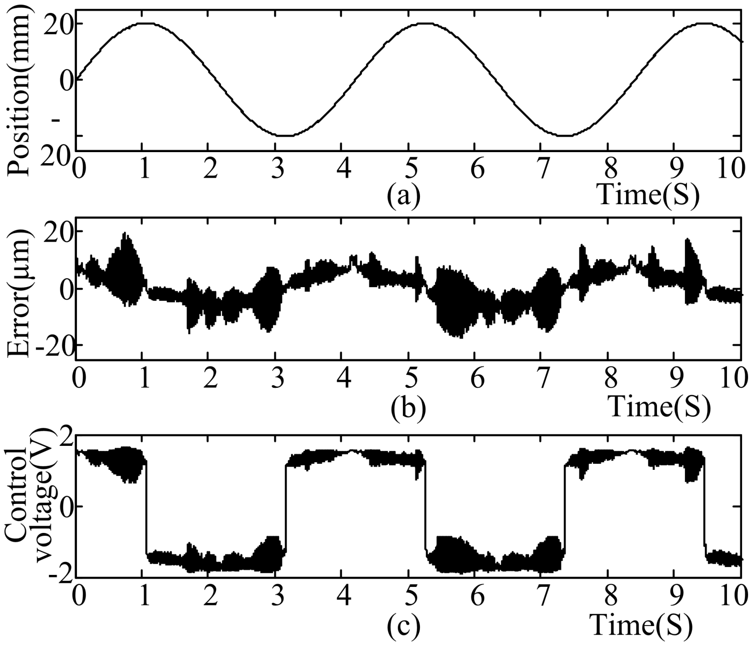


Figure 3.10: Periodic sinusoidal position tracking with sliding mode controller at no-load and reduced boundary layer $\phi = 60$: (a) Reference and actual position (b) Tracking error (c) Control voltage

time varying reference position tracking and under load variations. In order to alleviate this problem a robust sliding mode control has been implemented which gives similar type of performance under no-load and loaded conditions. Chattering in control voltage has been reduced by smoothing the switching control voltage by the introduction of boundary layer in it. With sliding mode position controller, the tracking error has been reduced by a factor of two as compared to PI position controller.

Chapter 4

Sliding Mode Position Control with Closed-loop Filtering

In Chapter 3 sliding mode control was proposed for closed-loop feedback position control of ultrasonic motor. Due to the presence of uncertainties in the system model and disturbances, it was not possible to extract the exact equivalent control. Thus a nominal equivalent control ignoring the uncertainties and disturbances has been used. In this chapter, sliding mode control with closed-loop filtering scheme is proposed. In this control architecture, two filters are used. A low pass filter is used to extract the equivalent control from the switching control signal while another filter is used to reduce the switching gain. Such a scheme has reduced chattering, provides compensation for the disturbance and therefore gives improved performance.

4.1 Introduction

The sliding mode controller has two phases. First phase is the *reaching phase* during which a high switching gain steers the system into the sliding manifold in a finite time. The other phase is called *sliding phase.* During this phase the system dynamics is restricted to the switching surface. The resulting motion is called sliding mode, it results in systems immune to bounded parametric uncertainties and external disturbances. Though sliding mode control results rejection of bounded parametric uncertainties and external disturbances, chattering will incur in a real implementation. Introducing a boundary layer will ameliorate the problem but it will still be there at a lower degree. In order to avoid the chattering problem and to meet systems specifications perfectly, one of the most imperative task in sliding mode control is the acquisition of the equivalent control profile. In tradition discontinuous control switches in sliding mode so as to imitate the equivalent control in the average, which can be defined as the continuous control leading to the invariance conditions for the sliding motion. According to equivalent control methodology, the slow component can be extracted by passing the discontinuous control through a low pass filter whose bandwidth is sufficiently large not to eliminate any slow component but small enough to filter out high frequency switching. However, it is in an open-loop scheme and we cannot use the filtered output equivalent control signal as feedforward term directly. Thus the high switching control still cannot be replaced by the average continuous control.

A sliding mode controller with the closed-loop filtering architecture is proposed for position tracking. Based on the assumption that actual frequency band in equivalent control is much lower than the sampling frequency, a first order filter is employed to acquire equivalent control signal from the switching control signal. This filter alone cannot fully extract the equivalent control signals if its output is fed back to the process as the feedforward compensation. To facilitate the acquisition of the equivalent control and reduce the switching gain a second filter is incorporated to shape the switching gain.

4.2 Principle of SMC with closed-loop filtering

From internal model principle it is known that in order to have a perfect tracking, it is essential to have equivalent component in the sliding mode control. However due to the existence of the system modelling error and disturbances, it is not possible to have direct acquisition of the equivalent control in practical systems. It is shown in [40] that equivalent control can be obtained from switching signals using a first order low pass filter, working as an averaging operator, provided the system is in sliding mode and the filter has infinite bandwidth. The filter needs to have infinite bandwidth only if the equivalent control signal possesses infinite bandwidth which is not the case in most practical systems. Acquiring equivalent control, which in other words incorporating the internal model in the closed-loop, from filtering helps to achieve good tracking performance.

During the reaching phase it is necessary to use high switching gain to ensure global attractiveness of the switching surface and finite reaching time. Once in sliding mode, it is necessary to have the equivalent control component to have a good tracking performance. Due to the presence of system modelling error and existence of disturbances, it is not possible to directly acquire the equivalent control. Thus a low pass filter $LPF1$ is used to extract equivalent control from the switching control signal assuming that the bandwidth of equivalent control is well within the bandwidth of $LPF1$. But once in the sliding phase, keeping high switching gain will result in chattering. A second low pass filter $LPF2$ is used to scale down the switching gain once the system enters sliding phase. The two filters work concurrently in closed-loop. The output of the $LPF1$ will approach to equivalent control only when the switching gain is kept decreasing and switching gain can be reduced to null only when the equivalent control signal is acquired and fed back. This requires the two filters $LPF1$ and $LPF2$ to be activated simultaneously once the system enters the sliding phase. The parameters of the filters are such that once the system operating point is knocked away from the sliding surface due to some unexpected disturbances, the switching gain will surge up and the system will re-enter sliding surface in finite time.

Consider the n^{th} order nonlinear system system

$$\begin{aligned}\dot{x}_i(t) &= x_{i+1}(t), \qquad\qquad i = 1, 2, \ldots, n-1 \\ \dot{x}_n(t) &= f(\mathbf{x},t) + b(\mathbf{x},t)[u(\mathbf{x},t) + d(\mathbf{x},t)] \end{aligned} \tag{4.1}$$

where $t \,\epsilon\, R^+$, $\mathbf{x} = [x_1,\, x_2,\, \ldots . x_n]^T$ is system state, $f(\mathbf{x},t)$, $b(\mathbf{x},t) \neq 0$ are known functions with respect to the arguments. $b(\mathbf{x},t)$ is positive definite. $u(\mathbf{x},t)$ and $d(\mathbf{x},t)$ are system control input and disturbance respectively. The system is required to track the trajectory $\mathbf{x}_d$ described as

$$\begin{aligned}\dot{x}_{id}(t) &= x_{(i+1)d}, \qquad\qquad i = 1, 2, \ldots, n-1 \\ \dot{x}_{nd}(t) &= \xi(\mathbf{x}_d, r(t), t) \end{aligned} \tag{4.2}$$

where $\mathbf{x}_d = [x_{1d},\, x_{2d},\, \ldots . x_{nd}]^T$, $\xi \,\epsilon\, C^1\,[0,\infty)$ with respect to all arguments and $r(t)$ is a reference input. The tracking errors are defined as $e_1 = x_{1d} - x_1$, $e_i = \dot{e}_{i-1}$, $i = 2,\, 3 \ldots n$. It is assumed that $f(\mathbf{x},t)$, $b(\mathbf{x},t)$ and $d(\mathbf{x},t)$ are continuously differentiable, $\forall\,(\mathbf{x},t) \,\epsilon\, R^n \times R^+, R^+ \overset{\Delta}{=} [0,\infty)$. It is also assumed that $d(\mathbf{x},t)$, $b(\mathbf{x},t)$ and their derivatives are upper-bounded as $|d(\mathbf{x},t)| \leq \beta_0$, $|\dot{d}(\mathbf{x},t)| \leq \beta_d$ and $0 < \underline{b} < b(\mathbf{x},t)$ $\leq \bar{b}, \dot{b}(\mathbf{x},t) \leq \bar{b}_d$ respectively. β_0, β_d, $\underline{b}$, $\bar{b}$ and $\bar{b}_d$ are known positive constants.

Considering the switching surface as $\sigma = \sigma(\mathbf{e},t)$, where $\mathbf{e} = [e_1, e_2, \ldots . e_n]^T$ and $\frac{\partial\sigma}{\partial e_n} \neq 0 \;\forall\,(e,t) \,\epsilon\, R^n \times R^+$. The equivalent control input (u_{eq}), which is responsible to maintain $\dot{\sigma} = 0$ is obtained by solving the equation $\dot{\sigma} = 0$.

$$\begin{aligned}\dot{\sigma} &= \frac{\partial\sigma}{\partial t} + \sum_{i=1}^{n-1} \frac{\partial\sigma}{\partial e_i} e_{i+1} + \frac{\partial\sigma}{\partial e_n} \dot{e}_n \\ &= \frac{\partial\sigma}{\partial t} + \sum_{i=1}^{n-1} \frac{\partial\sigma}{\partial e_i} e_{i+1} + \frac{\partial\sigma}{\partial e_n} (\dot{x}_{nd} - f - b(u_{eq} - d)) \\ &= \theta + \gamma(d - u_{eq}) = 0 \end{aligned} \tag{4.3}$$

where $\theta = \frac{\partial\sigma}{\partial t} + \sum\limits_{i=1}^{n-1} \frac{\partial\sigma}{\partial e_i} e_{i+1} + \frac{\partial\sigma}{\partial e_n}(\dot{x}_{nd} - f)$ and $\gamma = \frac{\partial\sigma}{\partial e_n} b$. Thus the equivalent control is given by

$$u_{eq} = u_c - d \tag{4.4}$$

The control input u_c denotes feed-forward compensation term which is to deal with system nominal part and is given by $u_c = \gamma^{-1}\theta$. As seen from equation (4.4), it is clear that u_{eq} cannot be achieved directly due to the existence of disturbance d. In most SMC, switching gain is decided by d. If d is estimated successfully from the switching signal by a low pass filter and if compensated, then the switching gain can be reduced. Keeping the switching gain constant and equal to the same value as before the compensation of disturbance would result in high chattering. The control input from new SMC with closed-loop filtering is

$$u = u_c + k_s(t)u_s + u_v \tag{4.5}$$

where u_s is switching quantity defined as $u_s = sgn(\sigma/\phi)$. The continuous term u_v is generated by low pass filter $LPF1$.

$$\tau_1 \dot{u}_v + u_v = \gamma_1 u_s \tag{4.6}$$

where τ_1 and γ_1 are positive constants. $LPF1$ is activated once the system enters the sliding mode, *i. e.* after the reaching time $t = t_r$, with zero initial condition $u_v(t_r) = 0$. The switching gain $k(t)$ is chosen as

$$k_s(t) = \begin{cases} \beta_0 + \beta_k & 0 \le t \le t_r \\ k(t) & t > t_r \end{cases} \tag{4.7}$$

where $\beta_k > 0$ and $k(t)$ is the output of second low pass filter $LPF2$.

$$\tau_2 \dot{k} + k = \gamma_2 g(\sigma) \tag{4.8}$$

where $g(\sigma) = k_g|\sigma|^q$ is an even function of σ. τ_2, γ_2, k_g and q are positive constants. The initial condition for $LPF2$ is $k(t_r) = \beta_0 + \beta_k$. During the reaching phase the control input is $u = u_c + k_s u_s$. With this control input, convergence of the system to the switching surface is shown in [44]. The parameters of the two filters are chosen to satisfy the conditions stated in (4.9). Such selected parameters of the

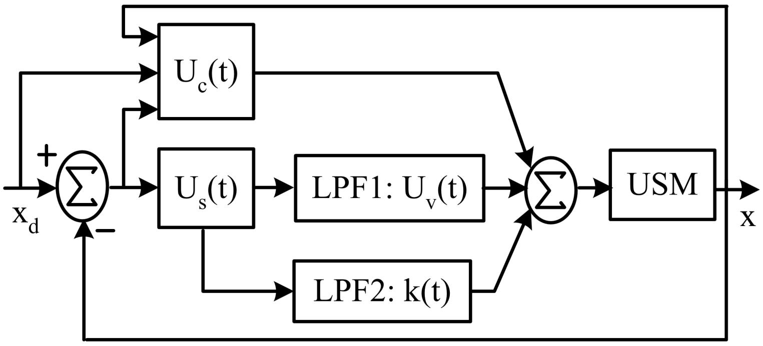


Figure 4.1: Control scheme of sliding mode control with closed-loop filtering

filter will guarantee the maintenance of the sliding motion [44], [45].

$$\begin{aligned} \tau_2 &\geq \tau_1 \\ \gamma_1 &\geq 2(2\beta_0 + \beta_d\tau_1) \\ \beta_k &\geq \varsigma\beta_0, \qquad \varsigma > 1 \end{aligned} \tag{4.9}$$

To faithfully acquire the equivalent control, τ_1 should be made as small as possible so as to increase the bandwidth of $LPF1$. τ_2 and γ_2 affects the attenuation rate of the switching gain $k(t)$ after entering the sliding motion. A smaller τ_2 will render a faster reduction of $k(t)$. A large γ_2 and fast increasing function $g(\sigma)$ will facilitate the switching gain recovery.

4.3 Application of SMC with Closed-loop Filtering to Position Control of LUSM

This control scheme for position control of linear ultrasonic motor is shown in Fig. 4.1. The simplified motor model in state space form is

$$\begin{aligned} &\dot{x}_1 = x_2 \\ &\dot{x}_2 = -\tfrac{B_v}{M}x_2 + \tfrac{K}{M}(u(\mathbf{x},t) - d(\mathbf{x},t)) \\ &y = x_1 \end{aligned} \tag{4.10}$$

where M is the net mass of the slider and d is bounded disturbance. The system is to track $[x_{1d}, x_{2d}]$. The switching surface is chosen to be same as equation (3.19).

$$\sigma = \lambda\, e_1 + e_2 = \lambda(x_{1d} - x_1) + (\dot{x}_{1d} - \dot{x}_1) \tag{4.11}$$

The feed-forward compensation control input is

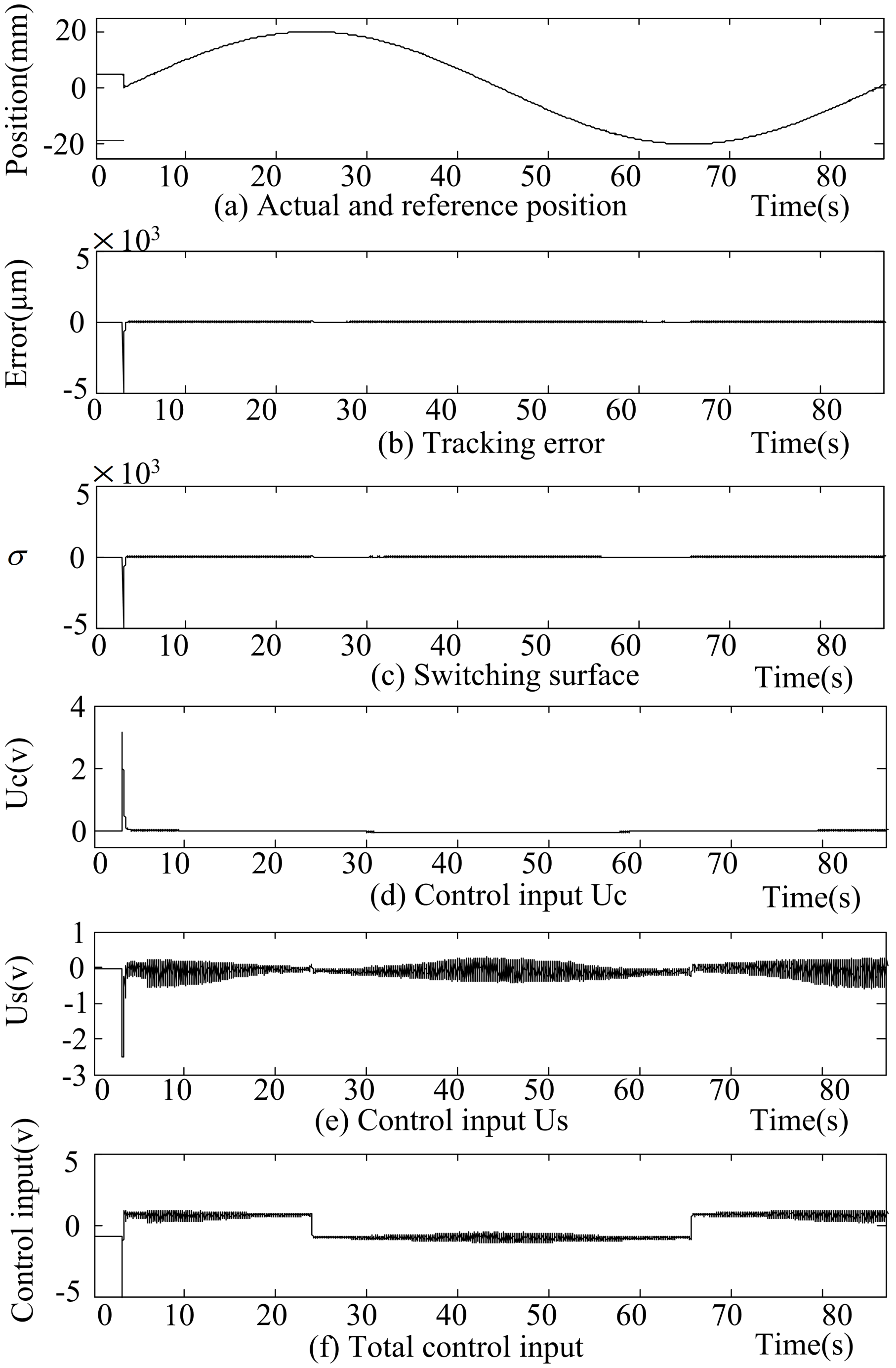
$$u_c = (M(\lambda\dot{e} + \dot{x}_{2d}) + B_v x_2)/K \tag{4.12}$$

The switching control is $u_s = sat(\sigma/\phi)$. The sat function is used in place of sgn function because using sgn function in practical system results in intense chattering. In this experiment the initial conditions are $x_1(0) = 5\,mm$, $x_2(0) = 0\,mm/s$ and other control parameters are same as in Chapter 3. The control input u_v and switching gain k_s are generated by $LPF1$ and $LPF2$ respectively upon entering the sliding surface. The parameters of the filters are chosen such that they satisfy the conditions (4.9). In addition to those conditions, τ_1 should be made as small as possible so as to increase the bandwidth of $LPF1$ to capture the u_{eq}. A small τ_2, a large γ_2 and fast increasing function $g(s)$ are used to facilitate fast switching gain recovery in case operating point is knocked out of sliding surface. The values of β_0 and β_k are 1.0 and 1.5 respectively. The parameters of two filers are: $\tau_1 = \tau_2 = 50\,ms$, $\gamma_1 = 3$, $\gamma_2 = 1$ and $g(\sigma) = \sqrt{|\sigma/10|}$. Thus $LPF1$ and $LPF2$ are

$$\begin{aligned} LPF1: \quad 0.05u_v + u_v &= 3u_s \\ LPF2: \quad 0.05k + k &= \sqrt{|\sigma/10|} \end{aligned} \tag{4.13}$$

The above filter equations are discretized at $50\,\mu s$ samplng time and implemented in digital processor.

First the experiment results for tracking a reference $x_{d1} = 20sin(0.0754t)\,mm$, $x_{d2} = 1.508cos(0.0754t)\,mm/s$ without filtering scheme are shown in Fig. 4.2. As shown in the figure, the system enters the sliding mode in finite time and remains on the sliding surface. Though small in magnitude, there is chattering taking place as


(a) Actual and reference position

(b) Tracking error

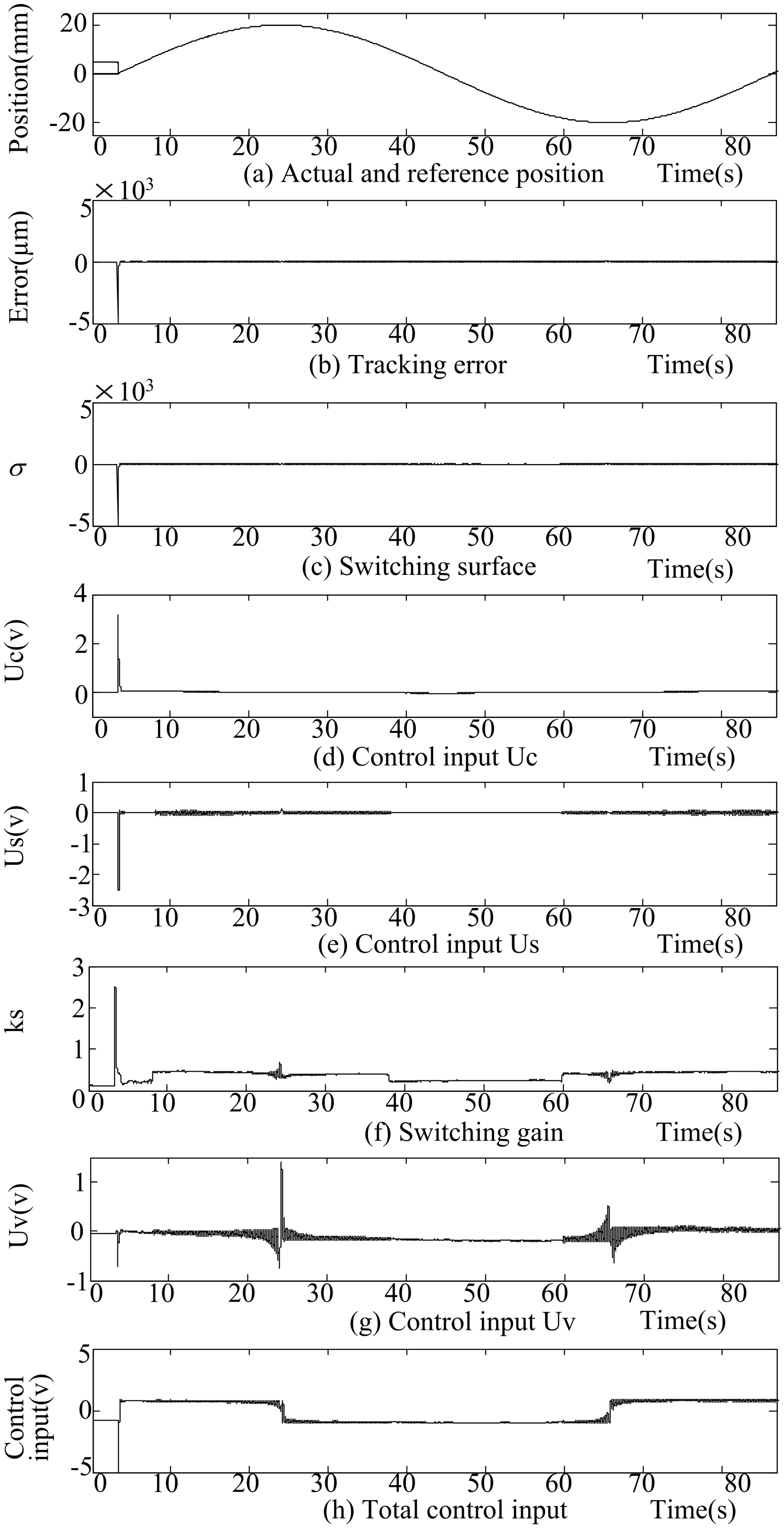
(c) Switching surface

(d) Control input Uc

(e) Control input Us

(f) Total control input

Figure 4.2: Position tracking with SMC without closed-loop filtering



(a) Actual and reference position

(b) Tracking error

(c) Switching surface

(d) Control input Uc

(e) Control input Us

(f) Switching gain

(g) Control input Uv

(h) Total control input

Figure 4.3: Position tracking with SMC with closed-loop filtering

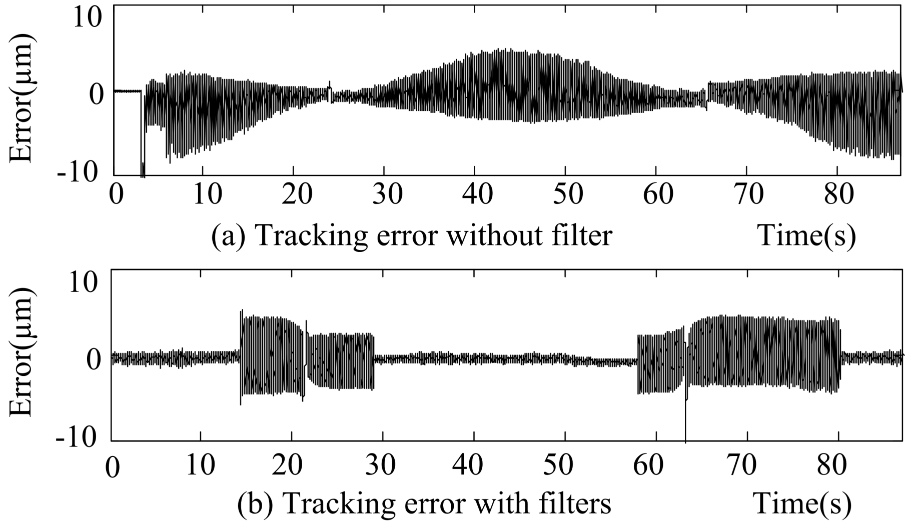


Figure 4.4: Tracking error comparison of SMC's with and without closed-loop filtering

can be seen from plot for u_s in fig. 4.2(e). The chattering results in higher tracking error, wear and tear in the motor. The effect of chattering is clear from the sound the motor makes while moving. The observations for tracking the same trajectory using SMC with closed-loop filtering is shown in Fig. 4.3. It is seen that the system enters the sliding phase in finite time after which two filters are activated. The switching signal is filtered to provide compensation for disturbance, which is friction in this case. Concurrently the switching gain is scaled down to reduce the chattering. It is seen that the switching gain slightly increases at the turning points where the large effect of friction is experienced by the slider. The switching control signal as shown in Fig. 4.3 (e) is smoother than without filtering (Fig. 4.2 (e)). The comparison of the tracking errors for SMC alone and SMC with closed-loop filtering are shown in Fig. 4.4. The tracking error in the case with closed-loop filtering is less than that with SMC alone in majority part of the trajectory period.

The robustness of the scheme is shown by introducing a disturbance into the system. The observations are shown in Fig. 4.5. In Fig. 4.5 (b) a impulse of $4\,V$ (i.e. $24\,N$) is given to the system at the control voltage input of the AB1A

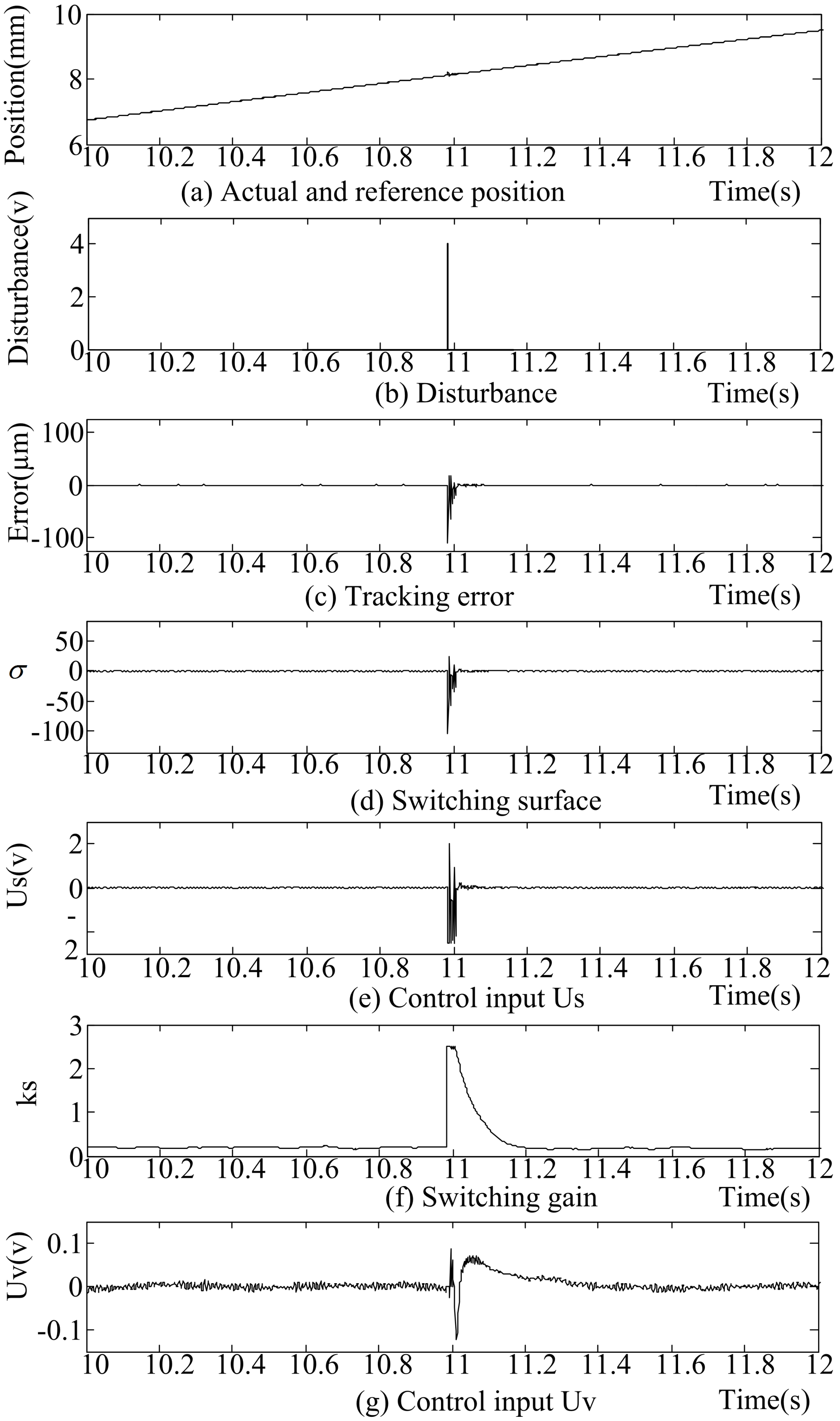


Figure 4.5: Performance of SMC with closed-loop filtering under sudden disturbance in control voltage

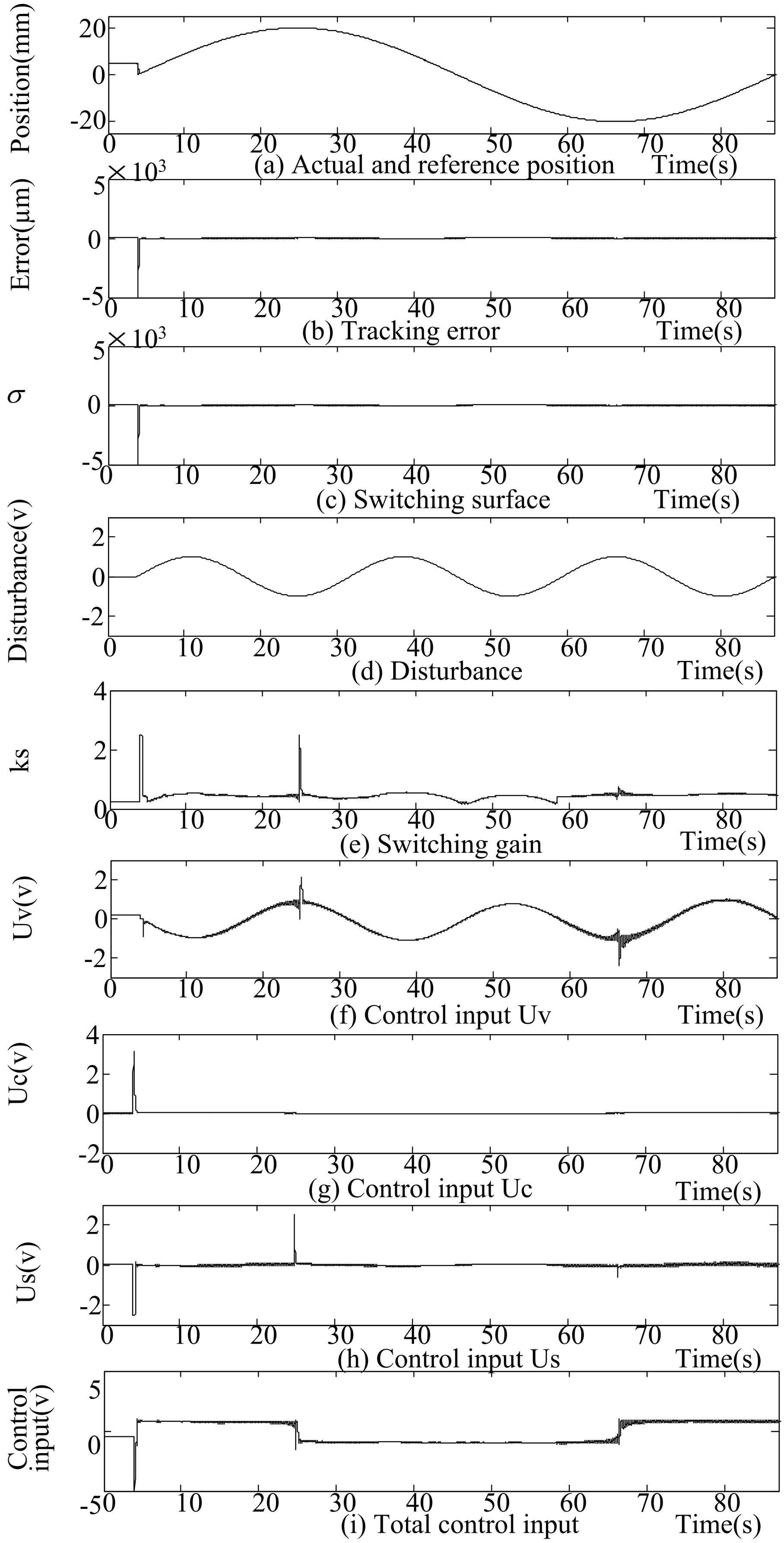


Figure 4.6: Performance of SMC with closed-loop filtering under disturbance in control voltage

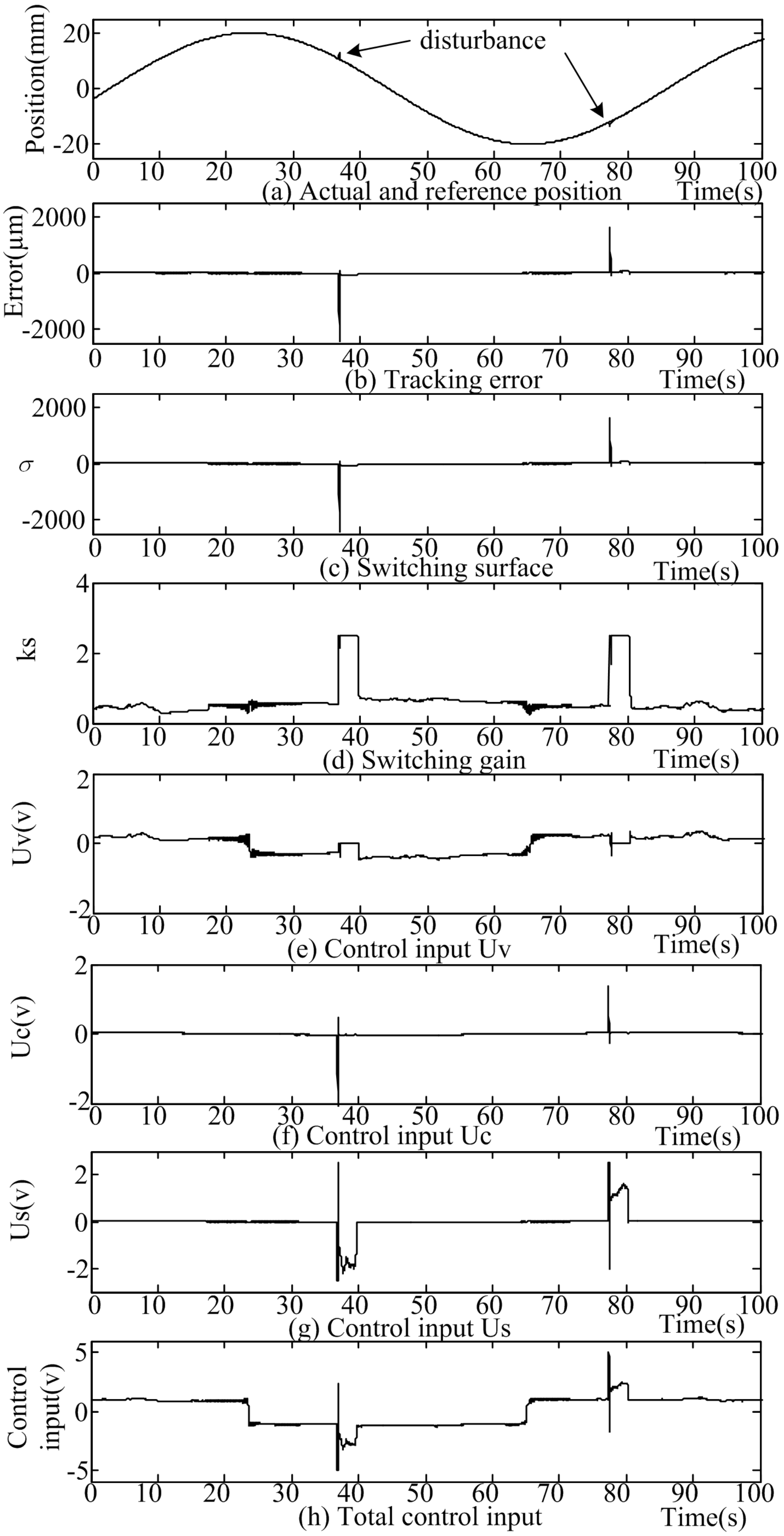


Figure 4.7: Performance of SMC with closed-loop filtering under external disturbance on moving stage

driver. A sector of the sine trajectory is expanded and shown. Though the system is knocked out of the sliding surface, as shown in Fig. 4.5 (f) the switching gain increases and pushes the LUSM back into sliding phase. Once the LUSM enters the sliding phase, the switching gain is reduced again to reduce the chattering. A disturbance of $d(t) = 1sin(0.0754t)$ is given into the LUSM system. The tracking results are shown in Fig. 4.6. It is noticed that even in the presence of disturbance the tracking performance is good. It is due to the fact that chattering is kept at lower level due to scaled down switching gain (Fig. 4.6 (e)) and compensation for the disturbance is provided by filtered control signal u_v (Fig. 4.6 (f)). It is seen from Figs. 4.6 (d) and (f) that disturbance is well approximated by the filter. The disturbance of $0.036\,Hz$ is well within the filter $LPF1$ bandwidth and thus it is able to approximate it faithfully. The slight variation on u_v is due to the fact that $LPF1$ is also approximating the frictional disturbances. We notice the peak occurs in u_v at the turning points. It is due to the fact that the slider experiences maximum effect of friction at these points. For the above disturbances, disturbance voltages were introduced at the control input terminal of the motor driver AB1A. Fig. 4.7 shows the experimental result in the case when an external disturbance comes on the moving stage. This kind of disturbances are more likely to occur in the practical applications. The disturbance is created by hammering the moving stage. Upon the appearance of the disturbance, the operating point is displaced from the switching surface. The switching gain surges up and the operating point moves into sliding surface. Upon reaching the sliding surface, the switching gain is reduced and chattering is reduced. Thus the experimental results verify the proposed scheme has less chattering, improved tracking performance, robustness and ability to estimate the disturbance in the system.

4.4 Summary

Using SMC for position control of ultrasonic motor, where exact model of the system is not known, results in chattering. A SMC with closed-loop filtering scheme is used for the position control of the LUSM. A low pass filter is used to extract the equivalent control from the switching signal. Concurrently, a second low pass filter is used to scale down the switching gain. This scheme is able to estimate the disturbance and compensate it. The robustness of the variable structure control is maintained and at the same time it has reduced chattering. The scheme is easy to implement as only two simple low pass filters are involved. Experiment results obtained proves that the scheme is useful in position control of the motor. Robustness of the scheme is shown by introducing impulse disturbance into the system. Further, it is shown that the low pass filter is able to approximate the disturbance in the system and provides compensation for it.

Chapter 5

Position Tracking Performance Improvement with Iterative Learning Control

In this chapter, iterative learning control (ILC) is presented to improve the tracking performance of existing controller for repetitive position tracking of ultrasonic motor. For repetitive reference position tracking, the position tracking error can be reduced by updating the control action in each cycle based on the error on the previous cycle. Such ILC scheme can be implemented in the existing drive system without any extra hardware modifications. The scheme requires an adequate size of memory and elementary algebraic computation.

First a brief review of ILC is given. It is followed by the application of the ILC to ultrasonic motor drive system. The effectiveness of the ILC is substantiated by the experiment results.

5.1 Introduction

The conventional control design methods require the parametric model of the practical system to be controlled. If the description of the system is available, the optimal solution is to inverse this description (if possible) and use this to compute the input that produces the desired output. For most practical systems, it is difficult to formulate an exact mathematical model. Thus, it is obvious for practical systems that the inverse dynamics approach will never achieve a perfect tracking. If the structure of the system model is known, though the exact values of the parameters may not be known, adaptive control technique can be used [46]. Adaptive controller estimates the unknown information and uses the information to improve the performance by tuning the control parameters. Adaptive controllers assure good performance as long as the structure of the system description is correct.

Iterative learning control is an alternative to the inverse dynamics and adaptive control approaches. The motivation of iterative learning control comes from the fact that the knowledge can be acquired from experience. The term iterative learning itself vividly explains its meaning. The word iterative refers to the process that executes the same trajectory over and over again. The word learning refers to the idea that by repeating the same action, anything should be able to perform it better. The last word control emphasizes that the result of the learning is used to control the plant. Thus, ILC will introduce new control signal based on learning from previous trials. Using the information from the previous trials as a new source of knowledge related to the dynamic process model, reduces the need for the process model knowledge.

Arimoto and co-workers were the pioneer researchers to introduce and apply iterative learning control (ILC) to a robot performing a repetitive task [47]. The main strategy of the ILC is to improve the performance of the system being controlled iteratively using the information obtained from the previous trials, and eventually to obtain an optimal control input that gives desired output [48]. It is to be noted that learning control differs from conventional adaptive control. Most adaptive control schemes are online algorithms that adjust the controller's parameters every sampling instant till a steady-state equilibrium is reached. In a learning control scheme, it is the control input that is varied (in an off-line fashion), over each trial or repetition of the system. It should also be pointed out that ILC is not an open-loop operation, although ILC only modifies the input command for the next repetition, ILC is closed-loop in repetitions since updates are performed for the next repetition using the feedback measurements of the previous repetition. This is similar to closed-loop structure of conventional controllers in time axis which updates the control signal of the next time step using the feedback at current or past time steps. The difference is that the thinking in ILC is in repetition domain which makes it appear as open-loop control in time domain.

Much work has been done recently on the application of ILC in servomechanism systems, especially in robotic manipulators, mechanical processes such as metal rolling, and chemical processes. ILC has been recently applied for the reduction of torque ripple in switch reluctance drives [49], [50] and for permanent magnet motors [51]. For high precision motion control ILC has been reported in [28] and [52]. It has been observed that such plug-in learning controller helps to improve largely the performance of existing controller.

5.2 Iterative Learning Control

The main idea of the iterative learning control is to utilize the information available when a plant with uncertainties carry out the same task several times. In such cases, the error in the output response will be repeated during each trial of the task. It will be possible to improve the performance of the control system by using the results from the previous trials. Refinements are made to the control input signal given to the plant using the tracking information from the previous trials until desired performance level is reached at the output of the plant. The control scheme with iterative learning control is illustrated in Fig. 5.1. The figure shows how the control signal is developed in iterative learning scheme from one iteration to the next. The basic idea of iterative learning control is illustrated in Fig. 5.2. All the signals used in Fig. 5.2 are defined on the finite time interval $[0, T_p]$. The subscript k indicates a specific trail.

The scheme operates as follows. During the r^{th} sample instant of the k^{th} trial, an input signal $u_k^{ilc}(r)$ is applied to the plant, thus resulting the output signal $y_k(r)$. The error signal is computed as the deviation of $y_k(r)$ from desired response $y_{dk}(r)$. At r^{th} sample instant, the control input $u_{k+1}^{ilc}(r)$ for the next trial is computed using the ILC control law (5.1).

$$u_{k+1}^{ilc}(r) = Qu_k^{ilc}(r) + Le_k(r) \tag{5.1}$$

The computed new control input signal is stored in a memory buffer of length T_p. Initially the memory used in ILC is initialized to zero and it is updated in successive cycles using the control law (5.1). The importance of the modification of the control signal is to reduce the error $e_k(r)$. ILC algorithm will help to find an optimal control input that minimizes the error $e_k(r)$ in an iterative manner. A major issue when applying ILC is the convergence of the tracking error. The iterative update law of the input signal and the tracking error has to converge to a unique equilibrium signal $u_\infty(r)$ and $e_\infty(r)$, thereby giving good performance. For learning control to be applicable, the following postulates have to be satisfied.

- A_1: A plant repeatedly performs a specific motion that ends in a fixed duration $(T_p > 0)$.
- A_2: The desired output $y_d(r)$ with $r\epsilon[0, T_p]$ is given a priori.
- A_3: For each trial the initial states are the same.
- A_4: The plant dynamics are time-invariant through out all iterations.
- A_5: The plant output $y(r)$ is measurable without noise.
- A_6: There exists a unique input, $u_d(r)$ that yields the desired output $y_d(r)$.
- A_7: The system is stable in closed loop.

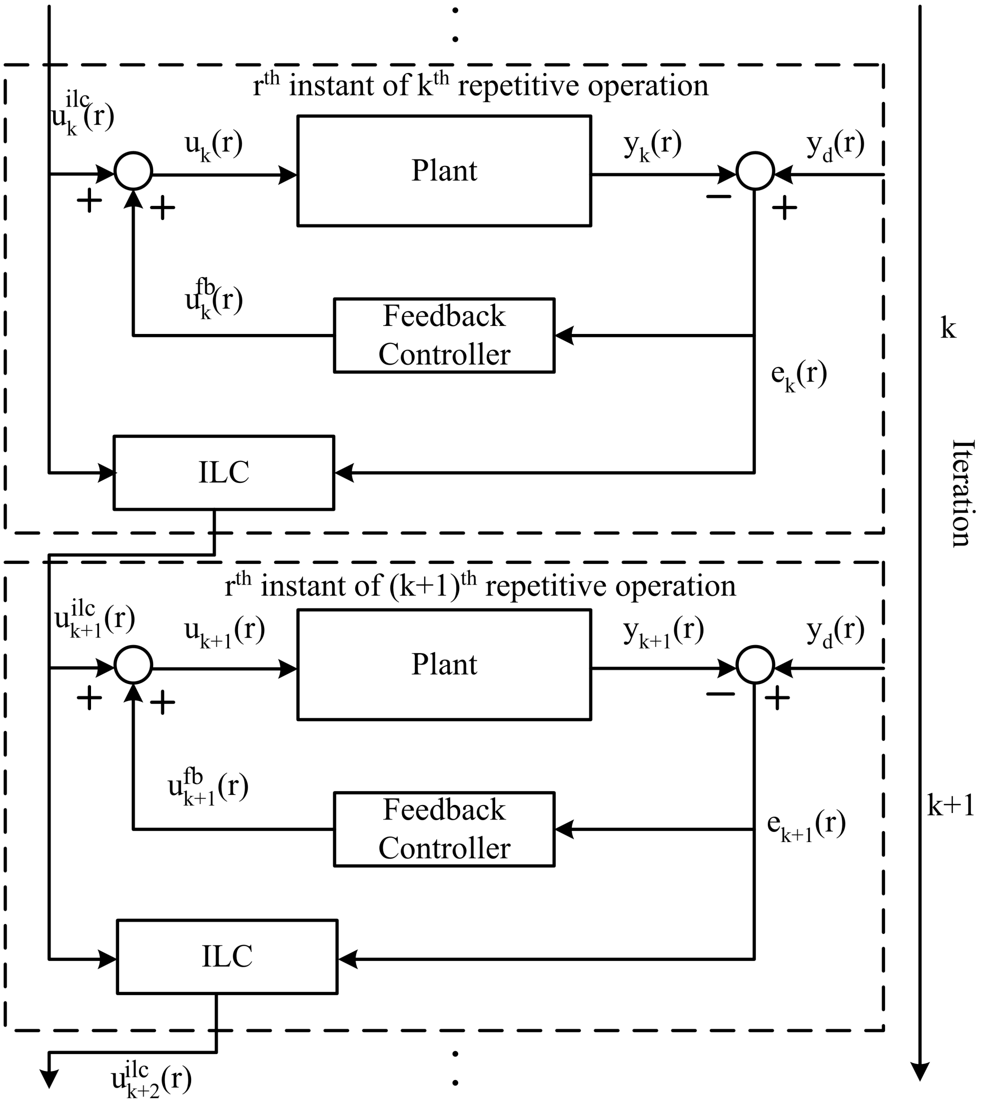


Figure 5.1: Block-Diagram of Iterative Learning Controller for System with a Feedback Controller

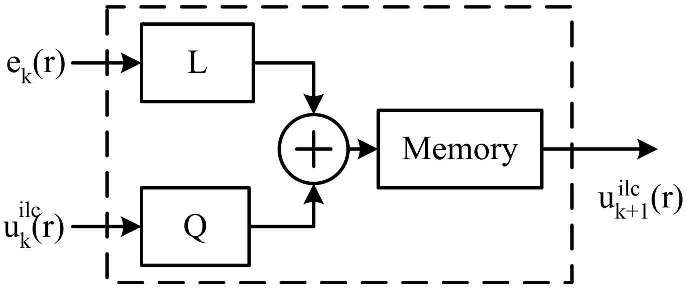


Figure 5.2: Block-Diagram of Iterative Learning Controller

From Fig. 5.1 and Fig. 5.2, considering the plant and the feedback controller transfer function to be G_p and G_{fb} respectively, we have

$$\begin{aligned} u_{k+1} &= Qu_k^{ilc} + Le_k + G_{fb}e_{k+1} \\ &= Q(u_k - u_k^{fb}) + Le_k + G_{fb}e_{k+1} \\ &= Qu_k + (L - QG_{fb})e_k + G_{fb}e_{k+1} \end{aligned} \tag{5.2}$$

Further considering $Q = 1$, we get

$$\begin{aligned} e_{k+1} &= y_d - y_{k+1} \\ &= y_d - G_p[u_k + (L - G_{fb})e_k + G_{fb}e_{k+1}] \\ (1 + G_pG_{fb})e_{k+1} &= y_d - G_p[u_k + (L - G_{fb})e_k] \\ \frac{e_{k+1}}{e_k} &= \frac{1 + G_pG_{fb} - G_pL}{1 + G_pG_{fb}} \end{aligned} \tag{5.3}$$

For convergence of the output trajectory to the desired one, the following condition should be satisfied.

$$\begin{aligned} \left\|\frac{e_{k+1}}{e_k}\right\| &= \left\|\frac{1 + G_pG_{fb} - G_pL}{1 + G_pG_{fb}}\right\| < 1 \\ \left\|\frac{e_{k+1}}{e_k}\right\| &= \|1 - G_cL\| < 1 \end{aligned} \tag{5.4}$$

where G_c is the closed loop system transfer function with feedback controller G_{fb}. The main concern in designing the iterative learning control is to choose L such that the learning convergence condition (5.4) is always satisfied.

5.3 Application of Iterative Learning Control to Position Control of LUSM

The position controllers we have designed in the previous chapters considered the simple model with all non-linearities lumped as disturbance in the system. A better tracking performance can be assured if the exact model of these non-linearities are known. Identifying these non-linearities exactly is a herculean task. As discussed above, iterative learning control is a control technique that helps to get better control performance when motor is doing a repetitive task. The iterative learning control will learn the motor dynamics from iteration to iteration and an improved positioning performance can be achieved.

In many applications, the LUSM is subjected to the periodic reference position tracking. In the industrial applications like spot welding, robotics, pick and place jobs, the motor will be subjected to repetitive position reference. In such cases the tracking errors are also periodic, repeating at the same frequency as the reference position. As seen in Chapter 3, Sections 3.2 and 3.3, using a simple linear position controller or a robust nonlinear controller for position control, significant repetitive errors are observed. To reduce the periodic tracking error an iterative learning controller is added to the exiting sliding mode controller as shown in Fig. 5.3. As stated in Section 5.2, for applying ILC those seven postulates have to be satisfied. But for practical systems it will very difficult to meet all those conditions like same initial states for each iteration and time-invariant plant dynamics. A forgetting factor δ is introduced in learning process, the robustness of ILC in the presence of initialization errors, fluctuations of the plant dynamics and measurement noise is increased. With the introduction of forgetting factor the postulates (A_3) to (A_5) are relaxed as following:

- A_3': The repeatability of the initial condition is satisfied within a small al-

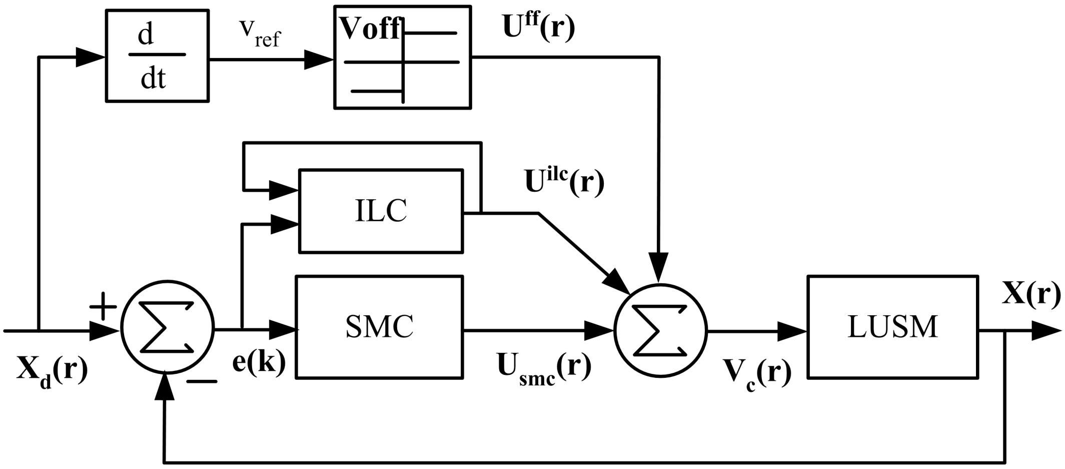


Figure 5.3: Configuration for closed-loop position control of LUSM with ILC and SMC

lowable deviation, i.e., $x_k(0) - x_0 = \delta_k$, $\parallel \delta_k \parallel < \varepsilon_1$ for some small $\varepsilon_1 > 0$.

- A_4': A small level of fluctuations in plant dynamics are is allowable.
- A_5': The plant output $y(r)$ is measurable within small specified noise level $\xi_k(r)$, i.e., $e_k(r) = y_d(r) - \{y_k(r) + \xi_k(r)\}$ where $\|\xi_k(r)\| \leq \varepsilon_3$ for some small $\varepsilon_2 > 0$.

The learning control law with forgetting factor is

$$u_{k+1}^{ilc}(r) = (1-\delta)u_k^{ilc}(r) + Le_k(r), \qquad 0 < \delta < 1 \tag{5.5}$$

The introduction of the forgetting factor provides additional robustness to the ILC scheme. However, the convergence of the output trajectory will be limited to an δ neighborhood of the desired one.

The switching control of sliding mode controller used for position control of the motor is approximated as PD control within the boundary layer. The equivalent control is approximately taken as $u_{eq} = \frac{M\lambda\dot{e}+B_v x_2}{K_f}$, considering the feedforward control term $\frac{M\dot{x}_{2d}}{K_f}$ as disturbance. The approximate closed-loop transfer function of SMC with ILC control scheme thus obtained is

$$G_c = \frac{K_f}{Ms^2 + (K_f/\phi + \lambda M)s + K_f\lambda/\phi} \tag{5.6}$$

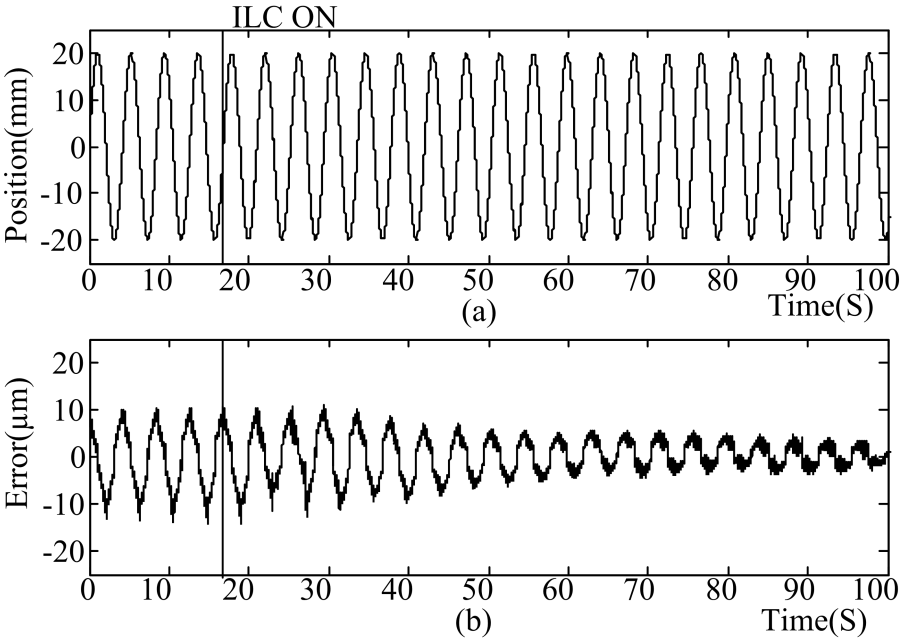


Figure 5.4: Position control of LUSM using SMC with ILC under no-load with learning gain $L = 5$: (a) Reference and actual position (b) Tracking error

With the learning gain of $L = 10$ and sliding mode control parameters as in Chapter 3, the convergence condition (5.4) is satisfied for frequency range upto 1.0545 Hz. To increase the stability of the learning process, a zero-phase low pass filter is implemented to filter the error signal. The cut off frequency of the filter is set 1Hz. Choosing a small learning gain will take larger number of learning iterations to achieve desired control signal while using larger learning gain can amplify the noise and non-linear factors causing learning to diverge. The experimental results for the same reference trajectory as in Fig. 3.5 with two different learning gains are shown in Fig. 5.4 and Fig. 5.5. With small learning gain $L = 5$, it takes large number of iterations, more than twenty, to learn the desired control voltage. Using larger learning gain $L = 15$, the learning convergence condition 5.4 is violated for frequencies within passband of low pass filter and the learning process does not converge. Thus the learning gain is chosen as a compromise between fast learning and stability of the learning process. With the parameters of the sliding mode controller same as before and that of ILC as $\delta = 0.01$ and $L = 10$, the experimental

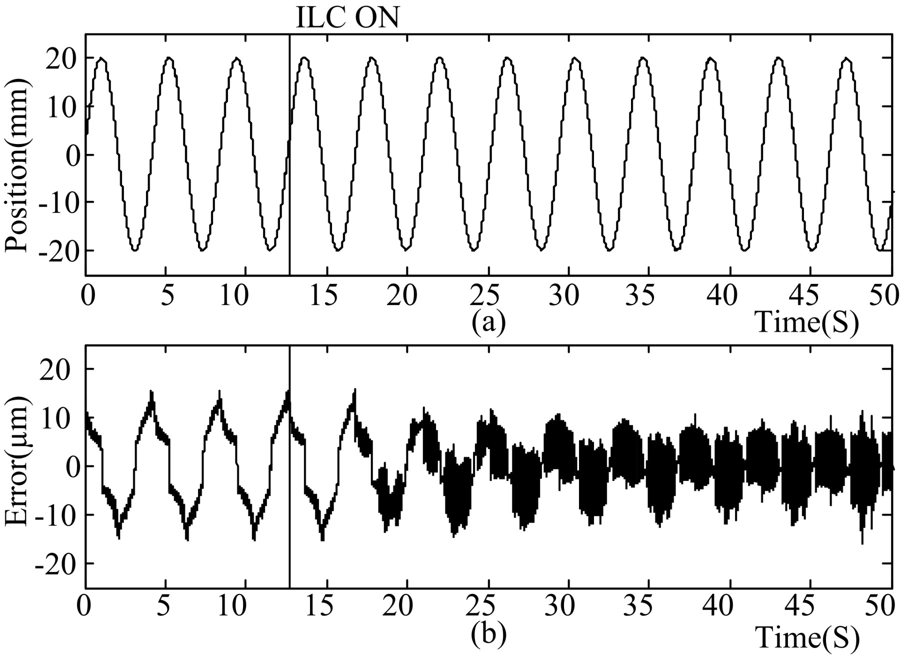


Figure 5.5: Position control of LUSM using SMC with ILC under no-load with learning gain $L = 15$: (a) Reference and actual position (b) Tracking error

results obtained for the same reference trajectory tracking as in Fig. 3.5 are shown in Fig. 5.6 and Fig. 5.7 respectively. Fig. 5.6 shows the results for no-load condition. The experimental results for loaded condition are presented in Fig. 5.7. In both the figures it is seen that once the ILC is put into action, the error reduces from cycle to cycle till it converges to small magnitude. ILC output control voltage keeps adjusting so as to reduce the tracking error. From these results, we conclude that the performance of the scheme is equally good both under no-load and loaded conditions. The error converges to approximately 1 μm after fifteen cycles. Fig. 5.6 (e) and Fig. 5.7 (e) show the plot of root mean square of the position tracking error for no-load and loaded conditions respectively. It is noted that the error in case of the loaded condition settles to lower value than unloaded condition. It is due to the added inertia to the slider which dampens the high frequency vibration of the moving stage. It is clear from the results that once the ILC learns the system parameters it completely takes over the control action.

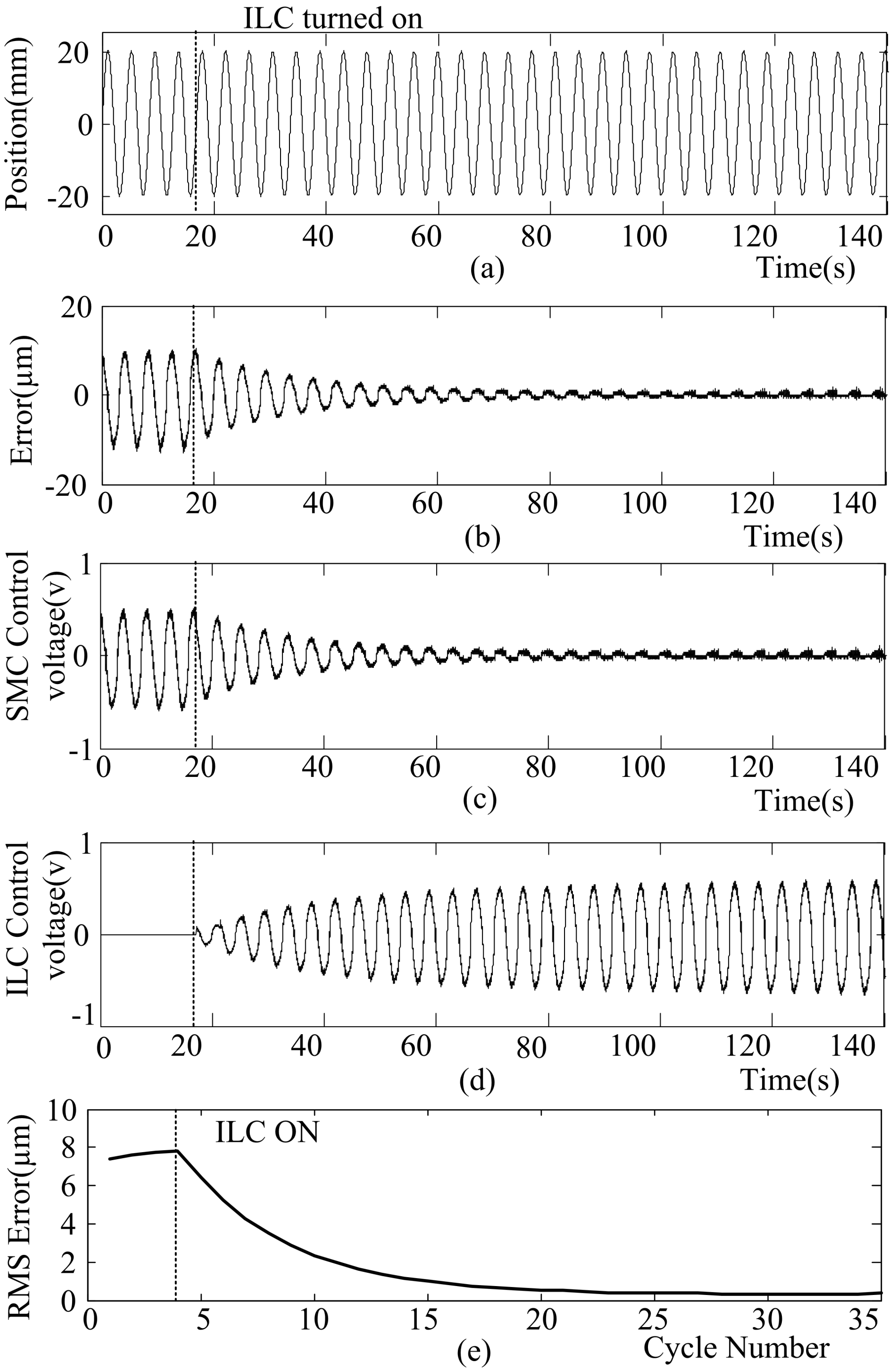


Figure 5.6: Position control of LUSM using SMC with ILC under no-load: (a) Reference and actual position (b) Tracking error (c) Control voltage from SMC (d) Control voltage from ILC (e) RMS Tracking error

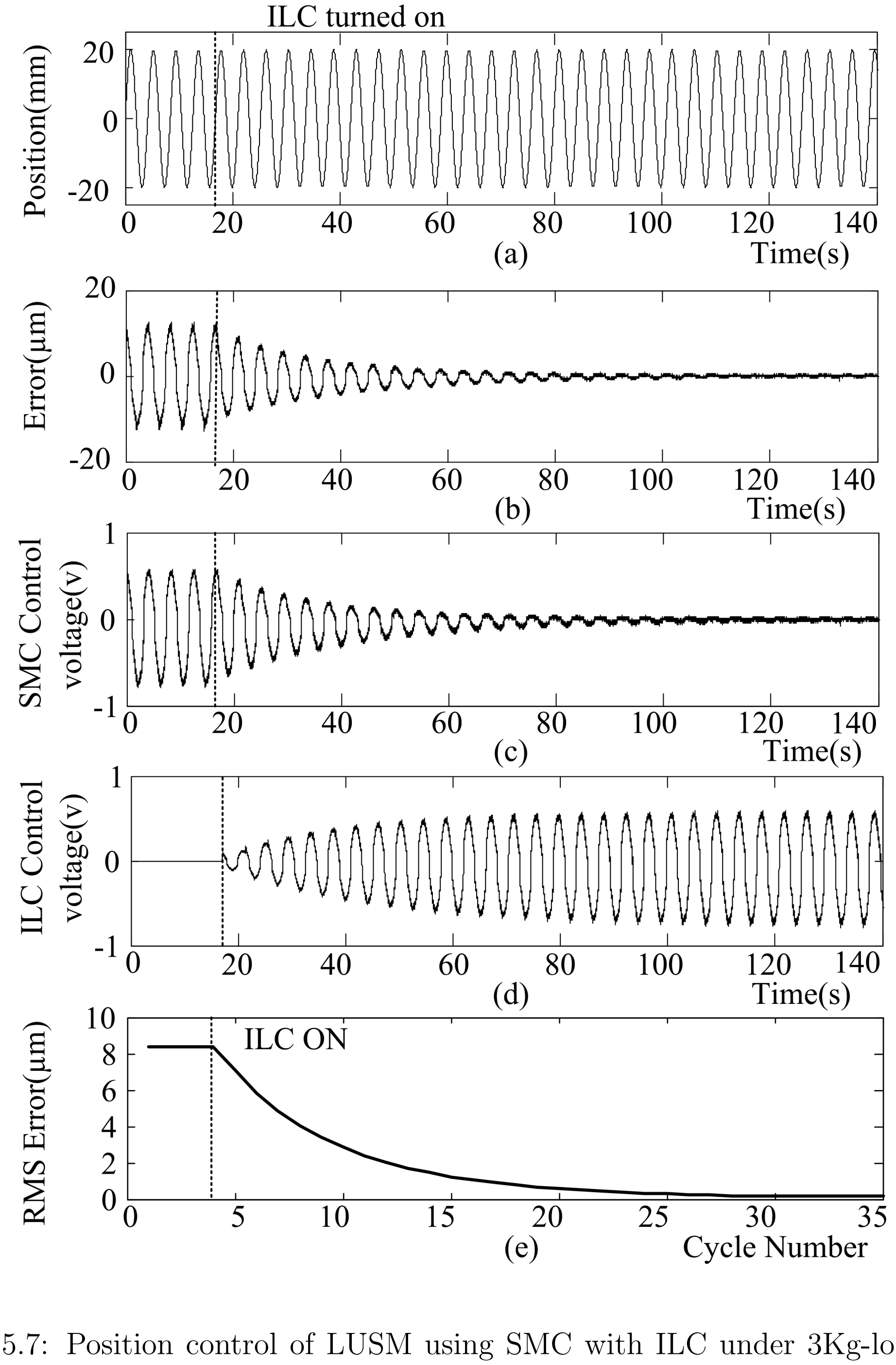


Figure 5.7: Position control of LUSM using SMC with ILC under 3Kg-load: (a) Reference and actual position (b) Tracking error (c) Control voltage from SMC (d) Control voltage from ILC (e) RMS Tracking error

Theoretically, the learning process can be kept on as long as the process is repeatable. Since the learning process of ILC is an integral process, keeping it on throughout will sum up high frequency measurement noise and it may cause instability. Using a low pass filter mitigates the problem to some extent. But it does not solve the long term stability problem because the practical filter will not have perfect cutoff at desired frequency [53]. So in practical learning scheme the learning process is frozen once no further improvement is observed. In these experiments, the learning is frozen after ten cycles once the tracking performance do not show any further improvement. It is also noticeable that though there is a large difference between the tracking errors before and after ILC is turned on, there is not much difference in the motor control voltage. The difference is that initially the control voltage is fully determined by feedback controller, while after ILC convergence the control voltage is almost fully determined by ILC.

5.4 Summary

In this chapter, an introduction to iterative learning control is presented. For periodic references, the periodic errors are minimized using an iterative learning controller in addition to the sliding mode controller. The experimental results obtained verify that the tracking performance can be improved by a factor of ten using the proposed scheme with much less computation overhead and ease of implementation. The performance of the proposed control scheme is equally good for the loaded conditions. ILC scheme is easy to be implemented in the existing drive system without any extra hardware modifications. The scheme requires an adequate size of memory and elementary algebraic computation.

Chapter 6

Position Control with Direct Learning Control

In Chapter 5 iterative learning control technique is discussed to improve the tracking performance for repetitive positioning applications. Iterative learning control is applicable for strictly repeatable tasks. This chapter discusses learning control scheme for non-repetitive position tracking applications. First, an introduction to direct learning control concept is given. Subsequently its formulation and application for position control of ultrasonic motor are discussed.

6.1 Introduction

Most of the industrial tasks are repetitive in nature such as scanning systems, radar tracking, metal cutting, industrial robots etc. For such repetitive tasks, the error resulting from unmodelled dynamics of the system can be suppressed by adding an iterative learning controller (ILC). In such iterative learning control scheme, little prior system and control knowledge are assumed. The control actions are learned gradually in iterations through trial and error fashion. Thus, it will take few cycles of the reference trajectory to correctly learn the control action. The iterative learning control will be effective if and only if the reference trajectory or disturbance in the system is strictly repetitive in nature. Any change in task specifications requires a fresh learning process to be started provided the new task

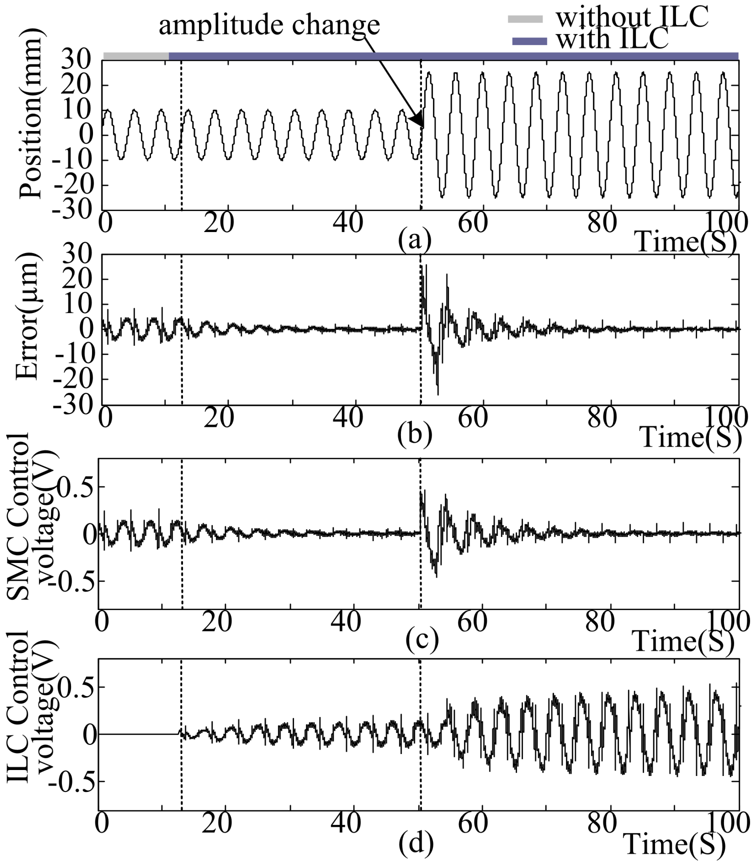


Figure 6.1: Position control of LUSM using SMC and ILC with variation in amplitude (a)Actual and reference position (b) Tracking error (c) SMC control voltage (d) ILC control voltage

is again repetitive in nature. Position tracking results for such task specifications are shown in Fig. 6.1 and Fig. 6.2. The iterative control we are considering is time indexed. The data memory required for it will change when the time scale of the trajectory is changed, keeping the sample time fixed. Fig. 6.1 shows the tracking results when reference position amplitude is changed from 10 mm to 25 mm. The tracking error is brought back to the initial level of 1 μm within few cycles after the change in reference. The information learned previously does not fit exactly for the new amplitude position reference. It takes few cycles of new reference to unlearn the control effort for the old trajectory and learn for the new one. When the frequency of the reference position is changed, the information stored will not be in appropriate phase and magnitude requiring the ILC to be switched off for the first

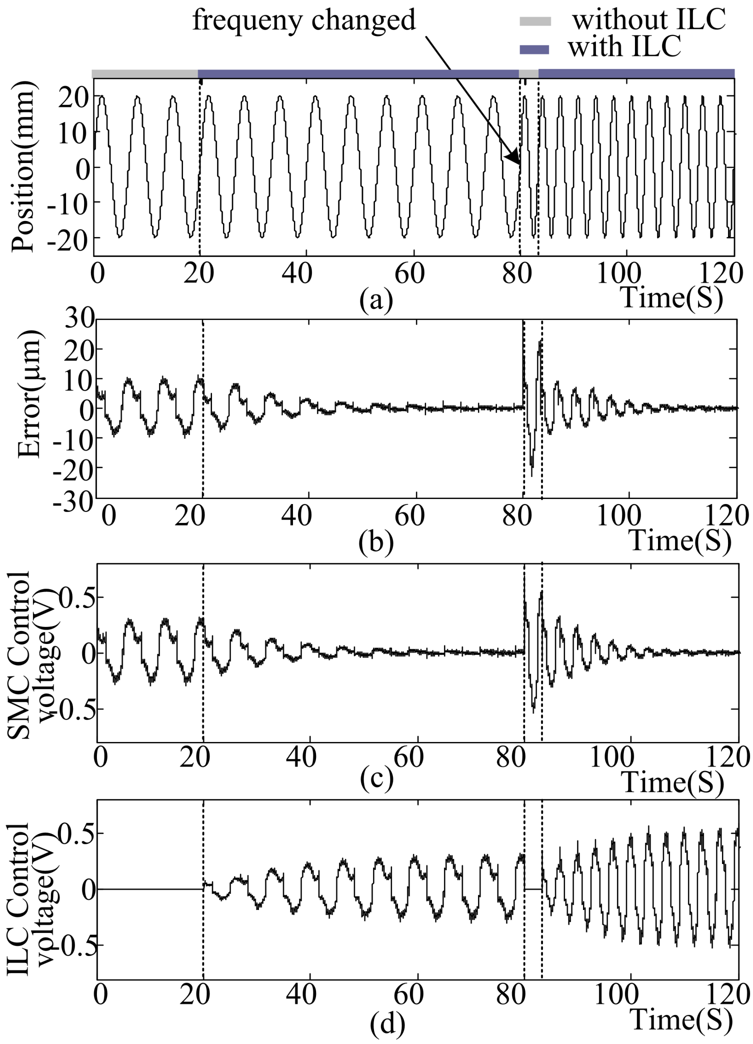


Figure 6.2: Position control of LUSM using SMC and ILC with variation in frequency (a)Actual and reference position (b) Tracking error (c) SMC control voltage (d) ILC control voltage

cycle of new reference position. Using the fixed sampling time, the memory size of ILC has to be accordingly adjusted when time scale of the reference trajectory is changed. During the first cycle of new reference position ILC history data is cleared, memory size is readjusted and new information is stored. From the successive cycle, ILC will provide the compensation. Fig. 6.2 shows results when frequency of sine reference is changed from $0.15\,Hz$ to $0.3\,Hz$. It is seen that during the transient part SMC will take over the control action while in steady state the ILC is more dominant. The tracking error reduced to same order as before the dynamic change in frequency once the ILC learned the new control pattern.

Iterative learning process cannot be used where reference trajectory change from cycle to cycle *i. e.* when the trajectory is non-periodic in nature. Many times, a system may have plenty of prior control knowledge acquired through all past actions corresponding to highly correlated different motion tasks. These control profiles for spatially identical motion patterns would be inherently correlated as they are generated by the same system dynamics. For a similar but new reference trajectory, a control profile can be generated directly from the knowledge of pre-learned control profiles. Such estimation of the desired control input profile from existing control input profiles without repeated learning is called *direct learning of control* (DLC). From the practical point of view, such non-repeatable learning control is important and indispensable as the reference trajectory may change due to changes in task specifications.

Direct estimation of control profile for a new reference trajectory from the control profiles for tracking similar references have been reported in [54]-[56]. In these papers, authors have shown through simulation results that directly learned control profile is close to the desired control profile. In direct estimation of control profile, the performance of the controlled system depends upon the number of control profiles used to estimate it and how they are spread in the reference trajectory space.

In this chapter, a hybrid DLC-ILC based control scheme is proposed to harness the advantages of iterative learning and direct learning control techniques. In most applications, the task specifications change once in a while. For such applications, where new reference trajectory is repeating for a considerable number of cycles of operation, direct learning controller is active only during the first cycle. Iterative learning controller becomes ineffective during this cycle due to change in reference trajectory. During this first cycle, iterative learning controller learns the control effort from direct learning controller and the resulting tracking error. From the successive cycles, the iterative learning controller is switched back into action and direct learning controller is deactivated. For such applications using direct learning controller throughout the cycles of the new reference trajectory will result in the replication of the tracking error resulting from the control signal estimation error in direct learning control. Using DLC-ILC scheme a better tracking performance and error convergence time compared to starting a fresh iterative learning process can be achieved. Only direct learning control will be active where reference trajectory change in every new cycle.

The proposed DLC-ILC hybrid scheme is applied for position tracking of the linear ultrasonic motor. Sliding mode position controller discussed in Chapeter 3 is used as feedback controller. DLC-ILC controller is added to improve the tracking performance. The DLC-ILC concept can be used for spatially same but of different amplitude and time scales reference position trajectories. In this research work, we have limited our study to the spatially same but different time scale trajectories, where the knowledge of the time scales are known a *priori*.

6.2 Principle of Direct Learning Control

Consider a nonlinear dynamic system described as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t)) + B\mathbf{u}(t) \\ \mathbf{y}(t) &= C\mathbf{x}(t) \end{aligned} \tag{6.1}$$

where $\mathbf{u}(t)$ is the control input vector, $\mathbf{y}(t)$ is the output vector, $f(\mathbf{x}(t)) \,\epsilon\, R^n$, $B \,\epsilon\, R^n$ is the input matrix, $\mathbf{x}(t) \,\epsilon\, R^{n\times 1}$ is the state vector of the system and $C \,\epsilon\, R^{1\times n}$ is a constant matrix. $f(\mathbf{x}(t))$ is unknown function vector, B and C are unknown matrices. We assume that matrix CB is of full rank.

Consider a period of the output trajectory as $\mathbf{y}_j(t_j) = [y_{j1}(0)\ y_{j2}(1)\ y_{j2}(2)\ ...$ $..\ y_{jn}(T_j)]^T, t_j \,\epsilon\, [0, T_j]$. Two trajectories $\mathbf{y}_i(t_i)$ and $\mathbf{y}_j(t_j)$ are said to have identical spatial path associated with different time scales if the following condition is strictly satisfied.

- $\forall\, t_i \,\epsilon\, [0, T_i]$ there exists an unique $t_j \,\epsilon\, [0, T_j]$ and $t_i \neq t_j$ such that
$$\mathbf{y}_i(t_i) = \mathbf{y}_j(t_j).$$

The trajectory $\mathbf{y}_j(t_j)$ is related to the another trajectory $\mathbf{y}_i(t_i)$ through a continuously differentiable mapping function $\rho(t_j)$ i.e. $\mathbf{y}_j(t_j) = \mathbf{y}_i(\rho(t_j))$. The mapping function $\rho(t_j)$ satisfies $\rho(0) = 0$ and $T_i = \rho(T_j)$. It is to note that $\forall\, t_j \,\epsilon\, [0, T_j]\ \ \exists\, t_i \,\epsilon\, [0, T_i]$ such that $\mathbf{q} = \mathbf{y}_i(t_i) = \mathbf{y}_i(\rho(t_j)) = \mathbf{y}_j(t_j)$, which implies that the spatial path described by $\mathbf{q}$ is same for both trajectories $\mathbf{y}_i(t_i)$ and $\mathbf{y}_j(t_j)$ regardless of their speed patterns. For direct learning of control, there must be information of at least two $(l > 2)$ such spatially same trajectories. These trajectories should be at least continuously differentiable once with time. All the stored trajectories are related with each other through a set of continuously differentiable mapping function $\rho_i(t)$ such that

$$\begin{aligned} & t_i = \rho_i(t) \\ & \mathbf{y}_1(t_1) = \mathbf{y}_2(t_2) =, ..., = \mathbf{y}_l(t_l); \qquad t \,\epsilon\, [0, T], t_i \,\epsilon\, [0, T_i]. \end{aligned} \tag{6.2}$$

Differentiating the output equation of (6.1) with respect to time.

$$\frac{d\mathbf{y}(t)}{dt} = C\frac{d\mathbf{x}(t)}{dt} = C(f(\mathbf{x}(t)) + B\mathbf{u}(t)) \tag{6.3}$$

$$\mathbf{u}(t) = [CB]^{-1}\left[\frac{d\mathbf{y}(t)}{dt} - Cf(\mathbf{x}(t))\right] \tag{6.4}$$

The desired control input signal with respect to new trajectory $\mathbf{y}_d(t_d)$, $t_d \,\epsilon\, [0, T_d]$ can be expressed as

$$\mathbf{u}_d(t_d) = [CB]^{-1}\left[\frac{d\mathbf{y}_d(t_d)}{dt_d} - Cf(\mathbf{x}_d(t_d))\right] \tag{6.5}$$

We cannot find $\mathbf{u}_d(t_d)$ directly using this formula since $f(\mathbf{x}(t))$, B and C are unknown. Thus we consider the output trajectories $\mathbf{y}_i(t_i)$ and their corresponding control profiles $\mathbf{u}_i(t_i)$ that has been achieved a *priori*.

$$\mathbf{u}_i(t_i) = [CB]^{-1}\left[\frac{d\mathbf{y}_i(t_i)}{dt_i} - Cf(\mathbf{x}_i(t_i))\right] \tag{6.6}$$

Differentiating $\mathbf{y}_d(t_d)$ with respect to t_d we get

$$\begin{aligned} \frac{d\mathbf{y}_d(t_d)}{dt_d} = \frac{d\mathbf{y}_i(t_i)}{dt_i} \times \frac{d(t_i)}{dt_d} &= \frac{d\mathbf{y}_i(t_i)}{dt_i} \times \frac{d(\rho_i(t_d))}{dt_d} \\ \frac{d\mathbf{y}_i(t_i)}{dt_i} &= \left(\frac{d(\rho_i(t_d))}{dt_d}\right)^{-1}\frac{d\mathbf{y}_d(t_d)}{dt_d} \end{aligned} \tag{6.7}$$

Since $\mathbf{y}_d(t_d)$ and $\mathbf{y}_i(t_i)$ all describe the same spatial path, we have $\mathbf{y}_d(t_d) = \mathbf{y}_i(t_i)$. Since C being a constant matrix, $\mathbf{y}_d(t_d) = \mathbf{y}_i(t_i)$ implies that $\mathbf{x}_d(t_d) = \mathbf{x}_i(t_i)$. Thus the pre-stored control profiles is given by

$$\mathbf{u}_i(t_i) = [CB]^{-1}\left[\left(\frac{d(\rho_i(t_d))}{dt_d}\right)^{-1}\frac{d\mathbf{y}_d(t_d)}{dt_d} - Cf(\mathbf{x}_d(t_d))\right] \tag{6.8}$$

Putting the pre-stored control profiles in the matrix form

$$\begin{bmatrix} \mathbf{u}_1(\rho_1(t_d)) \\ \mathbf{u}_2(\rho_2(t_d)) \\ \vdots \\ \mathbf{u}_l(\rho_l(t_d)) \end{bmatrix} = \begin{bmatrix} \left(\frac{d\rho_1(t_d)}{dt_d}\right)^{-1} I & I \\ \left(\frac{d\rho_2(t_d)}{dt_d}\right)^{-1} I & I \\ \vdots & \vdots \\ \left(\frac{d\rho_l(t_d)}{dt_d}\right)^{-1} I & I \end{bmatrix} \begin{bmatrix} [CB]^{-1}\frac{d\mathbf{y}_d(t_d)}{dt_d} \\ -[CB]^{-1}Cf(\mathbf{x}_d(t_d)) \end{bmatrix} \tag{6.9}$$

Defining the regressor matrix A_l and input vector $\bar{\mathbf{u}}_l$ as

$$A_l = \begin{bmatrix} \left(\frac{d\rho_1(t_d)}{dt_d}\right)^{-1} I & I \\ \left(\frac{d\rho_2(t_d)}{dt_d}\right)^{-1} I & I \\ \vdots & \vdots \\ \left(\frac{d\rho_l(t_d)}{dt_d}\right)^{-1} I & I \end{bmatrix} \quad , \quad \bar{\mathbf{u}}_l = \begin{bmatrix} \mathbf{u}_1(\rho_1(t_d)) \\ \mathbf{u}_2(\rho_2(t_d)) \\ \vdots \\ \mathbf{u}_l(\rho_l(t_d)) \end{bmatrix} \tag{6.10}$$

From equation (6.9)

$$\bar{\mathbf{u}}_l = A_l \begin{bmatrix} [CB]^{-1}\frac{d\mathbf{y}_d(t_d)}{dt_d} \\ -[CB]^{-1}Cf(\mathbf{x}_d(t_d)) \end{bmatrix} \tag{6.11}$$

From expression (6.5) we get

$$\mathbf{u}_d(t_d) = [I \quad I] \begin{bmatrix} [CB]^{-1}\frac{d\mathbf{y}_d(t_d)}{dt_d} \\ -[CB]^{-1}Cf(\mathbf{x}_d(t_d)) \end{bmatrix} \tag{6.12}$$

As $A_l^T A_l$ is invertible, multiplying both sides of (6.11) with $(A_l^T A_l)^{-1} A_l^T$ we obtain the direct learning control as

$$\mathbf{u}_d(t_d) = [I \quad I](A_l^T A_l)^{-1} A_l^T \bar{\mathbf{u}}_l \tag{6.13}$$

It is seen from expression (6.13) the direct learning control profile is the estimation from the previous control profiles with appropriate weights given to each of them.

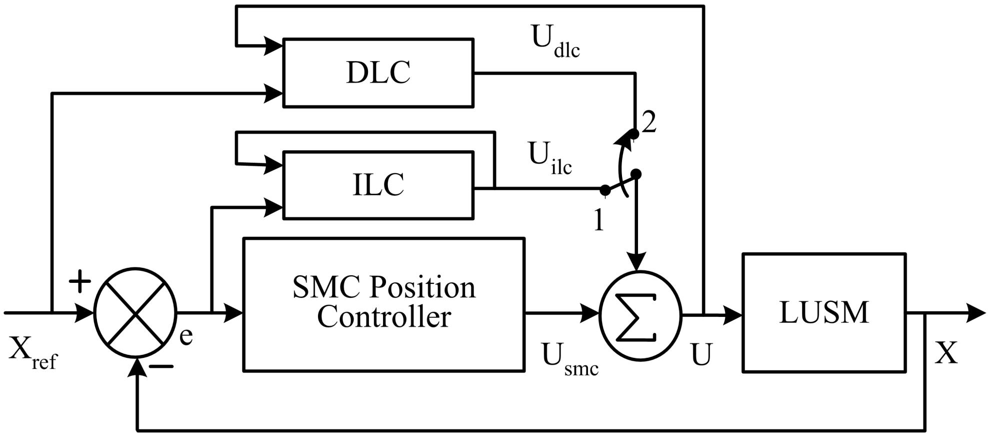
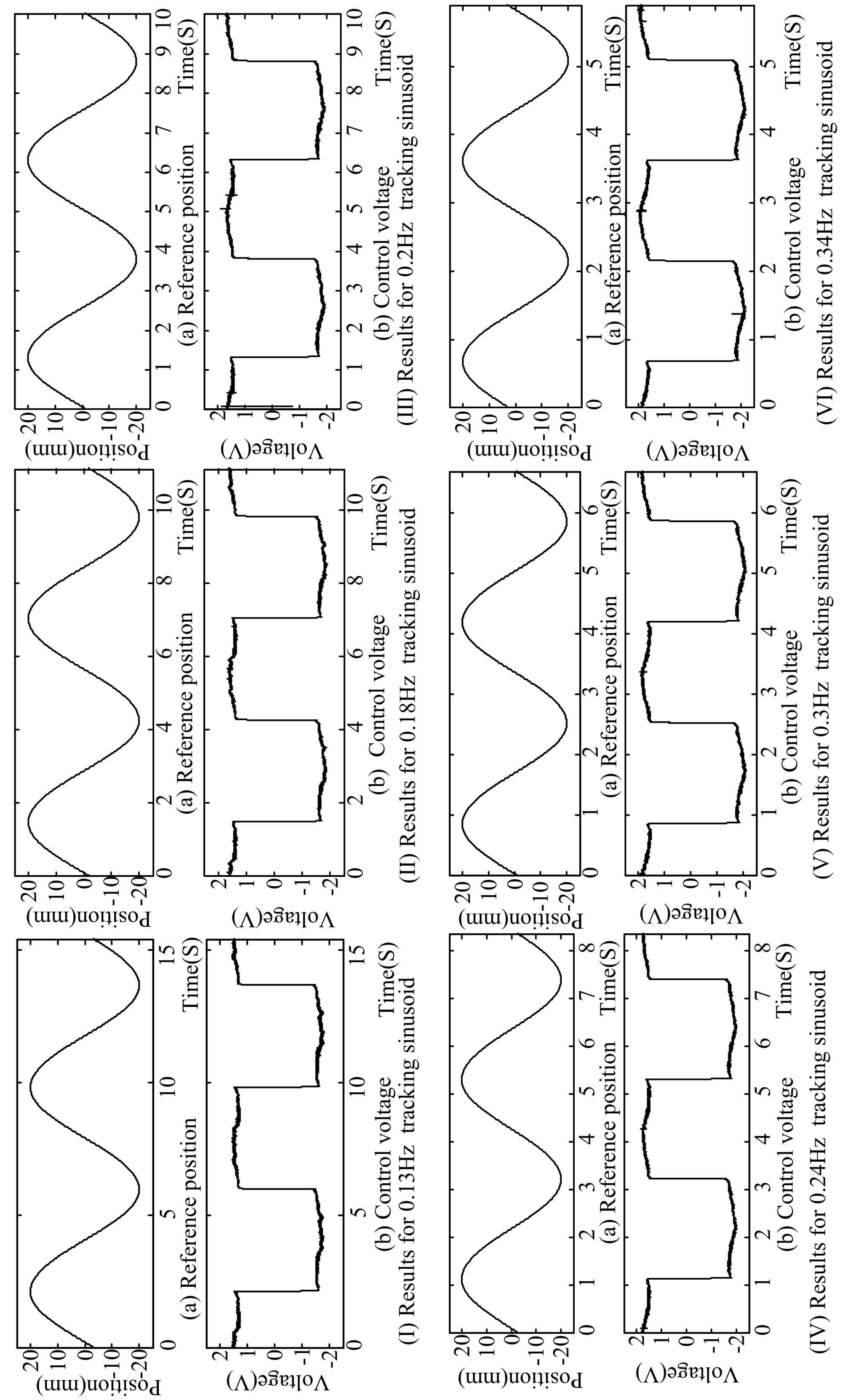


Figure 6.3: Linear Ultrasonic Motor DLC-ILC Position Control Scheme

6.3 Direct Learning Control for Position Control of LUSM

The principle of direct learning control discussed above is used for the position control of the ultrasonic motor. It is used along with sliding mode controller and iterative learning controller schemes to harness the advantages of each scheme. The control scheme is shown in Fig. 6.3.

The performance of DLC scheme depends upon the accuracy of the previous control profiles that are learned and the region they cover in the reference trajectory space. More accurate the control profiles and wider the region the control profiles cover the reference trajectory space, the better the performance will be. Six different control profiles corresponding to six different frequencies ($0.13\,Hz$,$0.18\,Hz$,$0.2\,Hz$,$0.24\,Hz$,$0.3\,Hz$, $0.34\,Hz$) sine reference positions are learned for experiment purpose. Initially sinusoidal position references of the frequency mentioned above are given to the motor to track. During this phase, the control switch is in position 1. The iterative learning control is used to generate the control profile for each trajectory. Once the error converged to small bound, the control effort is learned by DLC. The reference positions and the learned control profiles are shown in Fig. 6.4. After finishing the initial learning process, for any change in time scale of the reference trajectory, the control switch is moved to position 2. The control effort is directly estimated by DLC from the previously learned and stored control profiles. The position tracking results for a reference trajectory where the time scale is changing from cycle to cycle is shown in Fig. 6.5. Initially the motor is tracking a $0.34\,Hz$ signal, it is then followed by frequencies of $0.08\,Hz$, $0.26\,Hz$ and $0.4\,Hz$ respectively. For such reference tracking, once the learning is over only DLC is active. It is clear that in such non-periodic tracking applications ILC cannot be applied and using DLC the tracking error is maintained at lower level. The performance of the DLC depends upon how accurately it estimates the control profile from historical knowledge of the stored control profiles. The comparison of control voltages is shown in Fig. 6.6. The expected control voltages are obtained from iterative learning controller for the particular frequency trajectory where as the estimated control profiles are from the DLC scheme. It is seen that the estimated and expected control voltages are close to each other. For clarity, a zoomed view of a portion of the comparison of voltages for $0.08\,Hz$ is shown in Fig. 6.7. There is slight deviation of the estimated voltage from the expected voltage. This estimation error in control voltage results in tracking error but it is much lower than if no direct learning control is used. In some of the applications, the change in task specifications will not be so frequent. In such cases, the reference trajectory will repeat for considerable number of cycles. The tracking



(a) Reference position

(b) Control voltage

(I) Results for 0.13Hz tracking sinusoid

(a) Reference position

(b) Control voltage

(II) Results for 0.18Hz tracking sinusoid

(a) Reference position

(b) Control voltage

(III) Results for 0.2Hz tracking sinusoid

(a) Reference position

(b) Control voltage

(IV) Results for 0.24Hz tracking sinusoid

(a) Reference position

(b) Control voltage

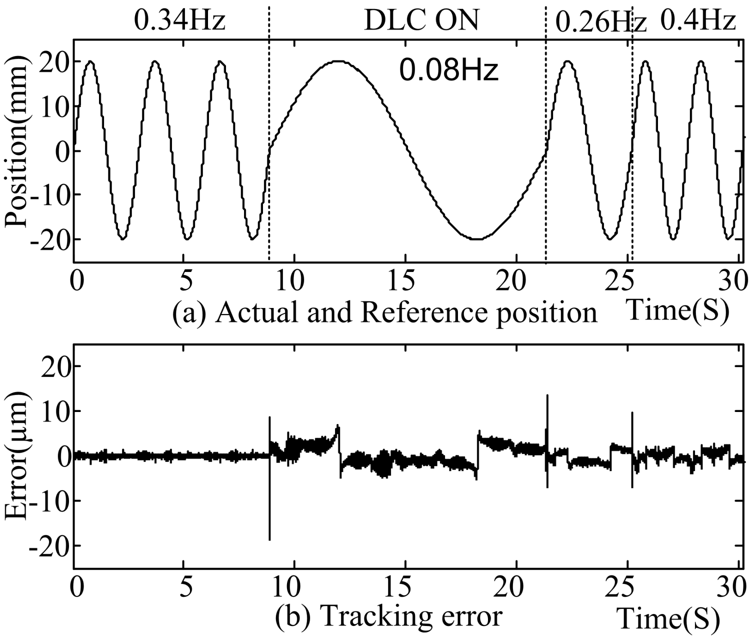
(V) Results for 0.3Hz tracking sinusoid

(a) Reference position

(b) Control voltage

(VI) Results for 0.34Hz tracking sinusoid

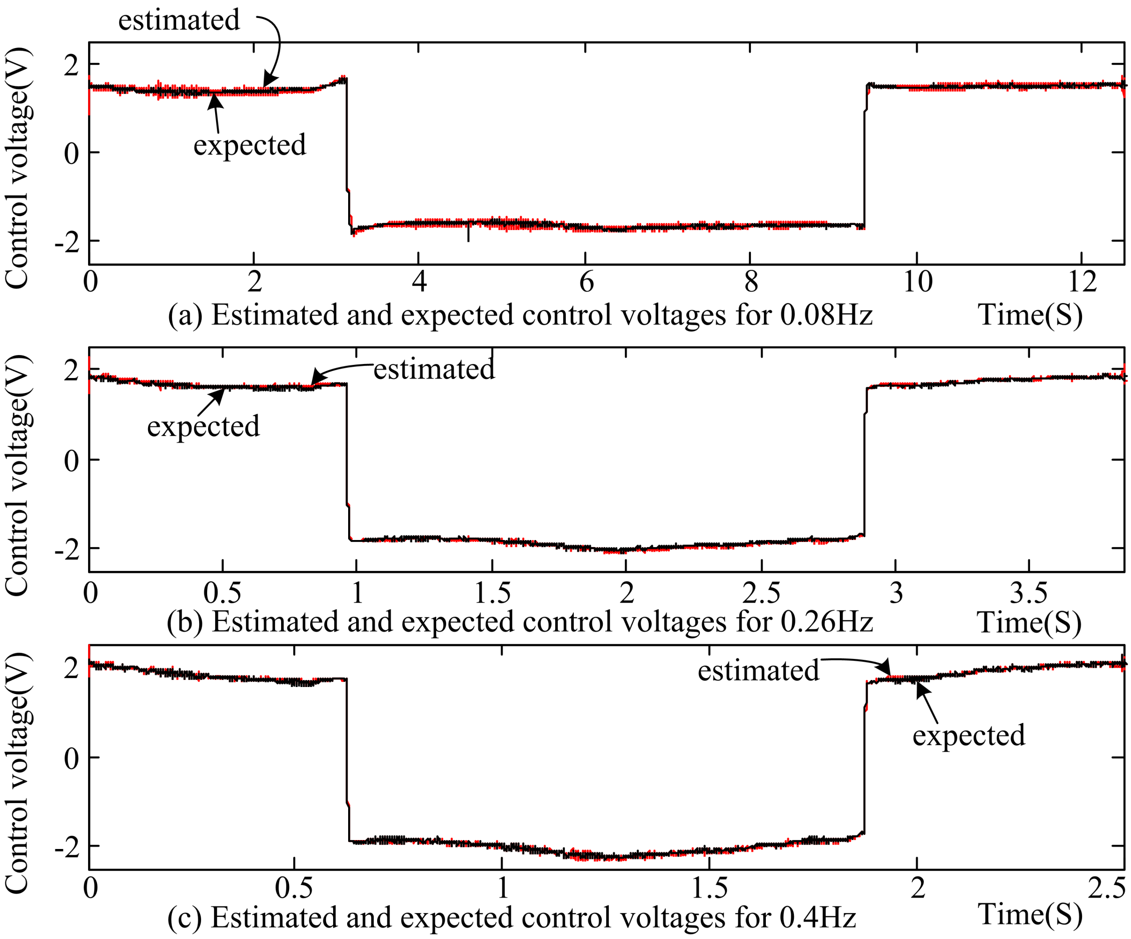
Figure 6.4: Trajectories and their control profiles learned by DLC



(a) Actual and Reference position

(b) Tracking error

Figure 6.5: Position tracking performance with DLC



(a) Estimated and expected control voltages for 0.08Hz

(b) Estimated and expected control voltages for 0.26Hz

(c) Estimated and expected control voltages for 0.4Hz

Figure 6.6: Comparison of the estimated and expected control voltages

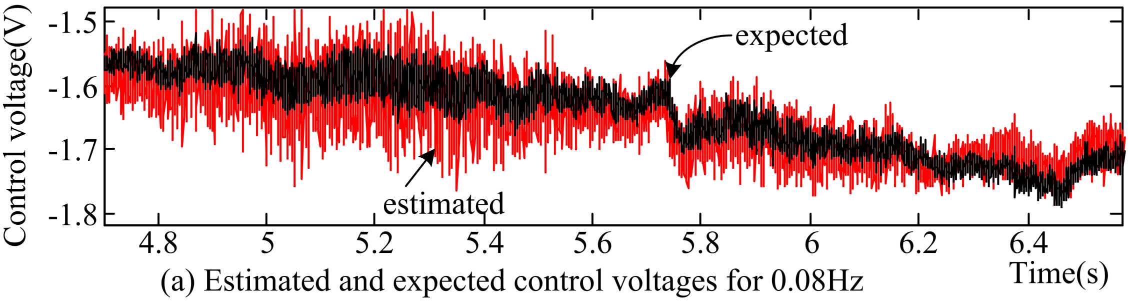


Figure 6.7: Zoomed view of comparison of the estimated and expected control voltages

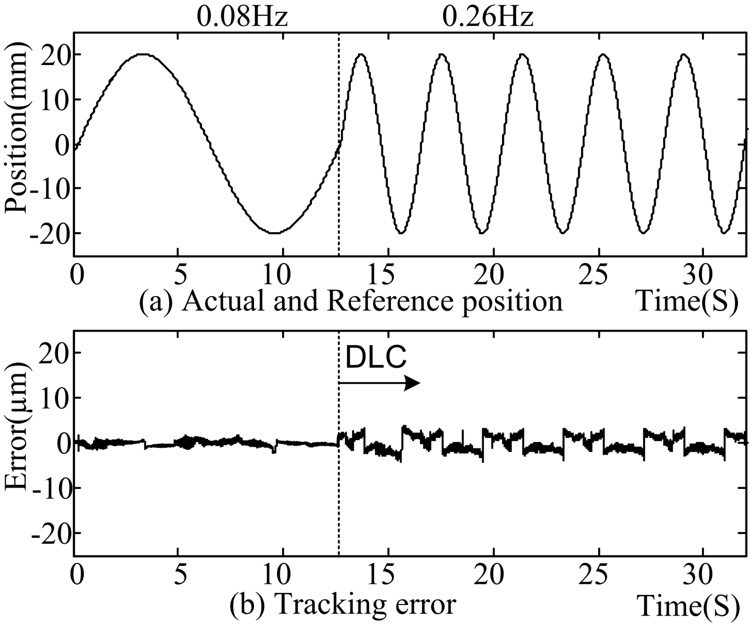


Figure 6.8: Position tracking performance of DLC scheme

error resulting due to the estimation of the control effort will keep repeating. Thus using DLC alone for new repeatable pattern will result in larger steady state error. Fig. 6.8 shows the position tracking performance when a new repeatable reference of 0.26 Hz is given. When the frequency is changed to 0.26 Hz, the DLC is switched in throughout all the cycles of new reference trajectory. Though the error in kept at lower level compared without DLC, the same error keeps repeating. For such situations, the repeating error can be further reduced by bringing in iterative learning controller. During the first cycle when DLC is active providing the control action, the iterative learning control is learning the control profile from the DLC

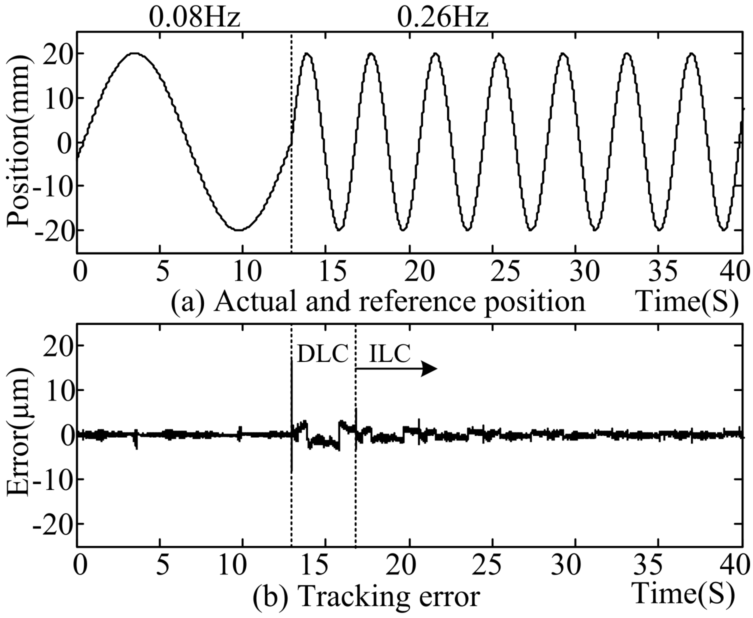


Figure 6.9: Position tracking performance of DLC-ILC scheme

and the resulting error. After the first cycle, the control switch is moved back to position 1 and the compensation is provided from iterative learning controller. Thus the DLC-ILC control law is as stated in equation (6.14).

$$V_c(r) = \begin{cases} U_{smc}(r) + U_{ff}(r) + U_{ilc}(r,k) & if\ k > 1 \\ U_{smc}(r) + U_{ff}(r) + U_{dlc}(r,k) & if\ k = 1 \end{cases} \tag{6.14}$$

where k is iteration number of the new position reference trajectory.

The tracking results for proposed DLC-ILC scheme is shown in Fig. 6.9. The tracking error during the period when DLC is on is less than 5 μm and the error converges to 1 μm within next two cycles. The frequency 0.26 Hz is within the learned frequencies range. The tracking performances for frequencies outside the learned frequencies range are shown in Fig. 6.10 and Fig. 6.11. Fig. 6.10 shows the tracking performance when frequency of reference is changed from 0.34 Hz to 0.08 Hz. The tracking results for change in frequency from 0.26 Hz to 0.4 Hz is shown in Fig. 6.11. These results show that DLC has helped to keep the tracking error at lower level and also reduce the learning time of ILC. Thus using the hybrid DLC-ILC scheme shown in Fig. 6.3, a better tracking performance is obtained

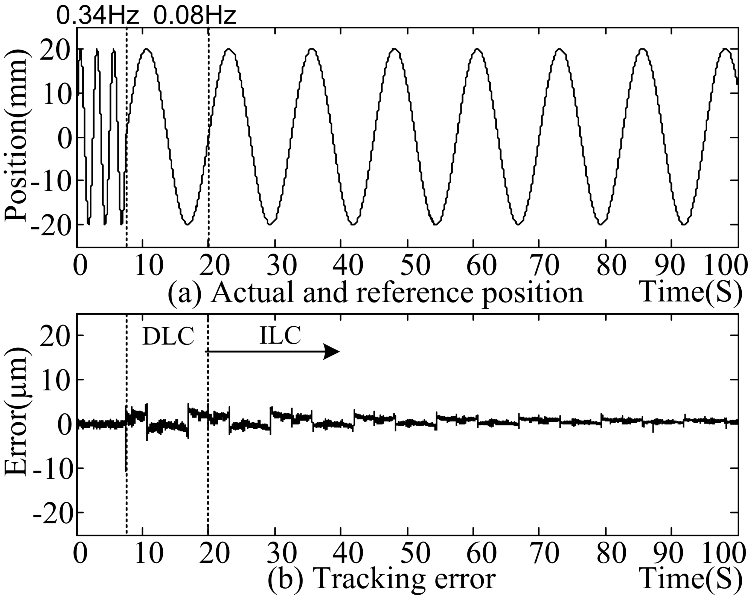


Figure 6.10: Position tracking with DLC-ILC scheme for new frequency of 0.08Hz

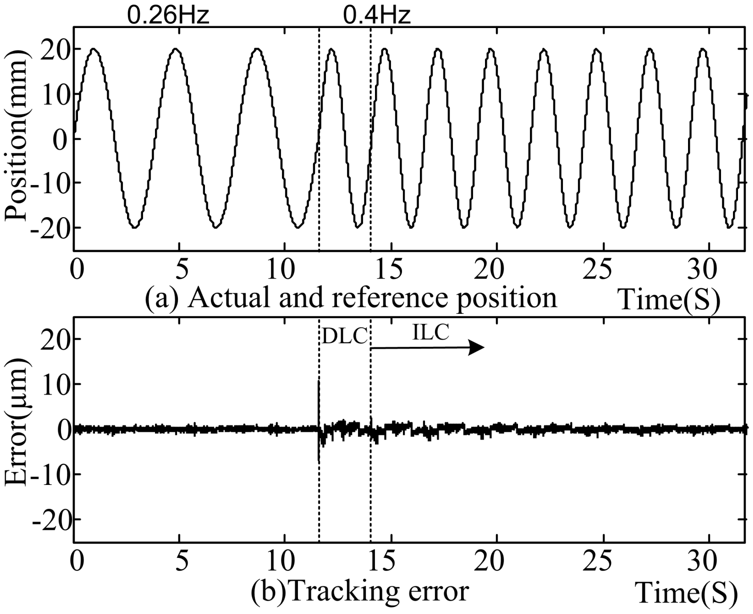


Figure 6.11: Position tracking with DLC-ILC scheme for new frequency of 0.4Hz

whether the new reference position is repetitive or not. A much better performance can be achieved if a numerous wide range of patterns are learned for estimating the control effort.

6.4 Summary

For repetitive position tracking, iterative learning control is used to improve the performance. The application of the iterative learning control is limited to strictly repeatable tasks. For changes in reference trajectory and if the new reference is repeatable then a fresh learning process has to be initiated, which is not desirable as initiating a fresh learning process takes long learning time. Using the proposed DLC-ILC scheme the number of learning cycles are reduced keeping the error at much lower boundary. For non-repeatable positioning tasks DLC scheme helps to get good positioning performance. The experimental results obtained validates the usefulness of the scheme. The position tracking performance can be further improved if wide range of trajectories are learned. The computation requirement is also less as all the matrix operations are done off-line. The experimental results obtained show that using direct learning control, the error is limited to one third of the value than without it.

Chapter 7

Extended Summary and Conclusions

7.1 Conclusions

In this thesis, the position control of the linear ultrasonic motor HR8 has been presented in a lucid way. In short, different types of electromagnetic positioning actuators and the need for ultrasonic motor are discussed in Chapter 1. Despite having simple construction, it is very difficult to develop a detailed mathematical model for the motor. It is due to the fact that the non-linear phenomenon of friction and inverse piezoelectric effects have to be addressed. In ultrasonic motors, the friction serves as the medium of force transfer from the piezoelectric stator to rotor/slider and at the same time it also contributes to the non-linearity of the motor. For ultra high precision motion control applications, a good model for force generation from piezoelectric ceramic and friction is required. Detailed discussions on literature reported on the position control of the ultrasonic motors have been presented. Most of the previous works have been focussed on design of the piezoelectric actuators. There are numerous works reported for control of ring type of ultrasonic motors. For position control of linear ultrasonic motor very few works have been reported in literature. Most of the reported literature are focussed on intelligent control techniques based on neural networks and fuzzy logic for control of linear ultrasonic motors. The control performance achieved is not enticing though the sophisticated control algorithms with large implementation overhead are used.

The operating principles of the HR8 motor has been discussed. The motor does not move till the control voltage exceeds a certain minimum value. This is due to static friction between the stator and rotor. This deadzone in motor characteristics is compensated by providing a feedfoward signal in motor control voltage depending on the reference velocity and position error profile. First, a simple linear proportional-intergral (PI) controller is used for the position control of the motor. A repeatable performance with resolution of $100nm$ has been achieved for set point position tracking. The performance of linear PI controller is not satisfactory when tracking time varying reference trajectory and under load variations. A linearized second order model of the motor is used to design a sliding mode position controller. The performance of the variable structure controller is better than linear PI controller for tracking time varying position reference trajectory and under load variations. A modified sliding mode controller with a small boundary layer is introduced to reduce chattering.

The conventional sliding mode controller suffers from intense chattering, which may not be desirable for practical applications. The chattering will result in wear and tear in motor. A high switching gain is desirable to ensure global attractiveness of the switching surface. Once in sliding mode, it is necessary to have equivalent control component to ensure good tracking performance. Due to the presence of system modelling error and existence of disturbances, it is not possible to directly acquire the equivalent control. Thus filtering technique is used to extract the equivalent control. A low pass filter is used to extract equivalent control from the switching control signal assuming that the bandwidth of equivalent control is within the bandwidth of the filter. Once in sliding phase, keeping high gain will result in chattering. A second low pass filter is used to scale down the switching gain. The two filters are activated simultaneously and they work concurrently when the system enters the sliding phase. It is shown that this scheme can be used to estimate the disturbance in the system and provide compensation for it. The parameters of the filters are such that once the operating point is knocked away from the sliding surface due to some unexpected disturbance, the switching gain will surge up and the motor will re-enter sliding surface in finite time.

For high-precision positioning performance, the detail knowledge of the system is a prerequisite. In the absence of good system knowledge and when system is performing a repetitive task, the performance of the system can be improved using an iterative learning process. The knowledge already available is made use of to design the controller and further system knowledge is acquired when the system is running using iterative learning control. In iterative learning control (ILC), a feed-forward control signal is determined in an iterative manner to achieve the desired result by compensating the periodic errors. The error measured during the current cycle is converted into an improved feed-forward signal for the next cycle. During the next cycle the computed feed-forward signal is added to the control signal provided by the feedback controller. At the same time, the feed-forward control signal for next run is updated with the current error. This process is continued until the input signal is modified to the point when applied to the system it produces the desired output. This is similar to the conventional feedback controller which updates the next time step control signal using the feedback at current or past time steps. The updates of control voltage in iterative learning control are performed using feedback from previous repetitions. Using iterative learning control for repetitive position tracking of the ultrasonic motor, the tracking error is improved significantly. The performance has been improved at least by a factor of ten compared to the scheme without the feed-forward learning control compensation signal.

The iterative learning control is effective in improving the tracking performance only if the trajectory is strictly repetitive. Any variation of task specifications requires the learning process to be resumed from the very beginning. The previously learned control knowledge can no longer be used to update the control signal. A system may have plenty of prior control knowledge obtained through all past actions corresponding to different but highly correlated tasks. The knowledge of such prior control profiles can be effectively utilized for non-repeatable but correlated tasks. The direct generation of the feed-forward control signal from the knowledge of control profiles for tracking spatially identical pattern, termed as direct learning control (DLC), is used for tracking spatially identical but different time scales reference position trajectory tracking for the ultrasonic motor. Such control technique is effective to limit the tracking error to a lower level as conventional iterative learning process cannot be used. In case, if the new reference trajectory is repetitive, iterative learning process in coordination with direct learning control can be used. During the first cycle of a new reference trajectory, control profile from the direct learning controller is used and for successive cycles iterative learning control provides the feed-forward compensation signal. Such a hybrid scheme reduces the initial cycles tracking error and learning convergence time. Experiment results showed the efficacy of this scheme.

7.2 Future Work

In this thesis, simple yet effective position controllers are proposed and the performances are verified by the experimental validations. While trying to keep the analysis and design simple, an in sight into the piezoelectric phenomenon has been overlooked and a linear model has been assumed for it. A future work delving more into the modelling of inverse piezoelectric phenomenon may help to design a better mathematical model for the motor. The model thus developed may be used for simulation purposes to have an idea about the performance of the actual motor.

The Nanomotion motors are also used in vertical mounting orientation. Though the intrinsic friction of the motor creates a significant holding/braking advantage in a vertical orientation, the force of gravity on the mass must be taken into account. It is necessary to include a counterbalance for the force of gravity in the upward direction and inertial force for stopping in the downward direction. Thus different control parameters will likely be required in each direction. Thus this is an interesting controller design problem and have plethora industrial applications.

Bibliography

[1] T. J. E. Miller, *Brushless Permanent-Magnet and Reluctance Motor Drives.* Clarendon Press, Oxford, 1989.

[2] A. Hughes, *Electric Motors and Drives: Fundamentals, Types and Applications.* Butterworth-Heinemann, 2^{nd} ed., 1993.

[3] T. Kenjo, *Electric Motors and their Controls: An Introduction.* Oxford University Press, 1991.

[4] A. Landt, *Lenses for 35mm Photography.* the KODAK Workshop Series, Silver Pixel Press, 1998.

[5] D. G. Baum, "Piezoelectric motors and their implementations," *Nanomotion Ltd.*

[6] T. Sashida and T. Kenjo, *An Introduction to Ultrasonic Motors.* Clarendon Press, Oxford, 1993.

[7] K. Uchino, "Piezoelectric ultrasonic motors: overview," *Smart Mater. Struct*, no. 7, pp. 273–285, 1998.

[8] K. Uchino, "Piezoelectric actuators/ultrasonic motors: Their development and markets," *Proceeding of IEEE International Symopsium on Applications of Ferroelectrics*, pp. 319–324, August 1994.

[9] S. Ueha and Y. Tomikawa, *Utrasonic Motors: Theory and Applications.* Clarendon Press, Oxford, 1993.

[10] D. Arnone, A. V. Pelt, and K. L. Dessau, "Piezoelectric motors : Control set-and-hold semiconductor applications," *Photonics Spectra*, pp. 70–74, February 2003.

[11] J. Mass, T. Schulte, and N. Fröhleke, "Model-based control for ultrasonic motors," *IEEE/ASME Trans. on Mechatronics*, vol. 5, pp. 165–180, June 2000.

[12] N. E. Ghouti, *Hybrid Modelling of a Travelling Wave Piezoelectric Motor.* Phd thesis, Aalborg University, 2000.

[13] F. J. Lin and L. C. Kuo, "Driving circuit for ultrasonic motor servo drive with variable-structure adaptive model-following control," *IEE Proc. Electrical Power Application*, vol. 144, pp. 199–206, May 1997.

[14] Y. Izuno, R. Takeda, and M. Nakaoka, "New fuzzy reasoning-based high-performance speed/position servo control schemes incorporating ultrasonic motor," *IEEE/ASME Trans. on Mechatronics*, vol. 28, pp. 613–618, June 1992.

[15] T. Senjyu, H. Miyazato, S. Yokoda, and K. Uezato, "Speed control of ultrasonic motors using neural network," *IEEE/ASME Trans. on Power Electronics*, vol. 13, pp. 381–387, May 1998.

[16] T. Senjyu, K. Uezato, and H. Miyazato, "Adjustable speed control of ultrasonic motors by adaptive control," *IEEE Trans. on Power Electronics*, vol. 10, pp. 532–538, September 1995.

[17] T. Senjyu, H. Miyazato, and K. Uezato, "Quick and precise position control of an ultrasonic motor with dual mode control," *Proceedings of 1995 Interna-*

tional Conference on Power Electronics and Drive Systems, vol. 2, pp. 605–609, February 1995.

[18] T. Senjyu, H. Miyazato, and K. Uezato, “Quick and precise position control of ultrasonic motors taking account of frictional force control,” *IEEE Conference on Industrial Applications : Thirtieth IAS Annual Meeting*, vol. 1, pp. 85–89, October 1995.

[19] T. Senjyu, S. Yokoda, and K. Uezato, “Position control of ultrasonic motors using sliding mode control with multiple control inputs,” *Applied Power Electronics Conference and Exposition APEC’98*, vol. 2, pp. 597–602, February 1998.

[20] T. Senjyu, H. Miyazato, and K. Uezato, “Precise speed control of ultrasonic motors with repetitive control,” *International IEEE/IAS Conference on Industrial Automation and Control: Emerging Technologies*, pp. 165–169, May 1995.

[21] X. Wang, S. K. Panda, C. L. Teo, and C. J. Ong, “Speed control of cylindrical ultrasonic motor,” *IEEE 28th Annual Conference of the Industrial Electronics Society*, vol. 4, pp. 2739 – 2744, November 2002.

[22] T. Senjyu, T. Yoshida, K. Uezato, and T. Funabashi, “Position control of ultrasonic motors using adaptive backstepping control and dead-zone compensation with fuzzy inference,” *IEEE International Conference on Industrial Technology*, vol. 1, pp. 560–565, December 2002.

[23] T. Senjyu, S. Yokoda, Y. Gushiken, and K. Uezato, “Position control of ultrasonic motors with adaptive dead-zone compensation,” *IEEE Industry Applications Conference : Thirty-third IAS Annual Meeting*, vol. 1, pp. 506–512, October 1998.

[24] C.-Y. Yen, F.-L. Wen, and M. Ouyang, "Thin-disc piezoceramic ultrasonic motor. part ii: system construction and control," *Ultrasonics*, vol. 41, pp. 451–463, August 2003.

[25] T. Hemsel and J. Wallaschek, "Survey of the present state of the art of piezoelectric linear motors," *Ultrasonics*, vol. 38, pp. 37–40, March 2000.

[26] V. Snitka, "Ultrasonic actuators for nanometre positioning," *Ultrasonics*, vol. 38, pp. 20–25, March 2000.

[27] M. Kümmel, S. Goldschmidt, and J. Wallaschek, "Theoretical and experimental studies of piezoelectric ultrasonic linear motor with respect to damping and nonlinear material behaviour," *Ultrasonics*, vol. 36, pp. 103–109, February 1998.

[28] K. K. Tan, T. H. Lee, and H. X. Zhou, "Micro-positioning of linear-piezoelectric motors based on a learning nonlinear pid controller," *IEEE/ASME Trans. on Mechatronics*, vol. 6, pp. 428–436, December 2001.

[29] F. J. Lin, R.-J. Wai, K. K. Shyu, and T. M. Liu, "Recurrent fuzzy neural network control for piezoelectric ceramic linear ultrasonic motor drive," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 48, pp. 900–913, July 2001.

[30] R.-J. Wai, F. J. Lin, R. Y. Duan, K. Y. Hsieh, and J. Lee, "Robust fuzzy neural network control for linear ceramic motor drive via backstepping design technique," *IEEE Trans. on Fuzzy Systems*, vol. 10, pp. 102–112, February 2002.

[31] R.-J. Wai, C.-M. Lin, and Y.-F. Peng, "Robust CMAC neural network control for LLCC resonant driving linear piezoelectric ceramic motor," *IEE Proc. Control Theory Appl.*, vol. 150, pp. 221–232, May 2003.

[32] F. J. Lin, R.-J. Wai, and M. P. Chen, "Wavelet neural network control for linear ultrasonic motor drive via adaptive sliding-mode technique," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 50, pp. 686–698, June 2003.

[33] R.-J. Wai, "Robust control of linear ceramic motor drive with LLCC resonant technique," *IEEE Trans. Ultrason., Ferroelect., Freq. Contr.*, vol. 50, pp. 911–920, July 2003.

[34] J. P. Shields, *Basic Piezoelectricity.* W. Foulsham, Slough, Bucks, 1966.

[35] U. Schaaf, "Pushy motors," *IEE Review*, pp. 105–108, May 1995.

[36] *Piezoelectric Ceramics: Principles and Applications.* APC International Ltd., 2002.

[37] J. Zumeris, "Ceramic motor," *United States Patent 5777423*, 1998.

[38] Nanomotion Ltd., *AB1A Driver Box User Manual*, 2^{nd} ed., August 2001.

[39] T. Senjyu, T. Kashiwagi, and K. Uezato, "Position control of ultrasonic motors using marc and deadzone compensation with fuzzy inference," *IEEE Trans. on Power Electronics*, vol. 17, pp. 265–272, March 2002.

[40] J. G. V. Utkin and J. Shi, *Sliding Mode Control in Electromechanical Systems.* Taylor and Francis, 1999.

[41] P. K. Nandam and P. C. Sen, "Industrial applications of sliding mode control," *IEEE/IAS International Conference on Industrial Automation and Control*, pp. 275–280, January 1995.

[42] D. Q. Zhang and S. K. Panda, "Chattering-free and fast response sliding mode controller," *IEE Proc.-Control Theory Appl.*, vol. 146, pp. 171–177, March 1999.

[43] W. J. Wang and J. L. Lee, "Hitting time reduction and chattering attenuation in multi variable structure systems," *Control Theory Adv. Technol.*, pp. 491–499, 1993.

[44] J.-X. Xu, Y.-J. Pan, and T.-H. Lee, "Sliding mode control with closed-loop filtering architecture for a class of nonlinear systems," *IEEE Trans. on Circuits and Systems - II: Fundamental Theory and Applications*, vol. 51, pp. 168–173, April 2004.

[45] P. Ya-Jun, *Acquistion and Applications of Equivalent Contrrol Approaches in Variable Structure Control*. Phd thesis, National University of Singapore, 2003.

[46] T. J. A. de Vries, W. J. R. Velthuis, and J. van Amerongen, "Learning feed-forward control: A survey and historical note," *Proceeding of First IFAC Conference on Mechatronic Systems*, pp. 949–954, September 2000.

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

[48] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *Int. J. Control*, vol. 731, no. 10, pp. 930–954, 2000.

[49] S. K. Sahoo, S. K. Panda, and J.-X. Xu, "Iterative learning-based high-performance current controller for switched reluctance motors," *IEEE Transactions on Energy Conversion*, vol. 19, pp. 491–498, September 2004.

[50] N. C. Sahoo, J.-X. Xu, and S. K. Panda, "Low torque ripple control of switched reluctance motors using iterative learning," *IEEE Transactions on Energy Conversion*, vol. 16, pp. 318–326, December 2001.

[51] W. Qian, S. K. Panda, and J.-X. Xu, "Torque ripple minimization in pm synchronous motor using iterative learning control," *IEEE Trans. Power Electronics*, vol. 19, pp. 272–279, March 2004.

[52] K. K. Tan, H. Dou, Y. Chen, and T. H. Lee, “High precision linear motion control via relay-tuning and iterative learning based on zero-phase filtering,” *IEEE Trans. on Control Systems Technology*, vol. 9, pp. 244–253, March 2001.

[53] R. W. Longman and S.-L. Wirkander, “Automated tuning concepts for iterative learning and repetitive control laws,” *IEEE Conference Proc. on Decision and Control*, pp. 192–198, December 1998.

[54] J.-X. Xu, “Direct learning of control efforts for trajectoris with different time scales,” *IEEE Trans. on Automatic Control*, vol. 43, no. 7, pp. 1027–1030, 1998.

[55] J.-X. Xu, “Dual-scale direct learning control of trajectory tracking for a class of nonlinear uncertain systems,” *IEEE Trans. Automatic Control*, vol. 44, pp. 1884–1888, October 1999.

[56] dSPACE, *DS1104 R&D Controller Board Feature Reference.* dSPACE GmbH, May 2002.

List of Publications

Conference Papers

1. Krishna Mainali, Sanjib Kumar Panda, Jian-Xin Xu, Tomonobu Senjyu, "Repetitive Position Tracking Performance Enhancement of Linear Ultrasonic Motor with Sliding Mode-cum-Iterative Learning Control", *IEEE International Conference on Mechatronics*, Istanbul, Turkey, 2004.

2. Krishna Mainali, Sanjib Kumar Panda, Jian-Xin Xu, Tomonobu Senjyu, "Position Tracking Performance Enhancement of Linear Ultrasonic Motor Using Iterative Learning Control", *35th Annual IEEE Power Electronics Specialist Conference*, Achen, Germany, 2004. pp. 4844 - 4849.

3. Krishna Mainali, Sanjib Kumar Panda, Jian-Xin Xu, "Position Tracking of Linear Piezoelectric Motor Using Sliding Mode Control with Closed-loop Filtering", *The 30th Annual Conference of the IEEE Industrial Electronics Society*, Busan, Korea, November, 2004.

4. Krishna Mainali, Sanjib Kumar Panda, Jian-Xin Xu, "Position Tracking Performance Enhancement of Linear Ultrasonic Motor with Direct Learning Control Techniques", *International Conference on Power System Technology - POWERCON*, Singapore, November 2004.

Submitted Journal Papers

1. Krishna Mainali, Sanjib Kumar Panda, Jian-Xin Xu, "Linear Ultrasonic Mo-

tor Positioning Performance Enhancement Using Iterative Learning Control", *submitted for review to IEEE/ASME Trans. on Mechatronics.*

Appendix A

Motor Parameters and Structure

1. **Performance Parameters**

Maximum Allowable Velocity :	250 [mm/sec]
Dynamic Stall Force :	30 to 36 [N]
Static Holding Force :	28 [N]
Non-Energized Stiffness :	3.3 to 3.8 [N/μ]
Nominal Preload on Stage :	144 [N]
Force constant K_f :	4 to 8 [N/Volt]
Velocity damping factor B_v :	120 to 144 [N s/m]
Attainable Resolution :	100 nm
Nominal Lifetime :	20,000 hours

2. **Electrical Specification**

Maximal Voltage :	270 Vrms, 39.6 KHz, sine wave
Maximal Current consumption :	600 mA rms
Maximal Power Consumption :	30 W

3. **Physical Dimensions of Motor**

Length :	41.9 mm
Width :	46.6 mm
Height :	23.8 mm

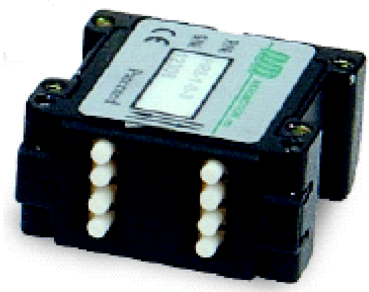


Figure A.1: HR8 ultrasonic motor.

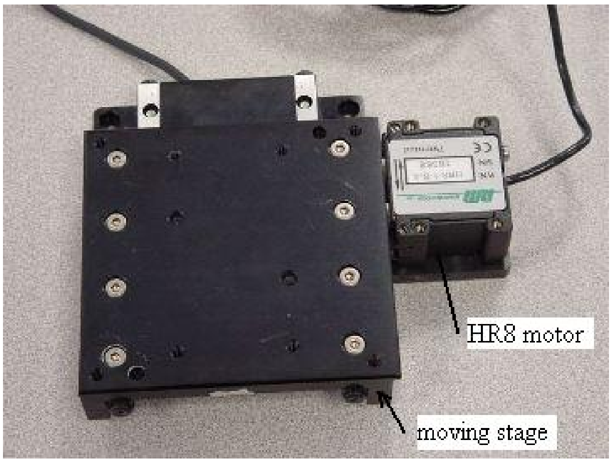


Figure A.2: HR8 ultrasonic motor with stage (stroke length=58 mm).

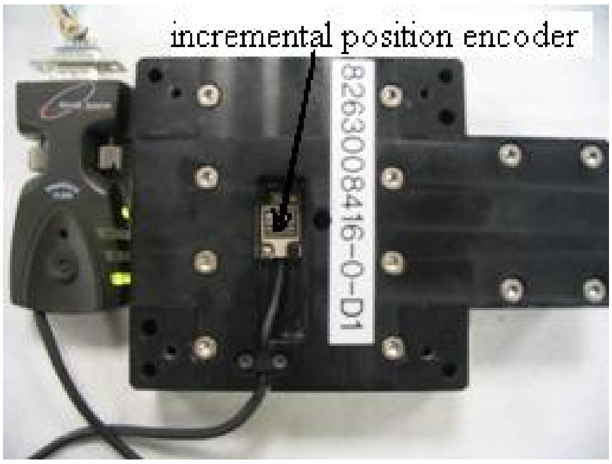


Figure A.3: HR8 ultrasonic motor with incremental position encoder (bottom view).

Appendix B

Motor Driver

1. **Specifications**

Input voltage range:	± 10 V
Input signal type:	Differential or single ended
Maximum output voltage:	260 Vrms
Power supply:	+ 48 V± 5%
Current consumption:	125 mA without motor
	1200 mA with one HR8 motor connected

2. **Description**

   The AB1A amplifier can drive upto 32 motor elements in parallel. The AB1A may be operated in velocity mode (continuous drive) or step mode (discrete step drive). The AB1A driver contains the AB1A Card and LC card. The AB1A card converts the analog input command signal into corresponding PWM square wave output signal that is fed to the LC card. The LC card filters the signal to produce the output voltage that drives the motor. The required DC voltages for the driver are supplied by internal DC-DC converter that is fed from an external + 48 V power supply. The schematic of the AB1A driver is shown in Fig. B.1.

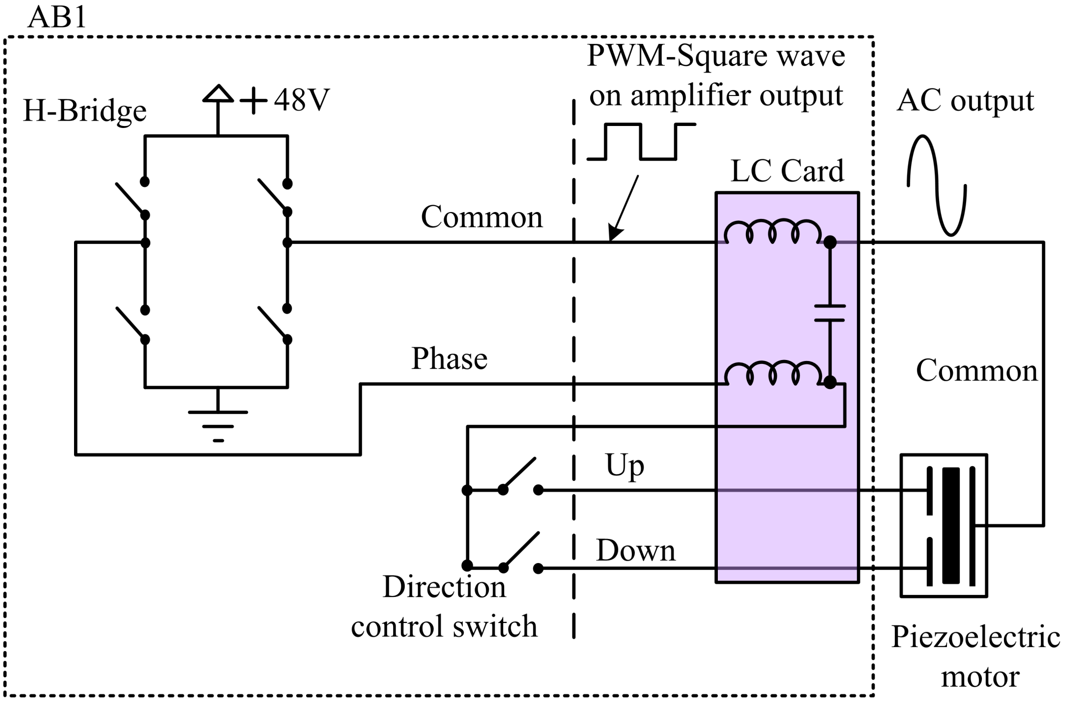


Figure B.1: Schematic diagram of AB1A Motor Driver.

Appendix C

Architecture of DS1104

Fig. C.1 shows the architecture of the DS1104 controller board. The DS1104's main processing unit consists of a PowerPC 603e microprocessor (master PPC) running at 250MHz (CPU clock) containing data and instruction cache of 16KB each. It has an interrupt controller, a synchronous DRAM controller, several timers, a PCI interface. The master PPC controls the fully programmable ADC unit, DAC unit, 20-bit I/O unit, incremental encoder interface, serial interface. The PCI interface provides an access from/to the host PC via 33 MHz-PCI interface. The interface serves the board setup, program downloads and runtime data transfers from/to the host PC. The host interface also provides a bidirectional interrupt line. Via this line, the host PC can send interrupt requests to the master PPC and vice versa.

The DS1104's slave DSP subsystem consists of Texas Instruments TMS320F240 DSP Running at 20 MHz. The slave DSP on the DS1104 provides 14-bit direction selectable digital I/O unit. The slave DSP on the DS1104 provides a timing I/O unit that can be used to generate and measure pulse width modulated (PWM) and square-wave signals. It also controls a serial peripheral interface (SPI), which can be used to perform high-speed synchronous communication with devices connected to the DS1104, such as an A/D converter.

The DS1104 R&D Controller Board upgrades PC to a development system for rapid control prototyping. The real-time hardware based on the PowerPC 603e microprocessor and its I/O interfaces make the board ideally suited for developing

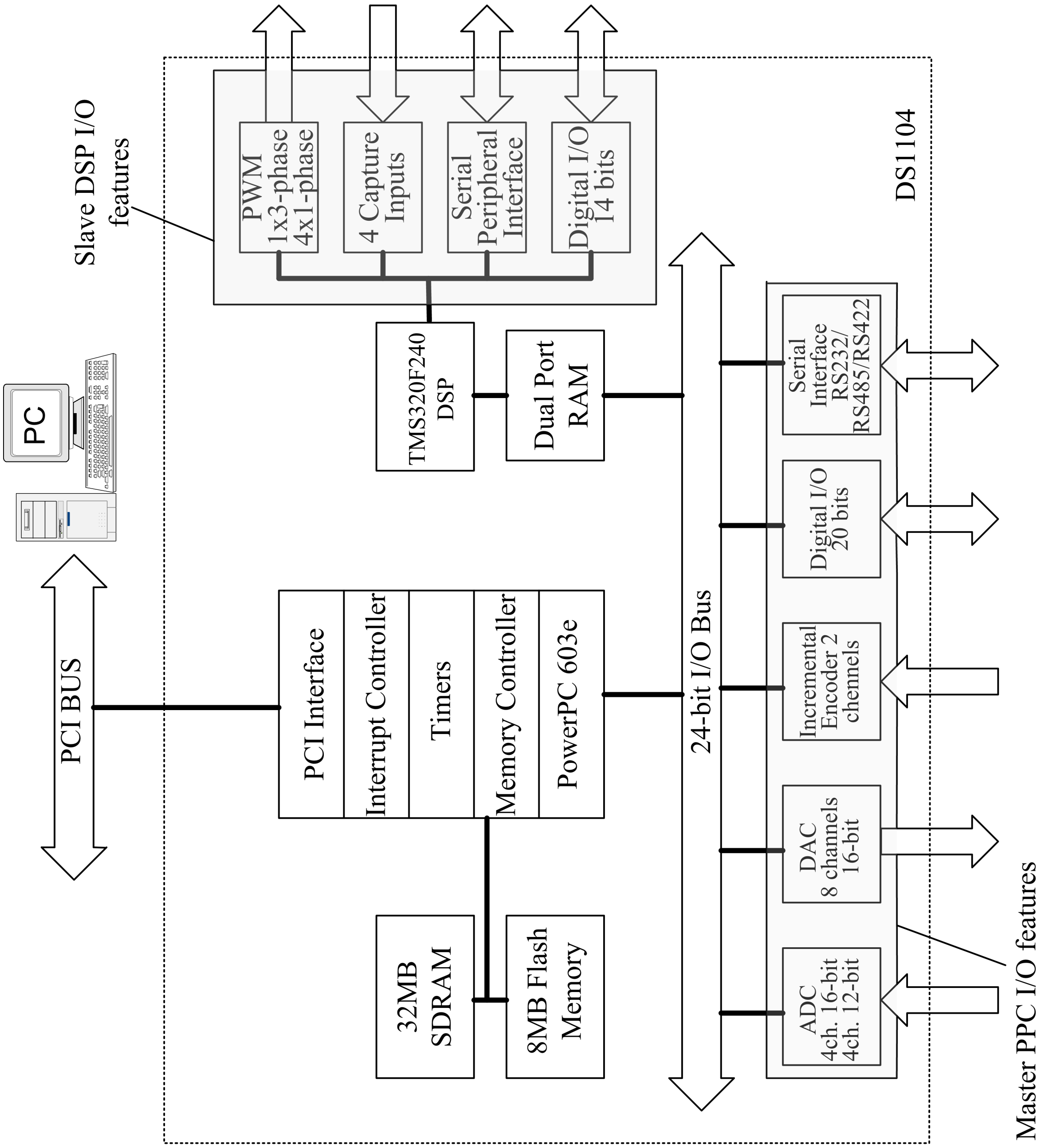


Figure C.1: Architecture of DSP DS1104 controller board.

controllers in various fields [57]. Detailed description of the DS1104 board are given in dSPACE User's Guide.

Appendix D

Position Control Program

The following C program was developed for the position control of linear ultrasonic motor. The program has implemented sliding mode controller with iterative learning and direct learning control schemes. The program can be used for set point position tracking and sinusoidal position tracking.

```
/*****************************************************************
*Program for the Position Control of Linear Ultrasonic Motor
*Written by Krishna Mainali, M.Eng Student, Department of ECE,
*National University of Singapore, Singapore
*2003-2004
*****************************************************************/
/*.........INCLUDING THE FILES.....................*/
#include"d:\g0203514\lusm\brtenv.h”
#include"d:\g0203514\lusm\math.h”
#include"d:\g0203514\lusm\const.h”
#include"d:\g0203514\lusm\var_init.h”
#include"d:\g0203514\lusm\control_func.h”


/*...........READ THE HARDWARE_INDEX...............*/
void read_hw_index(void)
  {
    /* set flag for index found */
    if (ind_reset)
       temp = ds1104_inc_index_read(1, DS1104_INC_IDXMODE_ON);
    else
       temp = ds1104_inc_index_read(1, DS1104_INC_IDXMODE_OFF);
    if (temp == DS1104_INC_IDX_SET)
      ind_found = temp;
  }

/*.............READ THE POSITION..................*/
void read_pos_vel(void)
{ read_hw_index();
  /* encoder interrupt functions for digital channel 1 */
  /* read with highest resolution, 1/4 line */
  /*position in mm*/
  act_pos = (ds1104_inc_position_read(1,DS1104_INC_LINE_SUBDIV_4)*PosFact)
            -pos_offset;
  if((home==0)&&(ind_found==1)&&(move==0))
  pos_offset=(ds1104_inc_position_read(1,DS1104_INC_LINE_SUBDIV_4)*PosFact);
  /* calculate the velocity in mm per second */
  v=(act_pos-old_position)/Ts;
  old_position=act_pos;
  /*v = (ds1104_inc_delta_position_read(1,DS1104_INC_LINE_SUBDIV_4)/Ts)*VelFact;*/
}

/*.....CONTROL VOLTAGE FOR SPECIFIED TIME(macro)....*/
void controller(void)
{   if(SCURVE==1)
     {SINECURVE=0;
      S_curve();
     }
     if(SINECURVE==1)
     {SCURVE=0;
      sine_wave();
     }
     voltage=final_control();
}


/*..........DECISION ON CONTROL VOLTAGE..................*/
void control_input(void)
{ if((home==1)&&(ind_found==1))
  {
   home=0;
   voltage=0;
   ref_pos_k=0;i=0;pp=0;S0=0;reached=0; /*intializing the reference position*/
  }

else if((home==1) && (ind_found==0)&& (emergency_stop==0) )
  { CENTER=0;
    move=0;
    voltage=-2.5;
  }

  else if((move==1)&&(home==0)&&(emergency_stop==0&&(CENTER==0)))
          {
            controller(); /* calling the controller to apply appropriate signal */
            ind_found=0;
          }
  else if(emergency_stop==1)
  voltage=0;
    else if((home==0)&&(CENTER==1)&&(act_pos!=28))
         { move=0;ind_found=0;
            voltage=((28-act_pos)/fabs(28-act_pos))*1.5;
          }
 else if((CENTER==1)&&(act_pos==28))
   {CENTER=0;voltage=0;/*pos_offset=pos_offset+28;*/}
   if((CENTER==0)&&(move==0)&&(home==0))
      pos_offset=pos_offset+act_pos;


if (voltage>4) voltage=2;                /* upper limit of the voltage */
  if (voltage<-4) voltage=-2;            /* lower limit of the volatge */
  voltage_reduced=voltage*Signal_reduce; /* downscaling the control voltage by 0.1 */
  ds1104_dac_write(1, voltage_reduced);  /* set DACH1 to contol voltage */
  ds1104_dac_write(2,pos_err*0.1);
}

/*...............SERVICE ROUTINE....................*/
void isr_srt(void)
{ RTLIB_SRT_ISR_BEGIN();    /* overload check */
  host_service(1,0);         /* Data Acquisition service */
  RTLIB_TIC_START();         /* start time measurement */
  read_pos_vel();            /* function reading the position */
  control_input();           /* function to decide the control input */
  exec_time=RTLIB_TIC_READ();/* code execution time */
  RTLIB_SRT_ISR_END();       /* overload check */
}

/*..................MAIN FUNCTION..................*/
void main(void)
{ move=MOVE;
  home=HOME;
  emergency_stop=Emerg_Stop;
  voltage=CONT_V;
  for(a=0;a<n;a++)
```

```
   { past_input[a]=0;       /* memory intialisation */
     present_error[a]=0;
   }
  init();                   /* DS1104 and RTLib1104 initialization */
  /* init D/A conv in Transparent mode */
  ds1104_dac_init(DS1104_DACMODE_TRANSPARENT);
  /* initialize channel 1 for TTL input signal*/
  ds1104_inc_init(1, DS1104_INC_MODE_TTL);
   /* set reset on index for channel 1 and latch to 0 */
  ds1104_inc_set_idxmode(1, DS1104_INC_IDXMODE_ON);
  msg_info_set(MSG_SM_RTLIB,0,"SYSTEM STARTED GOTO CONTROL DESK.");
  RTLIB_SRT_START(Ts,isr_srt);          /* start sample rate timer */
  while(1)
    {
        RTLIB_BACKGROUND_SERVICE();     /* background service */
    }
}

/**********************************************************************
*const.h
*defines all the constants used in the program
**********************************************************************/
#define Ts 0.000050         /* sample time */
#define PosFact -0.0004     /* position encoder multiplication factor to
                               convert the position in mm */
#define VelFact -0.0004     /* velocity multiplication factor */
#define Signal_reduce 0.1   /* siganl reduction factor as the DAC input
                               should be within -1 to 1 */
#define Emerg_Stop 0        /* for emergency stop 0 */
#define MOVE 0              /* initial move command */
#define HOME 0              /* initial Home command */
```

```
#define CONT_V 0           /* control voltage */

#define TIME 0             /* time for the application for the control */

/*** motor parameters ***/

#define MASS 0.8           /* Mass of the moving stage */

#define FORCE_CONST 6      /* Force constant */

#define B 0.132            /* 132N/m/s=132N/1000mm/s */

#define PI 3.141592654     /* value of PIE */


/*********************************************************************

*var_init.h

*all variables used in the program are declared and initialised

*********************************************************************/

Float64 voltage;            /* control volatge to AB1A driver */

Float64 act_pos;            /* motor position */

Float64 v;                  /* velocity */

Float64 exec_time;          /* execution time */

Int32 move;                 /* move command */

Int32 home;                 /* gotostarting point i.e homing */

Int32 emergency_stop;       /* emergency stop */

Int32 CENTER=0;             /* bring stage center command */

Int32 ind_found = 0;        /* index found flag */

Int32 ind_reset = 1;

Int32 temp;                 /* temporary variable */

Float64 pos_offset=0;

Float64 voltage_reduced;    /* reduced control voltage */

Float64 pos_err0=0;         /* previous sample position error*/

Float64 pos_err;            /* position error in mm */

Float64 pos_err_um=0;       /* position error in micrometer */

Float64 ref_speed=0;        /* reference speed */

Float64 ref_pos_k=0;        /* reference position */

Float64 ref_pos_0=0;        /* initial reference position */
```

```
Float64 old_position=0;      /* old motor position */


/*** S curve parameters ***/

Float64 S0=0.0;

Float64 S=20;                /* set point reference */

Float64 Jerk=0;              /* Jerk */

Float64 acc;                 /* acceleration */

Float64 vel;                 /* velocity */

Float64 s;                   /* intantaneous path value */

Float64 reached=0;           /* flag for reaching set-point */

Float64 t=0;

Float64 ref_pos=0;

Float64 rise_time=1.0;       /* rise time of s-curve */

Int32 pp=0;                  /* counter */

Int32 SCURVE=0;              /* set point flag */


/*** sine curve parameters ***/

Float64 i=0;                 /* time counter */

Float64 T;                   /* sine period */

Int32 SINECURVE=0;           /* sine reference flag */

Float64 Amp=20;              /* sine amplitude */

Float64 Amp0=20;             /* initial sine amplitude */

Float64 Freq=0.25;           /* sine frequency */

Float64 Freq0=0.25;          /* initial sine frequency */


/*Sliding Mode Controller*/

Float64 Ueq=0;               /* equivalent control */

Float64 BETAN=0.4;           /* negative switching gain */

Float64 BETAP=0.4;           /* positive switching gain */

Float64 BETA1;

Float64 PHI1=100;            /* bounday layer thickness */
```

```
Float64 ALFA1=0.002;         /* extra term in control */
Float64 LAMDA1=10000;        /* s=lamda*e+dot(e) */
Float64 spd1;
Float64 ss1;
Float64 Us1=0;               /* switching control */
Float64 Uc1=0;               /* nominal equivalent control */
Float64 S1=0;                /*  sliding surface */
Float64 deadvoltage=0;       /* dead zone compensation */
Float64 deadvoltagep=0.9;
Float64 deadvoltagen=-0.8;
Float64 d_pos_error=0;
Int32 UEQ=0;

/*** for Iterative Learning Control ***/
Float64 past_input[250000];   /* memory for previous cycle input*/
Float64 present_error[250000];/* memory for previous cycle error*/
Float64 present_input=0;      /* ILC present cycle control */
Float64 learning_gain=10;     /* learning gain L*/
Float64 forget_fact=0;        /* forgetting factor Q */
Int32  n=250000;              /* no of samples in one period */
Int32  sample_count=0;        /* ILC time index counter */
Int32 ILC=0;                  /* ILC controller flag
Int32 a=0;
Int32 timer;
Int32 lead=30;                /* phase adjustment */
Float64 lpf_err=0;
Float64 first_time=1;
Float64 d_ref_pos1;
Float64 dd_ref_pos;
Float64 d_ref_pos0=0;
```

```
/*** Low pass filter ***/
Float64 Mov_Avg_fc=1;         /* cutt off frequency */
Float64 filter_op=0;          /* filter ouput */
Float64 BBB=0;                /* filter coefficient */
Float64 y;                    /* output vector */
Int32 M=0;                    /* filter length */
Int32 j;

/*** dynamics part ***/
Int32 no_learn=0;           /* to disable the ILC */
Int32 no_learn_counter=0;   /* count time index of new reference */

/*** direct learning of control ***/
Int32 LEARN=0;          /* flag to learn */
Int32 NO_DILC=1;        /* flag to select DLC-ILC */
Int32 DILC=0;
Int32 DLCILC=1;
Int32 learned1=0;        /* flag showing end of 1st control learning */
Int32 learned2=0;        /* flag showing end of 2nd control learning */
Int32 learned3=0;        /* flag showing end of 3rd control learning */
Int32 learned4=0;        /* flag showing end of 4th control learning */
Int32 learned5=0;        /* flag showing end of 5th control learning */
Int32 learned6=0;        /* flag showing end of 6th control learning */
Int32 start_learn1=0;    /* flag to start learning 1st control voltage */
Int32 start_learn2=0;    /* flag to start learning 2nd control voltage */
Int32 start_learn3=0;    /* flag to start learning 3rd control voltage */
Int32 start_learn4=0;    /* flag to start learning 4th control voltage */
Int32 start_learn5=0;    /* flag to start learning 5th control voltage */
Int32 start_learn6=0;    /* flag to start learning 6th control voltage */
Int32 no_learn_count=0;  /* counter */
Int32 direct_learn=0;
```

```
Int32 learn_counter=0;
Int32 index1=0;          /* memory index for 1st control voltage */
Int32 index2=0;          /* memory index for 2nd control voltage */
Int32 index3=0;          /* memory index for 3rd control voltage */
Int32 index4=0;          /* memory index for 4th control voltage */
Int32 index5=0;          /* memory index for 5th control voltage */
Int32 index6=0;          /* memory index for 6th control voltage */
Float64 control;
Float64 freq1,freq2,freq3,freq4,freq5,freq6; /* frequencies to be learned */
Float64 freqd=0.2;       /* new desired frequency */
Float64 data1[160001];   /* memory for 1st control voltgae */
Float64 data2[114287];   /* memory for 2nd control voltgae */
Float64 data3[100001];   /* memory for 3rd control voltgae */
Float64 data4[88889];    /* memory for 4th control voltgae */
Float64 data5[66667];    /* memory for 5th control voltgae */
Float64 data6[50000];    /* memory for 6th control voltgae */
Float64 xu1,xu2,xu3,xu4,xu5,xu6;
Float64 map_time1,map_time2,map_time3,map_time4,map_time5,map_time6;
Float64 fract_check;
Float64 interpolation;
Float64 control_v1,control_v2,control_v3,control_v4,control_v5,control_v6;
Float64 control_v;
Float64 control_vd=0;      /* direclty estimated control votlage */

/*******************************************************************
*control_func.h
*control functions are coded here
*******************************************************************/
/******************** LOW PASS FILTER ******************/
float mov_avg_filter(int time_index)   /* moving average low pass filter*/
{ M=(int)(1.392/(2*PI*Mov_Avg_fc*Ts)); /* size of the averager for given cuttoff*/
```

```
  BBB=1.0/(2*M+1);                    /* coefficient of the low pass filter*/
  filter_op=0;
  for(j=-M;j<=M;j++)
  { if(((time_index+j)>=0)&&((time_index+j)<n))
    filter_op=filter_op+BBB*present_error[time_index+j];
    else if((time_index+j)<0)
          filter_op=error1+BBB*present_error[n+j];
         else if((time_index+j)>=n)
           filter_op=filter_op+BBB*present_error[(time_index+j)-n];
  }
  y=filter_op;
  filter_op=0;
}

/************* NEW CYCLE OF THE SINE WAVE *****************/
void new_cycle(void)
{  i = 0;              /* intialise time sequence at the end of time period */
   Amp0=Amp;           /* assign new amplitude dynamics */
   if(Freq!=Freq0)     /* for frequnecy dynamics */
    { no_learn=1;      /* off the ILC during first cycle of new frequency */
      sample_count=0;
       if(no_learn_count==6)/* if all waveforms learned and new frequency */
       { NO_DILC=0; DILC=1; /* change comes,then control from DILC */
            freqd=Freq;     /* new freq after learning is desired frequency */
       }
     }
   Freq0=Freq;freqd=Freq;n=(1/Freq0)/Ts;
   /* For the direct learning of control process */
   if(LEARN==1)
   no_learn_count=no_learn_count+1;
   if((no_learn_count==1)&&(LEARN==1))/* starts the first wave learning */
   { start_learn1=1; freq1=Freq0;}    /* from the beginning of the cycle */
   if(learned1==1)                    /* stops the first wave learning */
      start_learn1=0;
   if((no_learn_count==2)&&(LEARN==1))/* starts the second wave learning */
   { start_learn2=1; freq2=Freq0;}
   if(learned2==1)                    /* stops the second wave learning */
      start_learn2=0;
   if((no_learn_count==3)&&(LEARN==1))/* starts the third wave learning */
   { start_learn3=1; freq3=Freq0;}
   if(learned3==1)                    /* stops the third wave learning */
      start_learn3=0;
   if((no_learn_count==4)&&(LEARN==1))/* starts the fourth wave learning */
   { start_learn4=1; freq4=Freq0;}
   if(learned4==1)                    /* stops the fourth wave learning */
      start_learn4=0;
   if((no_learn_count==5)&&(LEARN==1))/* starts the fifth wave learning */
   { start_learn5=1; freq5=Freq0;}
   if(learned5==1)                    /* stops the fifth wave learning */
      start_learn5=0;
   if((no_learn_count==6)&&(LEARN==1))/* starts the sixth wave learning */
   { start_learn6=1; freq6=Freq0;}
   if(learned6==1)                    /* to stop the sixth wave learning */
      start_learn6=0;
   direct_learn=0;        /* starting the time index for direct learning */
   learn_counter=0;       /* index of the data learning part */
   if(LEARN==1)           /* reinitialising LEARN flag */
   LEARN=0;
}

/**************S-CURVE POSITION REFERENCE****************/
void S_curve(void)
```

```
{Float64 T,T1,T2,T3,T4; /* to minimise jerk, half time acceln, half decceln */
  T=rise_time;T1=T/4;T2=T/2;T3=3*T1;T4=T;
  Jerk=((ref_pos-S0)/2)*(64/pow(T,3));
  t=pp*Ts;
  if((t<T1)&&(reached!=ref_pos))/* uniformly increasing +acceleration */
   { acc=0+Jerk*t;
     vel=0+0.5*Jerk*pow(t,2);
     s=S0+(1.0/6)*Jerk*pow(t,3);
     pp=pp+1;
   }
  else if((t<T2)&&(reached!=ref_pos))/* uniformly decreasing +acceleration */
   { acc=Jerk*T1-Jerk*(t-T1);
     vel=(0.5*Jerk*pow(T1,2))+(Jerk*T1)*(t-T1)-0.5*Jerk*pow((t-T1),2);
     s=S0+((1.0/6)*Jerk*pow(T1,3))+(0.5*Jerk*pow(T1,2))*(t-T1)
       +0.5*(Jerk*T1)*pow((t-T1),2)-(1.0/6)*Jerk*pow((t-T1),3);
     pp=pp+1;
   }
   else if((t<T3)&&(reached!=ref_pos))/* uniformly decreasing -acceleration */
    { acc=Jerk*T1-Jerk*(T2-T1)-Jerk*(t-T2);
      vel=(0.5*Jerk*pow(T1,2))+(Jerk*T1)*(T2-T1)-0.5*Jerk*pow((T2-T1),2)
          +(Jerk*T1-Jerk*(T2-T1))*(t-T2)-0.5*Jerk*pow((t-T2),2);
      s=S0+(((1.0/6)*Jerk*T1*T1*T1)+(0.5*Jerk*T1*T1)*(T2-T1)
          +0.5*(Jerk*T1)*(T2-T1)*(T2-T1)-(1.0/6)*Jerk*(T2-T1)*(T2-T1)*(T2-T1))
          +((0.5*Jerk*T1*T1)+(Jerk*T1)*(T2-T1)-0.5*Jerk*(T2-T1)*(T2-T1))*(t-T2)
          +(0.5*(Jerk*T1-Jerk*(T2-T1))*(t-T2)*(t-T2))
          -(1.0/6)*Jerk*(t-T2)*(t-T2)*(t-T2);
      pp=pp+1;
    }
    else if((t<T4)&&(reached!=ref_pos))/* uniformly increasing -acceleration */
     { acc=(Jerk*T1-Jerk*(T2-T1)-Jerk*(T3-T2))+Jerk*(t-T3);
       vel=(((0.5*Jerk*T1*T1)+(Jerk*T1)*(T2-T1)-0.5*Jerk*(T2-T1)*(T2-T1))
           +(Jerk*T1-Jerk*(T2-T1))*(T3-T2)-0.5*Jerk*(T3-T2)*(T3-T2))
           +(Jerk*T1-Jerk*(T2-T1)-Jerk*(T3-T2))*(t-T3)+0.5*Jerk*(t-T3)*(t-T3);
       s=S0+(((1.0/6)*Jerk*T1*T1*T1)+(0.5*Jerk*T1*T1)*(T2-T1)
           +0.5*(Jerk*T1)*(T2-T1)*(T2-T1)-(1.0/6)*Jerk*(T2-T1)*(T2-T1)*(T2-T1))
           +((0.5*Jerk*T1*T1)+(Jerk*T1)*(T2-T1)-0.5*Jerk*(T2-T1)*(T2-T1))*(T3-T2)
           +(0.5*(Jerk*T1-Jerk*(T2-T1))*(T3-T2)*(T3-T2))
           -(1.0/6)*Jerk*(T3-T2)*(T3-T2)*(T3-T2)+(((0.5*Jerk*T1*T1)
           +(Jerk*T1)*(T2-T1)-0.5*Jerk*(T2-T1)*(T2-T1))
           +(Jerk*T1-Jerk*(T2-T1))*(T3-T2)-0.5*Jerk*(T3-T2)*(T3-T2))*(t-T3)
           +0.5*(Jerk*T1-Jerk*(T2-T1)-Jerk*(T3-T2))*(t-T3)*(t-T3)
           +(1.0/6)*Jerk*(t-T3)*(t-T3)*(t-T3);
       pp=pp+1;
     }
     else
      { acc=0;
        vel=0;
        S0=ref_pos;
        reached=s;
        pp=0;
      }
    ref_pos_k=s;
}
/************ SINE WAVE POSITION REFERENCE ***************/
void sine_wave(void)
{   /* generate a sine function with period T */
   T=1/Freq0;
   ref_pos_k= Amp0*sin(2*PI*Freq0*i);
   i=i+Ts;
   if (i >= T)
   new_cycle();
}
```

```
/**********DERIVATIVES OF REFERENCE POSITION**************/
void ref_derivative(void)
{  /**************Reference speed***************/
   ref_speed=(ref_pos_k-ref_pos_0)/Ts;
   /* first derivative of the reference signal */
   d_ref_pos1=ref_speed;
   ref_pos_0=ref_pos_k;        /*updating the reference position*/
   /* second derivative of the reference signal */
   dd_ref_pos=(d_ref_pos1-d_ref_pos0)/Ts;
   d_ref_pos0=d_ref_pos1;      /*updating the first derivative*/
}

/************** ITERATIVE LEARNING CONTROL ***************/
void learning_control(void)
{
 if(no_learn==0)
 { timer=sample_count+lead;
   if(timer>n) timer=timer-n;
  lpf_err=mov_avg_filter(timer);     /* using filtered error in learning */
  present_error[sample_count]=pos_err;
  present_input=(1-forget_fact)*past_input[sample_count]+learning_gain*lpf_err;
  past_input[sample_count]=present_input;
  sample_count=sample_count+1;
  timer=timer+1;
  if(sample_count==n)
     sample_count=0;
 }
 else
 { /* ilc output made zero,while the error is being learned */
  present_input=0;
  present_error[sample_count]=pos_err;
  sample_count=sample_count+1;
  if(sample_count==n)
     sample_count=0;
 }
}


/************* SMC POSITION CONTROL *****************/
void SMC_control(void)
{ /* finding the equivalent control input */
  d_pos_error=(pos_err-pos_err0)/Ts; /*first derivative of the error*/
  pos_err0=pos_err;
  Ueq=(MASS/FORCE_CONST)*(dd_ref_pos+LAMDA1*d_pos_error)*0.001;
  /* multiplied by 0.001 as the position is in mm and the force
  constant is in N/V=(kg*m/(s*s))/V */
  /* for the switching part */
  S1=LAMDA1*pos_err+1*d_pos_error; /* PD sliding */
  spd1=S1/PHI1;                    /* S/phi */
  if(fabs(spd1)<1) ss1=spd1;       /* using Us=beta*sat(s/phi) */
   else if(spd1>=1) ss1=1;         /* sat(s/phi)=s/phi,if abs(s/phi)<1 */
          else ss1=-1;             /* else equal to the sgn(s/phi) */
  if (spd1>0) BETA1=BETAP;
   else if(spd1<0) BETA1=BETAN;
  Us1=BETA1*ss1;                   /* switching control=beta*sat(s/phi) */
  if(UEQ==0)                       /* turn off equivalent control */
  Uc1=Us1+ALFA1*S1;
  if(UEQ==1)                       /* turn on equivalent control */
  Uc1=Us1+ALFA1*S1+Ueq;
}


/************ DLC: LEARNING CONTROL PROFILES **********/
```

```
void learn_history(void)
{ if(start_learn1==1)
  { data1[learn_counter]=control;/* learning 1st control voltage */
    learned1=1;
  }
  if(start_learn2==1)
  { data2[learn_counter]=control;/* learning 2nd control voltage */
    learned2=1;
  }
  if(start_learn3==1)
  { data3[learn_counter]=control;/* learning 3rd control voltage */
    learned3=1;
  }
  if(start_learn4==1)
  { data4[learn_counter]=control;/* learning 4th control voltage */
    learned4=1;
  }
  if(start_learn5==1)
  { data5[learn_counter]=control;/* learning 5th control voltage */
    learned5=1;
  }
  if(start_learn6==1)
  { data6[learn_counter]=control;/* learning 6th control voltage */
    learned6=1;
  }
  learn_counter=learn_counter+1;
}

/**************  SMC_ILC and no DILC CONTROL INPUT *****************/
void no_dilc(void)
{   SMC_control();
```

```
    if(no_learn==1)
    {
     past_input[no_learn_counter]=0;
     no_learn_counter=no_learn_counter+1;
    }
    if(no_learn_counter==n)
    {no_learn=0;
     no_learn_counter=0;
    }
    if(ILC==1)
    { if(first_time==1)   /* for dynamics intialising the memory size */
      { if(SINECURVE==1)
        n=T/Ts;
        first_time=0;
      }
      learning_control();
    }
    control=Uc1+present_input;
    control_v=control;  /* just to use the one variable at the final_control */
                        /* smc_ilc and dilc returns the control_v */
    learn_history();    /* to learn the similar spacial  trajectories */
}

/************ DLC MAPPING AND COMPUTING THE CONTROL ************/
void dilc(void)
{ if(freqd==0.08)
  { /* Regressor Matrix solution for freqd=0.08hz */
    xu1=0.67249863313286; xu2=0.42372881355932; xu3=0.32422088572991;
    xu4=0.12520503007108; xu5=-0.17331875341717; xu6=-0.37233460907600;}
  if(freqd==0.26)
  { /* Regressor Matrix solution for freqd=0.26hz */
    xu1=0.07217058501914; xu2=0.11864406779661; xu3=0.13723346090760;
    xu4=0.17441224712958; xu5=0.23018042646255; xu6=0.26735921268453;}
  if(freqd==0.4)
  { /* Regressor Matrix solution for freqd=0.4hz */
    xu1=-0.39475123018043; xu2= -0.11864406779661; xu3=-0.00820120284308;
    xu4=0.21268452706397; xu5=0.54401312192455; xu6=0.76489885183160;}
  if(freqd==0.5)
  { /* Regressor Matrix solution for freqd=0.5hz */
    xu1=-0.72826681246583; xu2=-0.28813559322034; xu3=-0.11208310552214;
    xu4=0.24002186987425; xu5=0.76817933296883; xu6=1.12028430836523;}*/
  /* mapping to the respective stored waveforms */
  map_time1=(freqd/freq1)*direct_learn; /* mapped time for 1st controlinput */
  map_time2=(freqd/freq2)*direct_learn; /* mapped time for 2nd controlinput */
  map_time3=(freqd/freq3)*direct_learn; /* mapped time for 3rd controlinput */
  map_time4=(freqd/freq4)*direct_learn; /* mapped time for 4th controlinput */
  map_time5=(freqd/freq5)*direct_learn; /* mapped time for 5th controlinput */
  map_time6=(freqd/freq6)*direct_learn; /* mapped time for 6th controlinput */
  /* mapping first control input */
  index1=(int)(map_time1);
  fract_check=map_time1-(int)(map_time1);
  if(fract_check==0.0) interpolation=0;
  else interpolation=1;
  if(interpolation==0) control_v1=data1[(int)(map_time1)];
  else if(interpolation==1)
   control_v1=0.5*(data1[(int)(map_time1)]+data1[(int)(map_time1)+1]);
  /* mapping second control input */
  index2=(int)(map_time2);
  fract_check=map_time2-(int)(map_time2);
  if(fract_check==0.0) interpolation=0;
  else interpolation=1;
  if(interpolation==0) control_v2=data2[(int)(map_time2)];
```

```
else if(interpolation==1)
 control_v2=0.5*(data2[(int)(map_time2)]+data2[(int)(map_time2)+1]);
/* mapping third control input */
index3=(int)(map_time3);
fract_check=map_time3-(int)(map_time3);
if(fract_check==0.0) interpolation=0;
else interpolation=1;
if(interpolation==0) control_v3=data3[(int)(map_time3)];
else if(interpolation==1)
 control_v3=0.5*(data3[(int)(map_time3)]+data3[(int)(map_time3)+1]);
/* mapping fourth control input */
index4=(int)(map_time4);
fract_check=map_time4-(int)(map_time4);
if(fract_check==0.0) interpolation=0;
else interpolation=1;
if(interpolation==0) control_v4=data4[(int)(map_time4)];
else if(interpolation==1)
 control_v4=0.5*(data4[(int)(map_time4)]+data4[(int)(map_time4)+1]);
/* mapping fifth control input */
index5=(int)(map_time5);
fract_check=map_time5-(int)(map_time5);
if(fract_check==0.0) interpolation=0;
else interpolation=1;
if(interpolation==0) control_v5=data5[(int)(map_time5)];
else if(interpolation==1)
 control_v5=0.5*(data5[(int)(map_time5)]+data5[(int)(map_time5)+1]);
/* mapping sixth control input */
index6=(int)(map_time6);
fract_check=map_time6-(int)(map_time6);
if(fract_check==0.0) interpolation=0;
else interpolation=1;
```

```
  if(interpolation==0) control_v6=data6[(int)(map_time6)];
  else if(interpolation==1)
   control_v6=0.5*(data6[(int)(map_time6)]+data6[(int)(map_time6)+1]);
  /* computation of the control */
  control_v=xu1*control_v1+xu2*control_v2+xu3*control_v3+xu4*control_v4
           +xu5*control_v5+xu6*control_v6;
  /* learning during the DILC */
  present_error[direct_learn]=pos_err;
  past_input[direct_learn]=control_v;
    direct_learn=direct_learn+1;
  /* move from DILC to normal ILC again swtich off DILC & turn on ILC */
  if(direct_learn==n)
  { direct_learn=0;
    if(DLCILC==0) /* switch back to ILC mode else DLC on all the time */
    { NO_DILC=1; DILC=0;no_learn=0; }
  }
  SMC_control();
  control_vd=control_v;
  control_v=control_v+Uc1;
}


/********************  FINAL CONTROL INPUT ************************/
float final_control(void)
{   Float64 final_voltage;
    pos_err=ref_pos_k-act_pos; /* position error in mm */
    pos_err_um=pos_err*1000;   /* position error in um */
    ref_derivative();          /* call to derivative function */
    if(NO_DILC==1)             /* SMC and ILC if on */
    no_dilc();
    if(DILC==1)                /* SMC and DLCILC if on */
    {present_input=0;
```

```
     dilc();

    }

    if(ref_speed<0 && pos_err_um<-0.1)

       deadvoltage=deadvoltagen; /* deadzone compensation */

    else if(ref_speed>0 && pos_err_um>0.1)

       deadvoltage=deadvoltagep;

    else deadvoltage=0;

    final_voltage=control_v+deadvoltage;

    if (deadvoltage==0) final_voltage=0;

    return(final_voltage);

}

/************************** END **************************************/
```