Lehigh University
Lehigh Preserve

Theses and Dissertations

1996

Repetitive control of a high-speed cam-follower system

Woosuk Chang
Lehigh University

Chang, Woosuk

Repetitive Control of a High-Speed Cam-Follower System

October 13, 1996

Repetitive Control of A High-Speed Cam-Follower System

by

Woosuk Chang

A Thesis

Presented to the Graduate and Research Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Mechanical Engineering

Lehigh University

June, 1996

This thesis is accepted and approved in partial fulfillment of the requirements for the Master of Science.

June 11, 1996

Date

Professor Meng-Sang Chew

Thesis Advisor

Professor Robert P. Wei

Chairperson of Department

Acknowledgment

First of all, I would like to send my deepest gratitude and thanks to my parents, Sun-Duck Chang and Hee-Ja Ahn for their invaluable support and encouragement to me. I would like to express my sincere thanks to my adviser, Professor Meng-Sang Chew, for his instruction and aid throughout this thesis. I also would like to thanks to Professor Richard W. Longman in Columbia University and Professor Minh Q. Phan in Princeton University for their instruction.

I would like to express special thanks to Mr. Soo-Chul Lee for his endless advice and encouragement to my study. I would like to thanks to the members of the Korean Students Club, especially to Do-Hawn Kim and Jin-Hyun Kim (thank you for the correction of my thesis) for their help throughout my study at Lehigh, and Young-Woo Lee, Hwa Sung Kim, Chee Hyun Cho, Young Jin Choi, In Gyu Park and his wife for "being there" every day and night and providing me their shoulder to lean on.

Contents

List of Symbols

$\mathbf{A}$	: Estimation parameter in Discrete Time Domain
$\mathbf{A}_c$	: System Parameter Matrix of State Space Equation in Continuous Time Domain
$\mathbf{A}_d$	: System Parameter Matrix of State Space Equation in Discrete Time Domain
$\underline{\mathbf{A}}$	: System Parameter Matrix in Repetition Domain
a	: A Subscript Representing the Rise Segment
$\mathbf{B}$	: Estimation Parameter in Discrete Time Domain
$\mathbf{B}_c$	: Control Input Matrix of State Space Equation in Continuous Time Domain
$\mathbf{B}_d$	: Control Input Matrix of State Space Equation in Discrete Time Domain
$\underline{\mathbf{B}}$	: Control Input Matrix of System in Repetition Domain
b	: A Subscript Representing the Dwell-1 Segment
$\mathbf{C}$	: Observation Matrix in State Space Equation
c	: Damping Constant
c	: A Subscript Representing the Return Segment
d	: A Subscript Representing the Dwell-2 Segment
$\mathbf{F}_c$	: Nonlinear System Fucntion in Continuous Time Domain
$\mathbf{F}_d$	: Nonlinear System Function in Discrete Time Domain
$\underline{\mathbf{G}}$	: Optimized Repetitive Control Gain in Repetition Domain
H	: Maximum Lift of Output Mass
H_c	: Maximum Lift of Cam
j	: Repetition Variable
k	: Steps in Discrete Time Domain
K_f	: Stiffness of The Follower
K_s	: Stiffness of Return Spring
M	: Mass of the Output
$\underline{\mathbf{Q}}$, $\underline{\mathbf{S}}$	: Weighting Matrices in Repetitive Control
$\underline{\mathbf{R}}$	: Projection Matrix in Adaptive Control
T	: Period of a Repetition in Continuous Time Domain

$\underline{\mathbf{U}}$: Control Input Vector in Repetition Domain

$\mathbf{u}$: Control Input Vector in Discrete Time Domain

$\mathbf{x}$: State Variables in Discrete Time Domain

$\underline{\mathbf{Y}}$: Output Vector in Repetition Domain

$\underline{\mathbf{Y}}^*$: Desired Output Vector in Repetition Domain

$\mathbf{y}$: Output Vector in Discrete Time Domain

y : Displacement of Output

y_c : Lift of the Cam

α, β : Adaptive Factor in the Least-Squares Method

θ_c : Angular Displacement of the Cam

θ : Input Command of the Motor

$\underline{\varphi}$: Input Matrix in System Estimation

$\underline{\phi}$: System Parameter Matrix for System Estimation in Repetition Domain

List of Tables

List of Figures

Abstract

The vibrational problem arising in a high-speed cam-follower machinery is handled with a repetitive control scheme since a cam-follower system repeats the same pattern of motion continuously. Repetitive control theory is an extension of learning control theory in the sense that it makes use of output errors in the previous repetition. It also takes the changes in the state at the beginning of each repetition into consideration. The design of a repetitive controller is achieved by applying adaptive control theory. Also, optimal control theory is added to guarantee stability. The control algorithm is applied to both the linear and the nonlinear portion of the cam-follower system. Due to the strong nonlinearity of the entire cam-follower system, repetitive control has to be modified to handle the tracking problem in a piecewise manner.

Chapter 1

Introduction

This thesis addresses problems concerning the control of a high-speed cam-follower system. A cam-follower system, one of the most widely used mechanisms in production machines, transforms rotational motion to linear back-and-forth motion. When such a system performs a task at high speed, an increase in vibration is caused by several features such as, the eccentric geometry of the cam, resonance between the operation speed and the natural frequency of cam-follower body, the heavy load applied to input and output of cam-follower system, and frame vibration caused by the power source or other external disturbances.

Several investigations of such a high speed cam-follower system problem were made through the analysis of vibrational behavior [1], study of flexibility of the cam and the driving shaft [2,3], characterization of the vibration levels in the system [4], analysis and synthesis of cam contours [5] and development of follower systems [1,6] Recently, with the development of learning control theory by Longman and Phan [7~9], the elimination of vibration in cam-follower system through this control theory was developed by Chew, Juang and Phan [10~12]. However, learning control theory is confined to systems operating in the same environment with zero initial conditions at every repetition. It is not suitable for a system performing continuous tasks with varying initial condition due to vibrations that spill over from the previous repetition. To handle the varying initial condition, this thesis extends the basic theory of learning control through the implementation of an adaptive control scheme [13~16] to control the high-speed cam-follower system that operates continuously for repetitive tasks.

Learning control theory is developed for the systems like cams and robots operating repetitive tasks. These systems will actually result in undesirable output motions if a standard controller is used. The difference between the actual motion and the desired motion is called tracking error. If the same conditions are given to the system for every repetition, a precisely similar pattern of the tracking error will be generated. Such characteristics of a repetitively operated system makes learning control an effective candidate for eliminating the tracking error. Learning control makes extensive use of the knowledge from the history of previous repetitions by measuring the outputs from inputs in discrete time steps. However, since learning control considers only the relationship of the outputs to the inputs, tracking error caused by changes in initial conditions cannot be treated with learning control approach. This implies that the controller must wait until the system return to the same initial condition after a task of a repetition.

In line with such control theory, this thesis utilizes a variation of learning control called repetitive control by introducing the concept of repetition domain [13]. Repetitive control has an advantage over learning control in regards to the application of initial conditions for the system at each repetitions. Repetitive control, which can be applied to any repetitively operated system, is independent of the changes of the initial conditions because it takes the error in initial conditions into consideration. This enables the system to operate repetitively without the wait, unlike learning control. The comparison of repetitive control to learning control is summarized in Table 1.1.

	Repetitive Control	Learning Control
Time Domain	Discrete	Discrete
Repetition to Repetition	Continuous	One repetition process and stop, start another
Initial Condition of a Repetition	Varying	Zero initial condition for all repetitions
Control Gain Matrix	Varying, obtained from system estimation	Constant, initially given
Stability	Guaranteed for linear and slightly nonlinear system	Same as repetitive control, but more stable
Convergence	Marginally converging, slower	Marginally converging, faster

Table 1.1 Comparison of Repetitive Control to Learning Control

Repetitive control procedures are composed of three steps. First, the inputs and the corresponding outputs are measured. Then, relationship between the inputs and the outputs are formulated. This is called identification of the system. The final step is called controller design, with information obtained through identification. Since a suitable controller can only be developed from exact identification, it is necessary to supply various inputs and measurements from corresponding outputs to help the controllers identify the system. Generally, at least 3~5 repetitions are required for the controller to identify the system correctly.

The model considered in this thesis is a single degree-of-freedom, lumped parameter system with an constant sampling time interval, and consists of a motor, a cam, and a follower. As the speed of a cam increases toward the natural frequency of the follower-output system, the intermediate elements between the cam and output behave as a flexible body, inducing residual vibrations even after the input command has stopped.

The role of repetitive control is to analyze the tracking errors, predict the error of the next repetition and correct the input command for the next repetition to reduce the tracking error. Such a procedure is conducted in real-time during the period of actuation of the high-speed cam-follower system. The objective of this paper is to determine the motor input command such that the motion of the output mass trajectory can be controlled exactly as desired.

The basic outline of this thesis is as follows: Chapter 2 describes the mathematical model of a high-speed cam-follower system. The profile of the cam and the desired output path is presented as two different polynomials. In Chapter 3, a state space expression of cam-follower system in discrete time in the repetition domain. An adaptive control algorithm is introduced in Chapter 4 as the method for the identification of the system. Also, the repetitive control scheme is presented and its simulations of the control with and without disturbances are also presented. Nonlinearity occurs due to the geometry of the cam in the relationship between cam rotation and its follower. Repetitive control methods can not be directly applied to the entire cam-follower system owing to this kinematic nonlinearity. In order to apply repetitive control method to handle such a nonlinearity, the system is analyzed piecewise and repetitive control is applied to each of the segments. This method, called piecewise repetitive control, is discussed in Chapter 5.

Chapter 2

Modeling of Cam-Follower System

Mathematical modeling of the cam lift and desired output motion in the form of polynomials are presented in this chapter along with the analysis of dynamic motions of the follower system and the entire system. The motion of the follower system is expressed as a linear differential equation, and that of the entire system as a nonlinear differential equation.

Most cam-follower system models are made up of three basic components: the driver (for this study, a direct current(DC) motor with proportional-derivative(PD) controller is used as a driver), a cam, and a follower as shown in Fig.2.1. The DC motor (1), including a gear box, connects to a cam (3) through the cam shaft (2). The cam contacts with a follower (4), which is connected to the output mass (6) through the flexible link (5). We assume there is no friction between cam and follower. The output mass is connected to the ground through a spring (8) and a damper (7). The modeling of cam-follower-system is made under the following assumptions:

Assumption 1. The system is a single degree-of-freedom system with a single-input and a single-output. Input is given as the angular displacement of motor command to drive in entire system. Output is the displacement of the output mass.

Assumption 2. The flexible link connected output mass is assumed flexible with

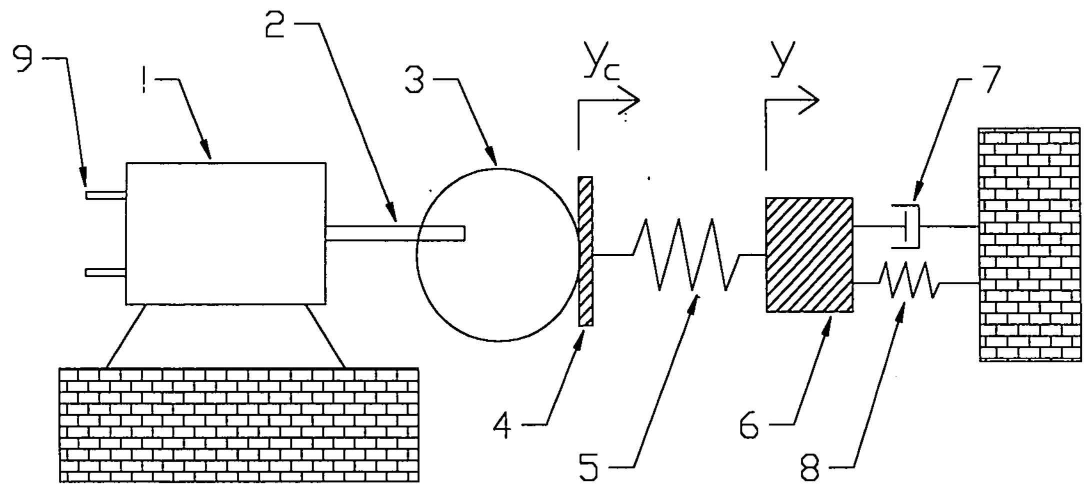


Figure 2.1 Diagram of the Cam-Follower Nonlinear System

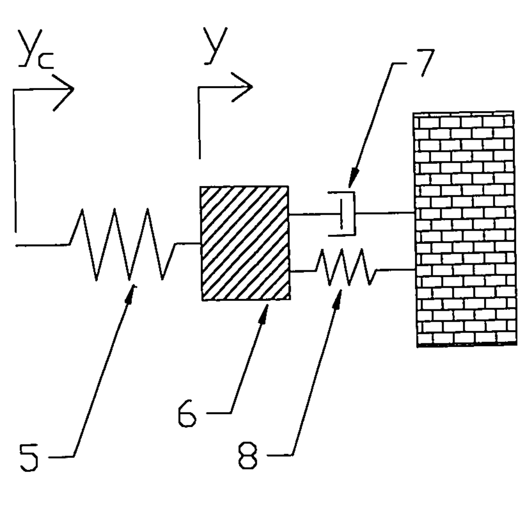


Figure 2.2 Diagram of the Follower-Output Linear Subsystem

No.1	Motor with Gear			
	K_m	:	motor torque constant	0.023 N-m/amp
	K_b	:	motor back emf	0.0318 V-s/rad
	R_m	:	amateur resistance	3.7 Ohm
	J_m	:	amateur & gear inertia	4.0 x10^{-6} N-ms^2
	J_c	:	cam inertia	1.0 x10^{-6} N-ms^2
	N_f	:	gear ratio	0.1
	L	:	amateur impedance	10 mh (neglected)
	K_p	:	proportional controller gain	1.0x10^5
	K_d	:	derivative controller gain	10.0
No.2	Cam Shaft (regarded as rigid)			
	* θ_c	:	angular displacement	rad
No.3	Cam			
	H_c	:	maximum lift	0.01 m
	* y_c	:	lift of the cam	m
No.4	Follower			
	M_f	:	follower mass	0.025 Kg
No.5	Flexible Link			
	K_f	:	shaft stiffness	300 N/m
No.6	Output Mass			
	M	:	mass of output mass	0.1 Kg
	* y	:	output mass displacement	m
No.7	Damper			
	c	:	damping constant	0.5 Kg/s
No.8	Return Spring			
	K_s	:	return spring stiffness	15 N/m
No.9	Control Input			
	* θ	:	angular input command	rad

* indicates the system variable

Table 2.1 Cam-Follower System Parameters & Variables

stiffness K_f. Other elements, such as a cam shaft and a power source system are assumed sufficiently stiff enough to be regarded as rigid. Thus any deviation of output motion from ideal will be due to the flexibility of the flexible link.

Assumption 3. The effect of components weight on cam vibration is neglected, and static preload of the return spring S_p is constant.

Assumption 4. Open track cam with flat follower surface shown in Fig.2.3 is used. The net lift of the cam ($y_c = D - R_{min}$) is a function of angular displacement of the cam θ_c and modeled as a 3rd degree polynomial

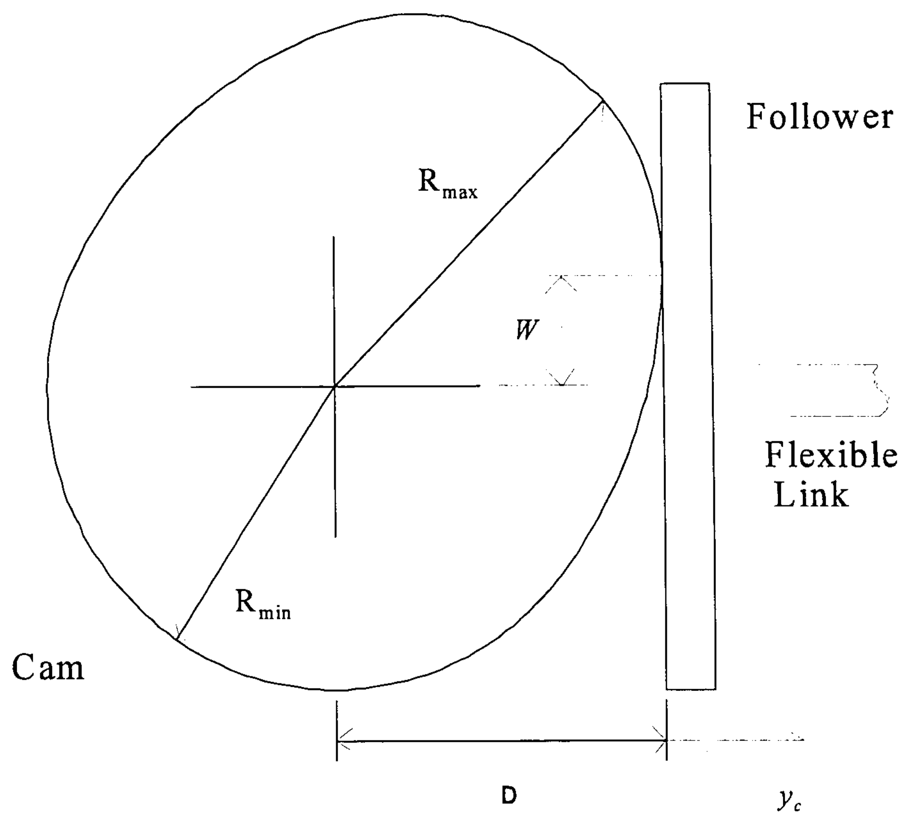
$$y_c/H_c = a_o + a_1\theta_c + a_2\theta_c^2 + a_3\theta_c^3 \tag{2.1}$$

where H_c is the maximum net lift of the cam, D is horizontal distance from cam axis to the follower and R_{min} is the minimum radius of the cam.

Assumption 5. The desired output trajectory y^* is function of θ_c and is selected as a 5th degree polynomial,

$$y^*/H = b_o + b_1\theta_c + b_2\theta_c^2 + b_3\theta_c^3 + b_4\theta_c^4 + b_5\theta_c^5 \tag{2.2}$$

where H is the maximum displacement of output mass.



Cam Lift $y_c = D - R_{min}$

Distance from the Cam Center to the Force Action Line

$$W = \frac{dy_c}{d\theta_c}$$

Figure 2.3 Diagram of the Cam and Its Follower

2.1 Trajectories of Cam and Desired Output

Both cam and desired trajectories are composed of four segments, such as Rise, Dwell-1, Return and Dwell-2. shown as Fig.2.4. The rise segment begins from 0° marked 'a' and finishes at 90° (point 'b') of cam angle θ_c. Then the dwell-1 segment starts from 'b' and continues for 36°. The return segment starts at 126°(point 'c') and ends at 216° (point 'd'). Dwell-2 segment starts from the point 'd' and ends at 360°.

2.1.1 Polynomial of Cam Lift Trajectory

Modeling of the cam lift can be achieved by assigning appropriate values to the unknown coefficients in Eq.(2.1) for each segments. Four unknowns a_0 ... a_3 are found from the following boundary conditions specified at both ends of each of the segments

$$\text{at } \theta_c = 0°(360°): \quad y_c = 0, \quad dy_c/d\theta_c = 0 \tag{2.2a}$$

$$\text{at } \theta_c = 90° \quad : \quad y_c = H_c, \quad dy_c/d\theta_c = 0 \tag{2.2b}$$

$$\text{at } \theta_c = 126° \quad : \quad y_c = H_c, \quad dy_c/d\theta_c = 0 \tag{2.2c}$$

$$\text{at } \theta_c = 216° \quad : \quad y_c = 0, \quad dy_c/d\theta_c = 0 \tag{2.2d}$$

where H_c is the maximum lift of the cam. With the first two conditions, the four unknown coefficients in Eq.(2.1) can be determined for the 'rise' segments as $a_0 = a_1$=0, a_2 = -0.5160 and a_3=1.2158.

Repeating this procedure for the other segments, the polynomials for the cam lift trajectory is determined. The plot of this trajectory is shown in Fig.2.4. The result of the third degree polynomial for the cam lift trajectory is as follows :

Rise : $y_c = Hc\,(-0.5160\,\theta_c^2 + 1.2158\theta_c^3)$ (2.3a)

Dwell-1 : $y_c = Hc$ (2.3b)

Return : $y_c = Hc\,(-10.3680 + 12.8342\theta_c - 4.6202\theta_c^2 + 0.5160\theta_c^3)$ (2.3c)

Dwell-2 : $y_c = 0$ (2.3d)

2.1.2 Polynomial of Desired Output Trajectory

The polynomial of desired output motion is chosen such that the output mass tracks a smoother path than that specified by the cam surface. This polynomial is extended up to the 5th order given in Eq.(2.2). The boundary conditions for each ends of the segments are :

at $\theta_c = 0°(360°)$: $y^* = 0,\ dy^*/d\theta_c = 0,\ d^2y^*/d\theta_c^2 = 0$ (2.4a)

at $\theta_c = 90°$: $y^* = H,\ dy^*/d\theta_c = 0,\ d^2y^*/d\theta_c^2 = 0$ (2.4b)

at $\theta_c = 126°$: $y^* = H,\ dy^*/d\theta_c = 0,\ d^2y^*/d\theta_c^2 = 0$ (2.4c)

at $\theta_c = 216°$: $y^* = 0,\ dy^*/d\theta_c = 0,\ d^2y^*/d\theta_c^2 = 0$ (2.4d)

where H is the maximum static displacement of the output mass obtained from the relationship between the stiffness of the follower shaft and that of return spring, as

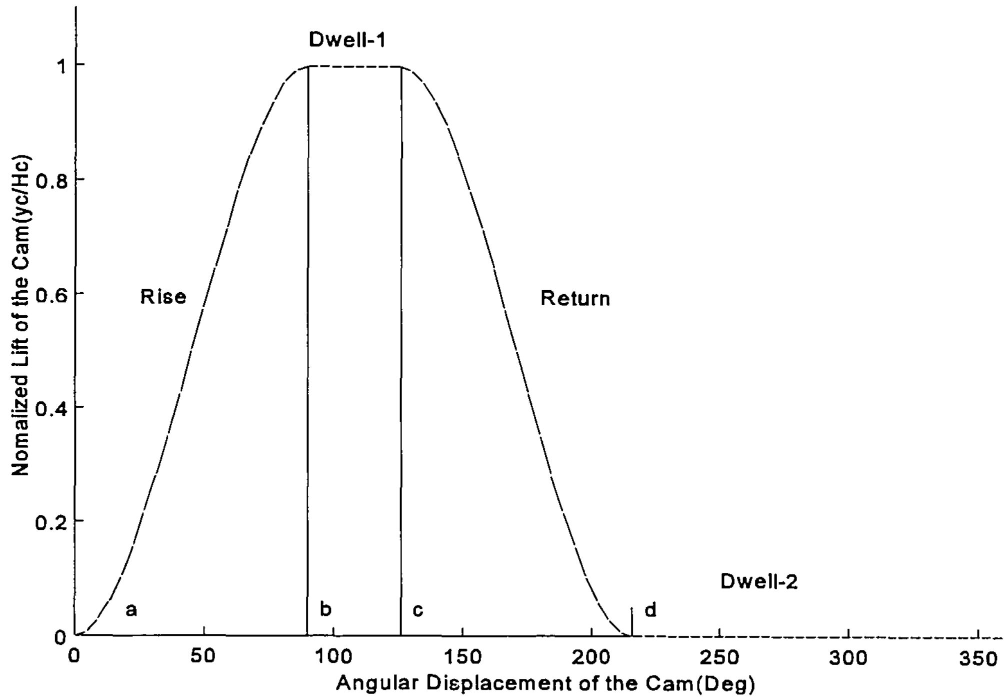


Figure 2.4 Cam Lift Trajectory (3rd Order Polynomials)

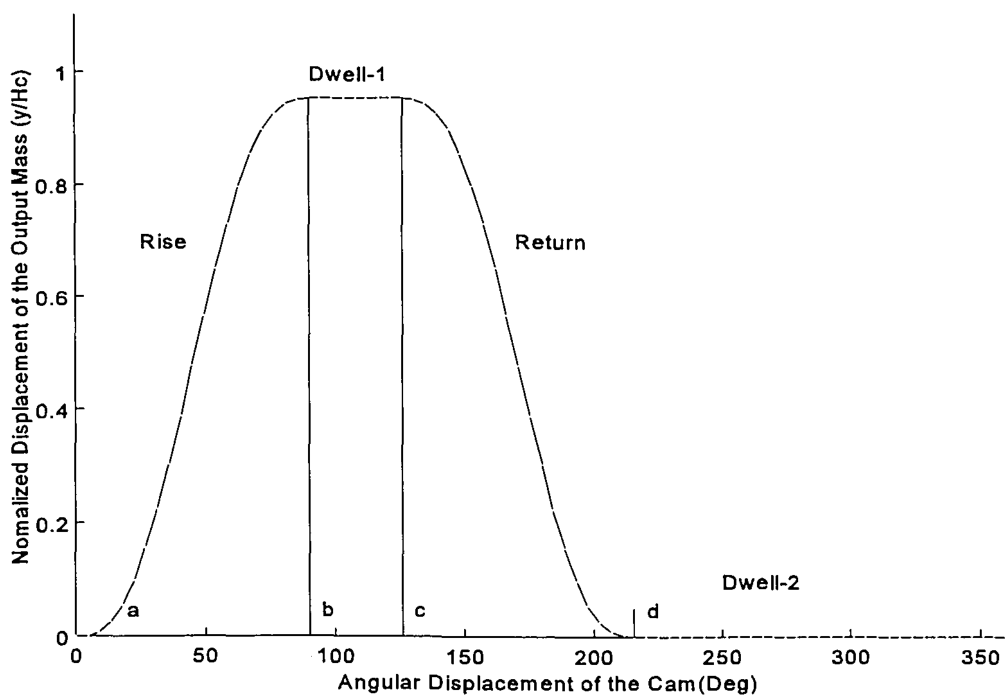


Figure 2.5 The Desired Output Trajectory (5th Order Polynomials)

$$H = \frac{K_f}{K_f + K_s} H_c \tag{2.5}$$

where K_f and K_s represent the stiffness of the flexible link and return spring, respectively and H_c is the maximum net lift of the cam.

Using the first six conditions for both ends of "rise" segment, six unknown coefficients are obtained as $b_0 = b_1 = b_2 = 0$, $b_3 = 2.5801$, $b_4 = 2.4648$ and $b_5 = 0.6274$. Using proper conditions for boundaries of each segment, the polynomials for the desired output path are found. The results of 5th degree polynomial for the output displacement in each segments are as follows.

Rise : $y^* = H(2.5801\theta_c^3 - 2.4638\theta_c^4 + 0.6274\theta_c^5)$ (2.6a)

Dwell-1 : $y^* = H$ (2.6b)

Return : $y^* = H(118.3335 - 215.6156\theta_c + 155.2404\theta_c^2 - 54.5954\theta_c^3 + 9.3625\theta_c^4 - 0.6274\theta_c^5)$ (2.6c)

Dwell-2 : $y^* = 0$ (2.6d)

The plot of this polynomial is shown in Fig.2.5.

2.2 Dynamics of Cam-Follower System

Two models from the cam-follower system are analyzed in this chapter. First, the equation of motion of the follower system composed of a flexible link and an output mass including a returning spring and a damper is presented. Next, dynamic equations of the entire cam-follower system will be derived. The nonlinearity problem arises when expressing the net cam lift y_c in terms of cam angular displacement θ_c. This kinematic nonlinearity causes the entire dynamic system to become highly nonlinear.

2.2.1 Dynamic Equation of Linear Follower-Output Subsystem

First, consider the follower system composed of the follower, the flexible link and the output mass with a springs and a damper as shown in Fig.2.2. Cam lift y_c is regarded as the system input and displacement y of output mass is the system output. According to Newton's second law [17], dynamics of the linear follower-output subsystem can be expressed as

$$M\frac{d^2y}{dt^2} + c\frac{dy}{dt} + (K_f + K_s)y = K_f y_c \tag{2.7}$$

where M is output mass, c is return spring damper, K_s is return spring constant and K_f is cam shaft stiffness. Eq.(2.7) can be expressed in state space equation [18] as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A_c x}(t) + \mathbf{B_c u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\,\mathbf{x}(t) \end{aligned} \tag{2.8}$$

where the state variables $\mathbf{x}(t) = [\, y(t) \;\; \dot{y}(t)]^{\mathrm{T}}$, the system input $\mathbf{u} = [y_c(t)]$, and the matrix $\mathbf{C} = [1 \;\; 0]$ is the observation matrix. Since the system is assumed to be single degree of freedom, the input $\mathbf{u}$ and the output $\mathbf{y}$ are scalar and state variable $\mathbf{x}$ is a column vector of dimension (2x1). The system parameter matrices are

$$\mathbf{A_c} = \begin{bmatrix} 0 & 1 \\ -\frac{K_s+K_f}{M} & -\frac{c}{M} \end{bmatrix}, \mathbf{B_c} = \begin{bmatrix} 0 & \frac{K_f}{M} \end{bmatrix}^T \tag{2.9}$$

where the subscript 'c' of the parameter matrices $\mathbf{A}_c$, $\mathbf{B}_c$ means that these matrices represent the parameters for the system in a continuous state space formulation.

2.2.2 Dynamic Equation of Cam-Follower System

The analysis of motion of the entire system shown in Fig.2.1 includes the relationship of cam lift to cam angle and analysis of the DC-motor. The DC-motor is equipped with a PD feedback controller with a proportional gain K_p and a derivative gain K_d. A gear box with gear ratio N_g is connected to the motor to reduce the system driving speed. A total of four equations is required to describe this dynamic system. In this system, the lift of the cam y_c is a function of a state variable θ_c. The system input is the motor command input θ and output is the displacement of output follower y. We begin the analysis with a mathematical modeling of the DC motor. The voltage drop across the motor is

$$e_m = L\frac{di}{dt} + R_m i + K_b\dot{\theta}_m = K_p(\theta - \theta_m) + K_d(\dot{\theta} - \dot{\theta}_m) \tag{2.10}$$

where θ and θ_m are the motor input command and the motor output angular displacement respectively. K_p and K_d are the proportional and derivative gains respectively of the DC-motor controller. R_m is resistance of the motor. Values of these parameters are given in Table2.2. There is a gear box connecting the motor shaft to the cam shaft to increase the torque to the cam. The relationship of the angular displacement of the cam to that of the motor output shaft is given by

$$N_g\theta_m = \theta_c \tag{2.11}$$

where θ_m is angular displacement of the motor output shaft and θ_c is that of the cam shaft. The amateur impedance L is small and can be neglected. Torque output from the DC-motor to the system is given by

$$K_m i = (J_m + J_c N_g^2)\ddot{\theta}_m + N_g T_c \tag{2.12}$$

where K_m is the motor constant. J_m and J_c are the inertias of amateur and the cam respectively. The offset distance W from the rotational axis of the cam to the line of action of the cam-follower contact force vector is $dy_c / d\theta_c$ as illustrated in Fig.2.2. T_c is the net torque that drives the linear system, and is given by

$$T_c = \frac{dy_c}{d\theta_c}\left(M_f \ddot{y}_c + K_f(y_c - y) + S_p\right) \tag{2.13}$$

where M_f is the mass of the follower and S_p is the preload on the cam. Assume a new operator representing the derivatives about θ_c

$$f' = \frac{df}{d\theta_c} \ , \ f'' = \frac{d^2 f}{d\theta_c^2} \tag{2.14}$$

According to the chain role, the second derivative of y_c about time can be expressed as the derivatives about θ_c as

$$\ddot{y}_c = \frac{d^2 y_c}{dt^2} = \frac{d}{dt}\left(\frac{dy_c}{d\theta_c}\frac{d\theta_c}{dt}\right) = y_c''\dot{\theta}_c^2 + y_c'\ddot{\theta}_c \tag{2.15}$$

So the Eq.(2.13) can be written as

$$T_c = y_c'(M_f y''\dot{\theta}_c^2 + K_f(y_c - y) + S_p) + M_f y_c'^2 \ddot{\theta}_c \tag{2.16}$$

where y_c is function of only θ_c given by Eq.(2.3). Relationship between follower and output mass is given in Eq.(2.8) as

$$M\ddot{y} + c\dot{y} + (K_f + K_s)y = K_f y_c \tag{2.17}$$

Equations from Eq.(2.14) through Eq.(2.17) governs the dynamic motions of the motor driven cam-follower system and will be used in the simulation for the piecewise repetitive controls in Chapter 5. These equations are combined and expressed in nonlinear state space form

$$\dot{\mathbf{x}} = \mathbf{F}_c(\mathbf{x}, \mathbf{u}) \tag{2.18}$$

where $\mathbf{x} = [\ \theta_c \ \ \dot{\theta}_c \ \ y_c \ \ \dot{y}_c\ ]^T$ and $\mathbf{u} = [\ \theta \ \ \dot{\theta}\ ]^T$ (2.19)

The system function is

$$\mathbf{F}_c(\mathbf{x},\mathbf{u}) = [f_1 \;\; f_2 \;\; f_3 \;\; f_4]^T \tag{2.20}$$

where

$$
\begin{aligned}
f_1 &= \dot{\theta}_c \\
f_2 &= \{K_p(N_g\theta - \theta_c) + K_d(N_g\dot{\theta} - \dot{\theta}_c) - K_b\dot{\theta}_c - R_oT_c\}/J_o \\
f_3 &= \dot{y} \\
f_4 &= \{K_f y_c - c\dot{y} - (K_s + K_f)y\}/M
\end{aligned}
$$

with $J_o = R_m(J_m + J_cN_g^2 + My_c')\,/\,K_m$, $R_o = R_mN_g^2/K_m$, and T_c is given by Eq.(2.16). The subscript "c" in $\mathbf{F}_c$ means above state space equations are described in continuous time. Conversion of these continuous linear and nonlinear systems to the corresponding discrete time format will be treated in the next chapter.

Chapter 3

System Equation in the Discrete Time Domain and Estimation Parameters in the Repetition Domain

The repetitive control formulation will be cast in a discrete-time periodic system model, where inputs and outputs are applied and measured at equally spaced sampling times. To understand this characteristics of a discrete system model, the concept of discrete time domain is introduced where the state of all the system variables are sampled at the sampling time steps spaced with constant intervals.

The concept of repetition domain [13] is also introduced to help the design of the repetitive controller. In the repetition domain, inputs and outputs of all steps in a repetition are packed into a corresponding single column matrix, yielding a system equation for a given repetition. With this system equation expressed in the repetition domain, estimation of the system parameters and design of the controller can be achieved.

3.1 State Space Equations in Discrete-Time Domain

Discrete time domain is one where the time variable t is defined only when it is a multiple of a sampling time interval Δt. The time variable is expressed by integer steps instead of continuous time in this domain. On the other hand, in the continuous time domain, time variable t is expressed as a real number and the state variables are available at any instant of time t. Comparison of continuous time domain and discrete time domain is given in Table 3.1.

Since the repetitive control theory was developed for a discrete-time system, it is necessary to convert the continuous-time state space equation, Eq.(2.8) and Eq.(2.20), to the corresponding discrete-time equations in discrete time domain. Before carrying out this conversion, the following assumptions need to be made:

Assumption 1. Sampling time interval Δt is constant.

Assumption 2. Input applications and output measurements are made only after each sampling time.

Assumption 3. Period T of the Rise-Dwell-1-Return-Dwell-2 motion of the cam-follower system, is divided into p (integer) steps where p is chosen sufficiently large enough so that the discrete system equation represent output motion close to that of continuous equation.

Assumption 4. In discrete time domain, state variable $\mathbf{x}(t)$ is expressed as $\mathbf{x}(k)$ where

$$t = k\Delta t \qquad (k = 0, 1, 2, \dots) \tag{3.1}$$

Assumption 5. A system equation in the discrete time domain is expressed as

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k)+\mathbf{B}_d\mathbf{u}(k) \qquad \text{linear system} \tag{3.2a}$$

$$\mathbf{x}(k+1) = \mathbf{F}_d(\mathbf{x}(k), \mathbf{u}(k)) \qquad \text{nonlinear system} \tag{3.2b}$$

Assumption 6. The time rate change of input $(d\mathbf{u}/dt)_{t=k\Delta t}$ is zero between the sampling time $k\Delta t$ and $(k+1)\Delta t$, and the input command $\mathbf{u}(k)$ is constant over the sampling interval.

	Discrete-Time Domain	Continuous-Time Domain
Time Variables	k(steps) (k=0, 1,2, ...)	t(second) (t>0, real number)
State Variables	$\mathbf{x}(k)$	$\mathbf{x}(t)$
Period	p (steps) (p integer)	T(second) (T>0, real number)
State Space Eq.s	$\mathbf{x}(k+1) = \mathbf{F}_d(\mathbf{x}(k),\mathbf{u}(k))$	$\mathbf{x}(t) = \mathbf{F}_c(\mathbf{x}(t),\mathbf{u}(t))$

Table 3.1 Continuous Time Domain and Discrete Time Domain

The conversion of a continuous linear system equation to a discrete-time system [19,20] will now be made using the assumptions above. Consider the continuous-time state space equation of the follower-output subsystem as

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c\,\mathbf{u}(t) \qquad (3.3)$$
$$\mathbf{y}(t) = \mathrm{C}\mathbf{x}(\mathrm{t})$$

where $\mathbf{A}_c = \begin{bmatrix} 0 & 1 \\ -\frac{K_s+K_f}{M} & -\frac{c}{M} \end{bmatrix}$, $\mathbf{B}_c = \begin{bmatrix} 0 \\ \frac{K_f}{M} \end{bmatrix}$ and $\mathbf{C} = [1\ 0]$. $\mathbf{A}_c$, $\mathbf{B}_c$ are system parameter matrices, and $\mathbf{C}$ is the system observation matrix. To obtain the discrete-time system equation, integrate Eq.(3.3) with respect to time t from $k\Delta t$ to $(k+1)\Delta t$ as

$$\mathrm{x}((k+1)\Delta t) = e^{\mathbf{A}_c\Delta t}\mathbf{x}(k\Delta t) + \int_{\tau=0}^{\Delta t} e^{\mathbf{A}_c\tau}\mathbf{B}_c\mathbf{u}(\tau)d\tau \qquad (3.4)$$

In the integration of Eq.(3.4), $\mathbf{u}(\tau)$ remains constant during the integration, in accordance to the assumption 6. Thus, Eq.(3.4) can be written as

$$\mathbf{x}((k+1)\Delta t) = e^{\mathbf{A}_c\Delta t}\mathbf{x}(k\Delta t) + \int_{\tau=0}^{\Delta t} e^{\mathbf{A}_c\tau}\mathbf{B}_c d\tau\, \mathbf{u}(k\Delta t) \tag{3.5}$$

Eq.(3.5) yields a discrete-time state space equation of the follower-output system in the discrete time domain as

$$\mathbf{x}(k+1) = \mathbf{A}_d\mathbf{x}(k) + \mathbf{B}_d\mathbf{u}(k) \tag{3.6}$$

$$\mathbf{y}(k+1) = \mathbf{C}\mathbf{x}(k+1)$$

where $\mathbf{A}_d = e^{\mathbf{A}_c\Delta t}$, $\mathbf{B}_d = \int_{\tau=0}^{\Delta t} e^{\mathbf{A}_c\tau}\mathbf{B}_c d\tau$ and $\mathbf{C} = [1\ 0]$. (3.7)

Next, consider the conversion of the nonlinear continuous-time system given by Eq.(2.18) through Eq.(2.20) to discrete-time system. Due to the nonlinear relationship between the cam lift and the angular displacement of the cam, there is no analytic approach to convert it from continuous-time to the discrete-time. Integrate Eq.(2.18) with respect to time t from $k\Delta t$ to $(k+1)t$, assuming that the time rate change of the input $(d\mathbf{u}/dt)_{t=k\Delta t}=0$ and $\mathbf{u}(\tau)$ is constant over the integration range (Assumption 6). The integration gives the discrete-time solution to the nonlinear system equation expressed as

$$\mathbf{x}(k+1) = \mathbf{F}_d(\mathbf{x}(k),\mathbf{u}(k)) \tag{3.8}$$

$$\mathbf{y}(k+1) = \mathbf{C}\mathbf{x}(k+1)$$

Since the integration of the nonlinear equation is extremely difficult to carry out analytically, the numerical integration was made by Runge-Kutta 4th and 5th order method.

Note that these linear and nonlinear system equations Eq.(3.6) and Eq.(3.7) will be used only for the measure the output as a response of given input command, not for design of controller.

3.2 Estimated System Parameter Equations in Repetition Domain

Since the repetitive controller is obtained from the estimated linear system matrices in the repetition domain, not from the actual parameters, the formulation of the system parameter matrices to be estimated are provided. Consider an arbitrary discrete-time state space equation with constant sampling time interval Δt, given as

$$\mathbf{x}(i+1) = \mathbf{A}\mathbf{x}(i)+\mathbf{B}\mathbf{u}(i) \qquad (3.9)$$

$$\mathbf{y}(i) = \mathbf{C}\mathbf{x}(i)$$

where $\mathbf{x}\in R^n$, $\mathbf{y}\in R^q$, $\mathbf{u}\in R^m$, and this system are assumed periodic with a period consisting of p time steps. The parameters **A**, **B**, **C** are to be estimated in the next chapter and assumed unknown at this point. For i = 0,.1,.2,..p-1, the solution to Eq.(3.9) is

$$\mathbf{y}(i+1) = \mathbf{C}\mathbf{A}^i\mathbf{x}(0) + \sum_{\tau=0}^{i} \mathbf{C}\mathbf{A}^{i-\tau}\mathbf{B}\mathbf{u}(\tau) \qquad (i = 0,1,\ldots,p-1) \qquad (3.10)$$

Introducing a repetition variable j and a new time step variable k and $i = jp+k$, the general solution to Eq.(3.9) at the jth repetition can be expressed as

$$\mathbf{y}(i+1) = \mathbf{CA}^{i-jp}\mathbf{x}(jp) + \sum_{\tau=0}^{i-jp} \mathbf{CA}^{i-\tau-jp}\,\mathbf{Bu}(\tau+jp) \quad (k = 0,1, ..., p\text{-}1) \tag{3.11}$$

For $i=jp+k$; $j=0, 1,2, ...$; $k = 0, 1, ..., p$-1; all the state variables at any time step i in the jth repetition can be written as

$$\mathbf{x}(jp+k) = \mathbf{x}_j(k),\ \mathbf{u}(jp+k) = \mathbf{u}_j(k) \text{ and } \mathbf{y}(jp+k) = \mathbf{y}_j(k) \tag{3.12}$$

By making all the changes of variables from i to j and k, we obtain the following description of the repetitive process in the repetition domain

$$\underline{\mathbf{Y}}_j = \underline{\mathbf{A}}_j\mathbf{x}_j(0) + \underline{\mathbf{B}}_j\,\underline{\mathbf{U}}_j. \tag{3.13}$$

where

$$\underline{\mathbf{Y}}_j = [\mathbf{y}_j^{\mathrm{T}}(1)\ \mathbf{y}_j^{\mathrm{T}}(2)\ ...\ \mathbf{y}_j^{\mathrm{T}}(p)]^{\mathrm{T}}, \quad \underline{\mathbf{U}}_j = [\mathbf{u}_j^{\mathrm{T}}(0)\ \mathbf{u}_j^{\mathrm{T}}(1)\ ...\ \mathbf{u}_j^{\mathrm{T}}(p\text{-}1)]^{\mathrm{T}}$$

and

$$\underline{\mathbf{A}}_j = [(\mathbf{CA}_j)^{\mathrm{T}}\ \ (\mathbf{CA}_j^2)^{\mathrm{T}}\ \ ...\ \ (\mathbf{CA}_j^p)^{\mathrm{T}}\ ]^{\mathrm{T}}$$

$$\underline{\mathbf{B}}_j = \begin{bmatrix} \mathbf{CB} & 0 & \cdots & 0 \\ \mathbf{CAB} & \mathbf{CB} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{p-1}\mathbf{B} & \mathbf{CA}^{p-2}B & \cdots & \mathbf{CB} \end{bmatrix}$$

Eq.(3.13) can be rewritten in a simpler form by defining an augmented parameter and input variation matrices, namely

$$\underline{\phi}_j = [\, \underline{\mathbf{A}}_j \, \underline{\mathbf{B}}_j \,]$$

$$\underline{\varphi}_j = [\, \mathbf{x}_j^{\mathrm{T}}(0) \;\; \mathbf{u}_j^{\mathrm{T}}(0) \; \mathbf{u}_j^{\mathrm{T}}(1) \ldots \mathbf{u}_j^{\mathrm{T}}(p\text{-}1) \,]^{\mathrm{T}}$$

Thus the system equation in repetition domain becomes

$$\underline{\mathbf{Y}}_j = \underline{\phi}_j \; \underline{\varphi}_j \tag{3.14}$$

where $\underline{\phi}_j$ is a system parameter matrix to be estimated from the identification and $\underline{\varphi}_j$ is an input column vector in the repetition domain.

Chapter 4

Repetitive Control of a Linear System

Repetitive control scheme is readily applicable to the follower-output system which is a linear dynamic system. This system does not includes the cam and DC-motor. Without disturbances, this subsystem has a perfect linear relationship between the input and output since the static displacement of the input is always proportional to that of the corresponding output. If the relationship is approximately proportional, it can be said that the system is slightly nonlinear. The periodic disturbances presented to the linear system result in the slight nonlinearity of the system..

When repetitive control is involved with a linear, or slightly nonlinear system, stability of the control process is guaranteed despite the estimated system parameters are different from actual system parameters [13]. Also, optimal control [8,21] is employed to ensure stability and to avoid excessively large input commands to the system. A general procedure of repetitive control is given as follows :

Step 1. Initialize estimation parameters and desired output variables given in Eq.(2.4) in matrix format in the repetition domain.

Step 2. Apply an input command step-by-step in discrete time domain and measure the corresponding output during a given period using the linear system equation given by Eq.(3.6)

Step 3. Estimate the system parameter matrices from the measurements for a repetition using the adaptive control method explained in Section 4.1

Step 4. Design a controller with estimated matrices from Step 3, using the optimized control gain matrix.

Step 5. Update new input command for the next repetition. The control procedure starts from Step 2 to Step 5. Repeat until the vibration in the system is reduced acceptably.

System estimation is carried out through adaptive control theory based on a least-squares algorithm. Note that the controller is designed from the estimated parameter matrix $\underline{\phi}_j$ which is obtained only from the measurement between the input and the output in the repetition domain. No information about the actual system is included in the design of the controller. The mathematical modeling of the system in Section 2.2 is presented only to simulate the system and to obtain the output response from the given input command, not to design the controller.

4.1 Adaptive Control Scheme for Parameter Estimation

For a system equation in the jth repetition, assume that one knows the values in the input vector $\underline{\mathbf{U}}_j$, the initial condition $\mathbf{x}_j(0)$ and corresponding output vector $\underline{\mathbf{Y}}_j$ from the measurements during the jth repetition. From Eq.(3.13), we have

$$\underline{\mathbf{Y}}_j = \underline{\mathbf{A}}_j \mathbf{x}_j(0) + \underline{\mathbf{B}}_j \underline{\mathbf{U}}_j . \tag{4.1}$$

where $\underline{\mathbf{Y}}_j = [\mathbf{y}_j^T(1)\ \mathbf{y}_j^T(2)\ \ldots\ \mathbf{y}_j^T(p)]^T$, $\underline{\mathbf{U}}_j = [\mathbf{u}_j^T(0)\ \mathbf{u}_j^T(1)\ \ldots\ \mathbf{u}_j^T(p\text{-}1)]^T$ and parameter matrices $\underline{\mathbf{A}}_j$, $\underline{\mathbf{B}}_j$ are unknown. Through an adaptive control scheme, these unknown parameters will be estimated.

Eq.(4.1) is rewritten with a newly defined input vector and a parameter matrix as shown in Eq.(3.14)

$$\underline{\mathbf{Y}}_j = \underline{\phi}_j\ \underline{\varphi}_j \tag{4.2}$$

where
$$\underline{\varphi}_j = [\ \mathbf{x}_j^T(0)\ \ \underline{\mathbf{U}}_j^T\ ]^T \tag{4.2a}$$

$$\underline{\phi}_j = [\ \underline{\mathbf{A}}_j \underline{\mathbf{B}}_j\ ] \tag{4.2c}$$

Once the initial estimation of the first system parameter matrix $\underline{\phi}_0$ is provided, the parameter estimation of $\underline{\phi}_j$ at the jth repetition can be updated according to the following rule, Ref.[6]

$$\underline{\phi}_j = \underline{\phi}_{j-1} + a_j \frac{(\underline{\mathbf{Y}}_j - \underline{\phi}_{j-1}\ \underline{\varphi}_j\ )\ \underline{\varphi}_j^T \underline{\boldsymbol{R}}_j}{1 + \underline{\varphi}_j^T \underline{\mathbf{R}}_j\ \underline{\varphi}_j} \qquad (j\text{=}1,2,\ldots) \tag{4.3}$$

where a_j is adaptation factor taken to be $a_j > 0$, and the projection matrix $\underline{\boldsymbol{R}}_j$ is updated according to the rule

$$\underline{\boldsymbol{R}}_j = \underline{\boldsymbol{R}}_{j-1} - \frac{\underline{\boldsymbol{R}}_{j-1}\ \underline{\varphi}_j^T\ \underline{\varphi}_j \underline{\boldsymbol{R}}_{j-1}}{1 + \underline{\varphi}_j^T \underline{\boldsymbol{R}}_{j-1}\ \underline{\varphi}_j^T} \qquad (j\text{=}1,2,\ldots) \tag{4.4}$$

where $\underline{\boldsymbol{R}}_0 = \alpha \boldsymbol{I}_{(q\mathrm{x}q)}$, $q = p+n$, $\alpha > 0$. As the repetition progresses, the estimated parameter matrix converges to a constant matrix as long as the measured system is linear. The 1 in the denominator in Eq.(4.3) and Eq.(4.4) is intended to prevent division by zero. In practice, any positive number may be used.

4.2 Repetitive Controller

Assume that the desired system output response is given by the output history $\underline{\mathbf{Y}}^*$ for p steps over the course of a repetition. As repetition increases, estimated values in matrices $\underline{\mathbf{A}}_j$, $\underline{\mathbf{B}}_j$ marginally converge to constant values as long as the system itself is linear and desired output is feasible, i.e. control input vector $\underline{\mathbf{U}}_j^*$ exists which produces the desired output motion $\underline{\mathbf{Y}}^*$. If there is the same number of sensors as actuators (the estimated system parameter $\underline{\mathbf{B}}_j$ is square) then the output desired vector $\underline{\mathbf{Y}}^*$ is feasible and is independent of the initial condition $\mathbf{x}_j(0)$, the control input $\underline{\mathbf{U}}_j^*$ that produces $\underline{\mathbf{Y}}^*$ is unique. The proof is given in [7]. Compare the two system equations of two consecutive repetitions as

$$\underline{\mathbf{Y}}_{j+1} = \underline{\mathbf{A}}_{j+1}\, \mathbf{x}_{j+1}(0) + \underline{\mathbf{B}}_{j+1}\, \underline{\mathbf{U}}_{j+1} \tag{4.5}$$

$$\underline{\mathbf{Y}}_j = \underline{\mathbf{A}}_j \mathbf{x}_j(0) + \underline{\mathbf{B}}_j\, \underline{\mathbf{U}}_j$$

Assuming that the output vector of the (j+1)th system equation has the desired ideal values, $\underline{\mathbf{Y}}_{j+1} = \underline{\mathbf{Y}}^*$ and that the estimated system parameters have converged so that $\underline{\mathbf{A}}_{j+1} \approx \underline{\mathbf{A}}_j$ and $\underline{\mathbf{B}}_{j+1} \approx \underline{\mathbf{B}}_j$, subtraction of both equation yields the repetitive control law as

$$\underline{\mathbf{U}}_{j+1} = \underline{\mathbf{U}}_j + \underline{\mathbf{B}}_j^{-1}\,(\underline{\mathbf{Y}}^* - \underline{\mathbf{Y}}_j - \underline{\mathbf{A}}_j(\mathbf{x}_{j+1}(0) - \mathbf{x}_j(0)) \tag{4.6}$$

provided the inverse of $\underline{\mathbf{B}}_j$ exists. To avoid a possible singularity problem in this inversion and the resulting excessive control effort of input vector $\underline{\mathbf{U}}_j$ and also to ensure stability in the process of repetitive control, optimal control theory is introduced despite the cost to the convergence speed, [8]. The control input vector can be computed by minimizing a quadratic cost function instead

$$\underline{\mathbf{J}}_j = \frac{1}{2}\,\underline{\varepsilon}_j^{\mathrm{T}}\,\underline{\mathbf{Q}}\,\underline{\varepsilon}_j + \frac{1}{2}\delta\underline{\mathbf{U}}_j^{\mathrm{T}}\,\underline{\mathbf{S}}\;\delta\underline{\mathbf{U}}_j \tag{4.7}$$

where $\underline{\varepsilon}_j = \underline{\mathbf{Y}}'^*_j - \underline{\mathbf{Y}}'_j$, $\underline{\mathbf{Y}}'^*_j = \underline{\mathbf{Y}}^* - \underline{\mathbf{A}}_j\mathbf{x}_j(0)$, $\underline{\mathbf{Y}}'_j = \underline{\mathbf{Y}}_j - \underline{\mathbf{A}}_j\mathbf{x}_j(0)$, $\delta\underline{\mathbf{U}}_j = \underline{\mathbf{U}}_j - \underline{\mathbf{U}}_{j-1}$ and the weighting matrices $\underline{\mathbf{Q}}$ and $\underline{\mathbf{S}}$ are taken to be symmetric and positive definite. By taking the partial derivative of $\underline{\mathbf{J}}_j$ about $\delta\underline{\mathbf{U}}_j$ equal to zero, the minimized cost function produces the repetitive control law as

$$\underline{\mathbf{U}}_{j+1} = \underline{\mathbf{U}}_j + \underline{\mathbf{G}}_j(\underline{\mathbf{Y}}^* - \underline{\mathbf{Y}}_j - \underline{\mathbf{A}}_j(\mathbf{x}_{j+1}(0) - \mathbf{x}_j(0)) \tag{4.8}$$

where $\underline{\mathbf{G}}_j$ is a gain matrix defined as

$$\underline{\mathbf{G}}_j = (\underline{\mathbf{B}}_j^{\mathrm{T}}\,\underline{\mathbf{Q}}\,\underline{\mathbf{B}}_j + \underline{\mathbf{S}})^{-1}\,\underline{\mathbf{B}}_j^{\mathrm{T}}\,\underline{\mathbf{Q}} \tag{4.9}$$

For a more detail procedure of the minimization of the cost function, see Appendix A.

4.3 Simulation of Repetitive Control with a Linear System

Simulations of repetitive control with linear follower-output subsystem without DC-motor and cam, were processed to test its control capability in the following cases:

Case 1. No disturbance is presented

Case 2. A periodic disturbance which has a vibration mode that has the same phase at the beginning step of every repetition is applied to the input. The period of the system motion is integer times that of the disturbance. This disturbance does not effect the deviation at initial conditions of every repetition.

Case 3. Periodic disturbance which has a phase shift at the beginning step of every repetition is applied to the input command. This type of disturbance increases the deviation at initial condition immensely and therefore very seriously disturbs the condition.

4.3.1 Simulation Based on Case 1

For the simulation of the follower-output system in Case 1, the following control parameters and initial guess of estimated system parameters are given by :

p :	number of steps in a repetition	(50)
T :	Period of the output motion	1(second)
a_j :	Adaptation factor	1 (regardless of j)
$\underline{\mathbf{Q}}$, $\underline{\mathbf{S}}$:	weighting matrix	$\underline{\mathbf{Q}} = \boldsymbol{I}_{(p \times p)}$, $\underline{\mathbf{S}} = 0.01\boldsymbol{I}_{(p \times p)}$
$\underline{\boldsymbol{R}}_0$:	Projection matrix of the first repetition	$\underline{\boldsymbol{R}}_0 = e^{40}\boldsymbol{I}_{(p \times p)}$

$l = p+m,\ m = 2$ (dimension of state variable $\mathbf{x}(k)$)

$\underline{\mathbf{A}}_0$: Estimation parameter of the first repetition $\underline{\mathbf{A}}_0 = zero_{(pxm)}$

$\underline{\mathbf{B}}_0$: Estimation parameter of the first repetition $\underline{\mathbf{B}}_0 = \boldsymbol{I}_{(pxp)}$

Other parameters used for the equations of a follower-output subsystem are given in Table 2.1. The initial condition for the first repetition is given by $\mathbf{x}(0)=[0\ 0]^T$. The original input, the exactly same trajectory of the cam lit, is applied to the follower-output system without control. The plots of actual output motion to the response of the original input vs. the desired output motion are shown in Fig.4.1. At the first repetition, repetitive control is not applied, only estimation of the system parameters are proceeded. In Fig.4.2, we can see the trend of vibration elimination where the repetitive control is applied from the second repetition. Fig.4.3 gives the result of repetitive control after 40 repetition. The corrected input command that produces the desired output is plotted in Fig.4.4 together with the original input. As shown in Fig.4.2, the repetitive control algorithm guarantees the perfect elimination of vibration in linear follower-output subsystem provided no disturbance is presented.

4.3.2 Simulation Based on Case 2

The second simulation with the control of the follower-output system takes place with the following disturbance :

$$w_j(k) = [\tfrac{H_c}{20} sin(20\pi k/p)\ \ 0]^T \tag{4.10}$$

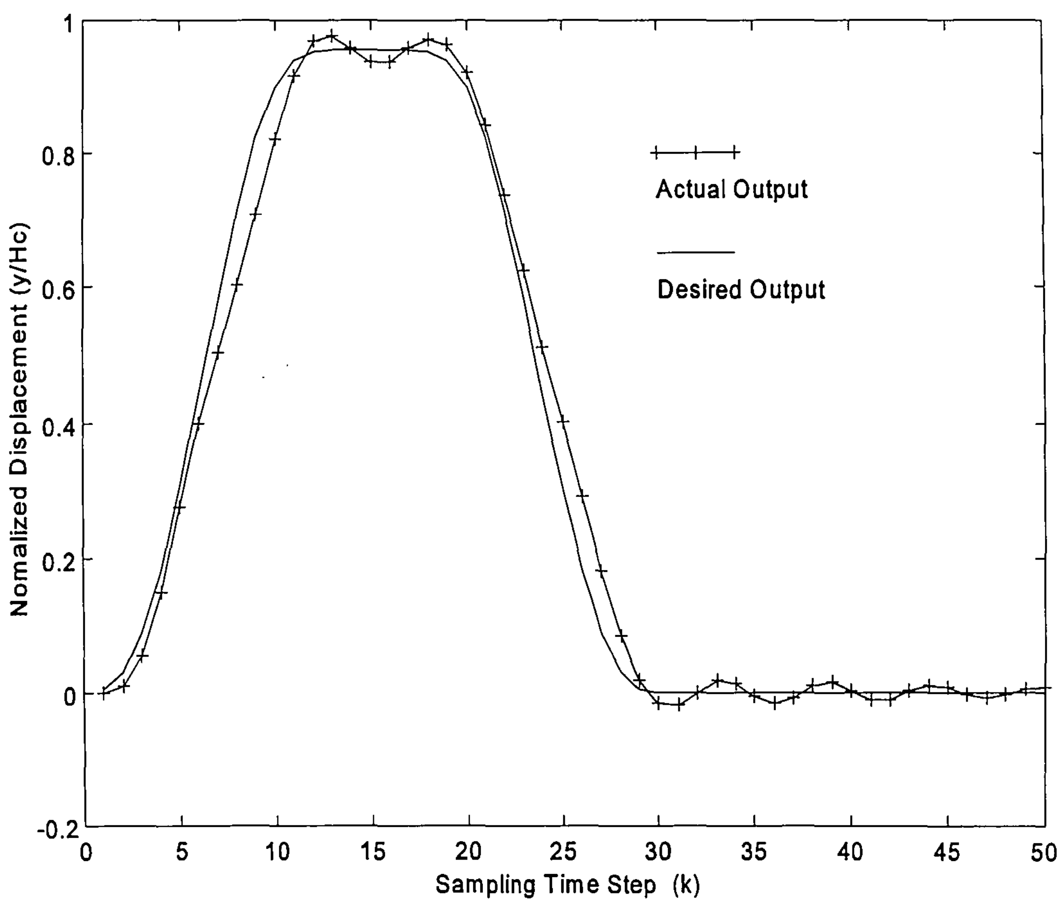


Figure 4.1 Dynamic Motion of the Output Mass without Repetitive Control Applied

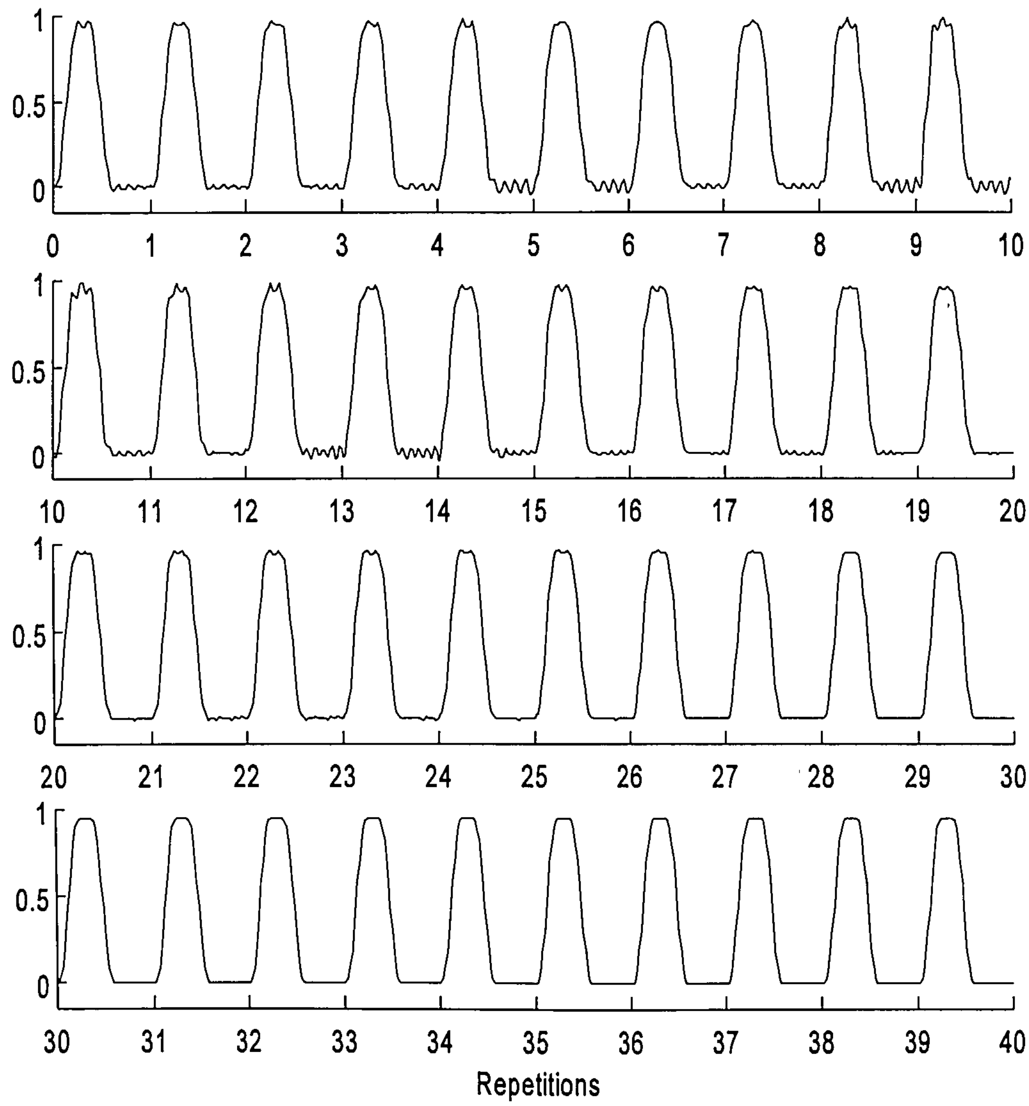


Figure 4.2 Trajectories of the Output Mass Motion Converging to the Desired Path with Repetitive Control

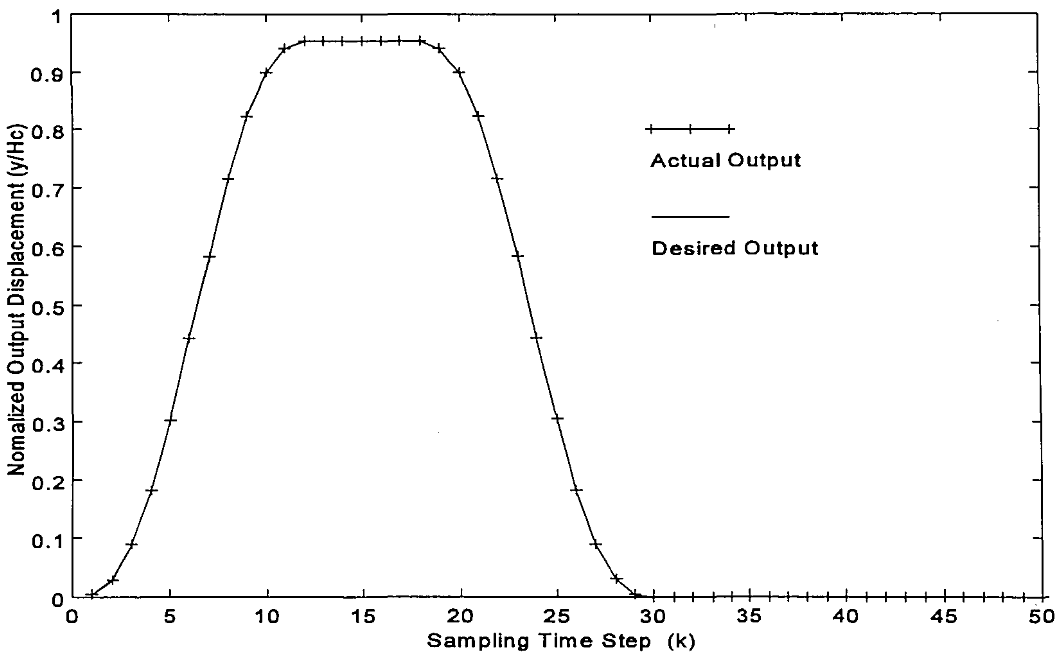


Figure 4.3 Trajectories of the Actual Output vs. Desired Motion after 40 Repetitions of the Control

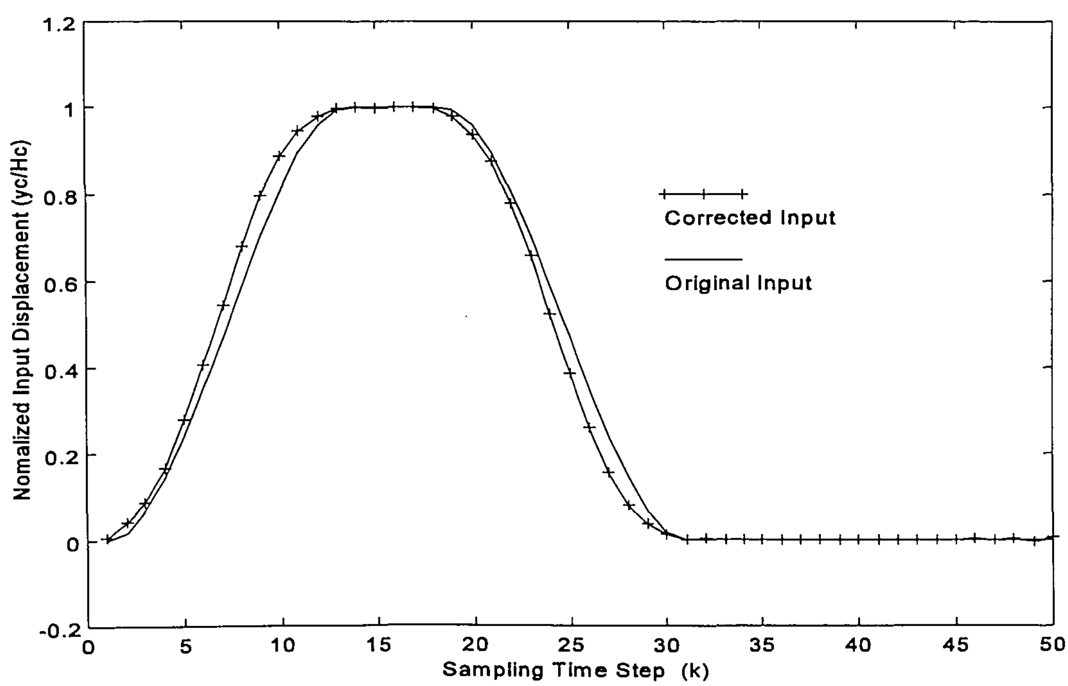


Figure 4.4 Corrected Input Command for the Desired Output

This disturbance has the same phase when k=0 and k=p so that

$$w_j(0) = w_j(p) = w_{j+1}(0) = 0$$

The state space equation in discrete time domain is therefore given by

$$\mathbf{x}_j(k+1) = \mathbf{A}\mathbf{x}_j(k) + \mathbf{B}\mathbf{u}_j(k) + w(k) \tag{4.11}$$

$$\mathbf{y}_j(k+1) = \mathbf{C}\mathbf{x}_j(k+1)$$

The same control parameters and estimation parameters in the first simulation are again used in the Case 2. Also, the same initial conditions are applied to the system at the beginning of the first repetition. Fig.4.5 shows the vibrational disturbance applied to the system with the same input command as in case 1, resulting in the vibration motion of the output shown in Fig.4.6. In Fig.4.7, vibration is drastically increased for the first 10 repetition cycles. This temporary vibration is generated due to the unfinished system identification. After correct identification is achieved, the vibration is eliminated. Fig.4.8 shows the result of the repetitive control after 50 repetition cycles. Fig.4.9 presents the corrected input command for desired output vs. original input. This simulation demonstrates that even though a constant phase disturbance is presented to the control of a linear system, repetitive control algorithm is still applicable.

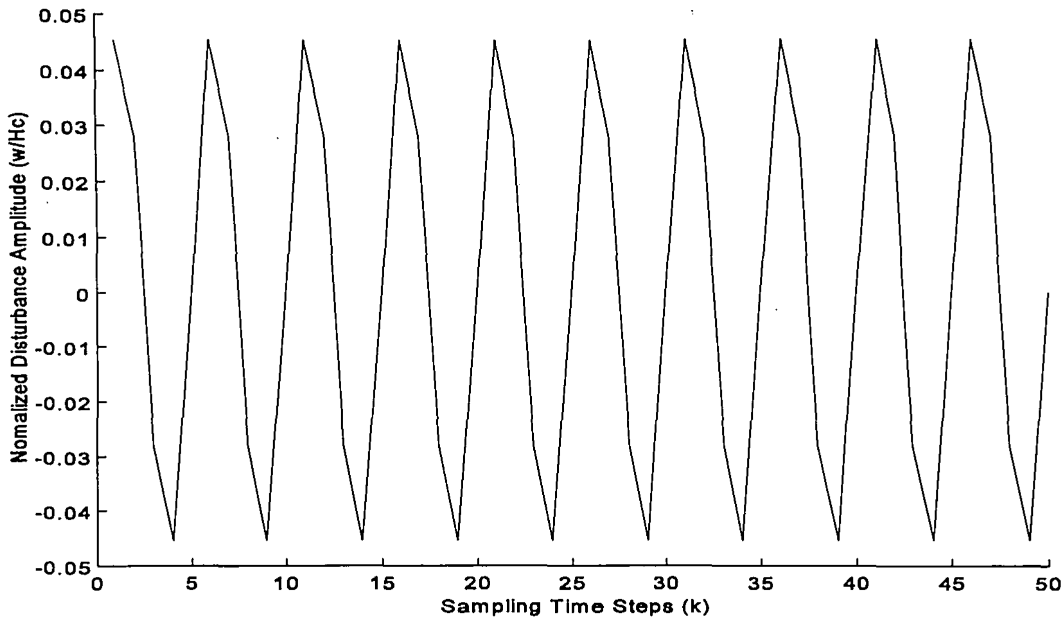


Figure 4.5 A Periodic Disturbance with a Constant Phase at the Beginning of Every Repetitions

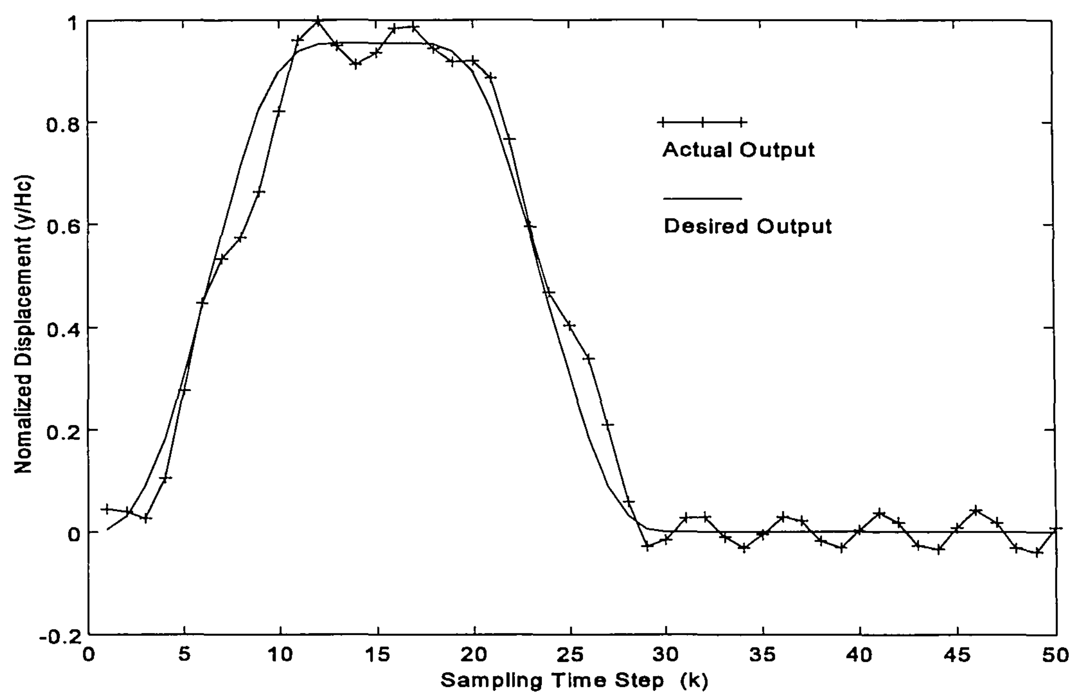


Figure 4.6 Dynamic Motion of Output Mass Presented with a Constant Phase Disturbance (No Control is Applied)

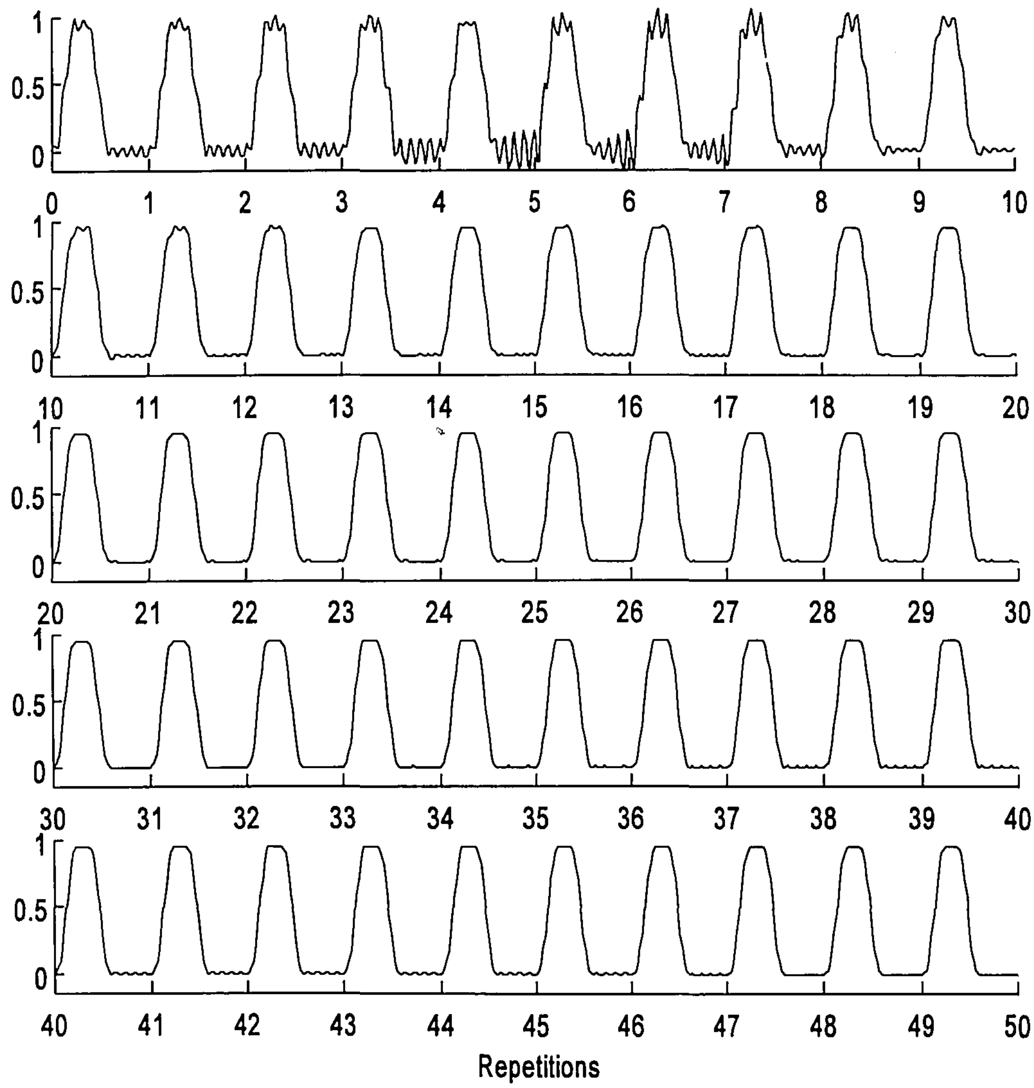


Figure4.7 Trajectories of Output Motion under Repetitive Control

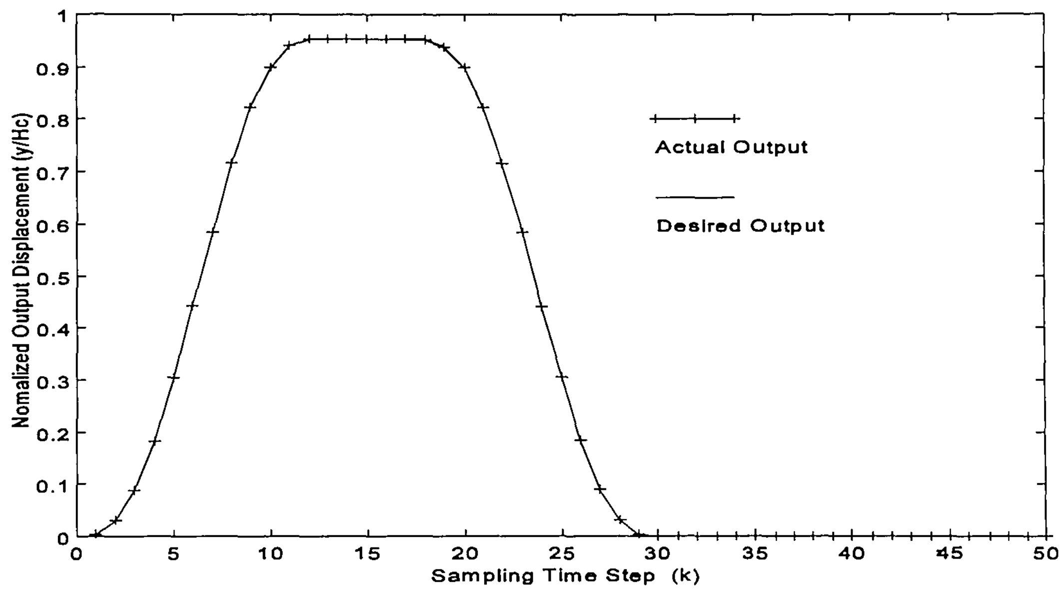


Figure 4.8 Actual Output Motion with Repetitive Control

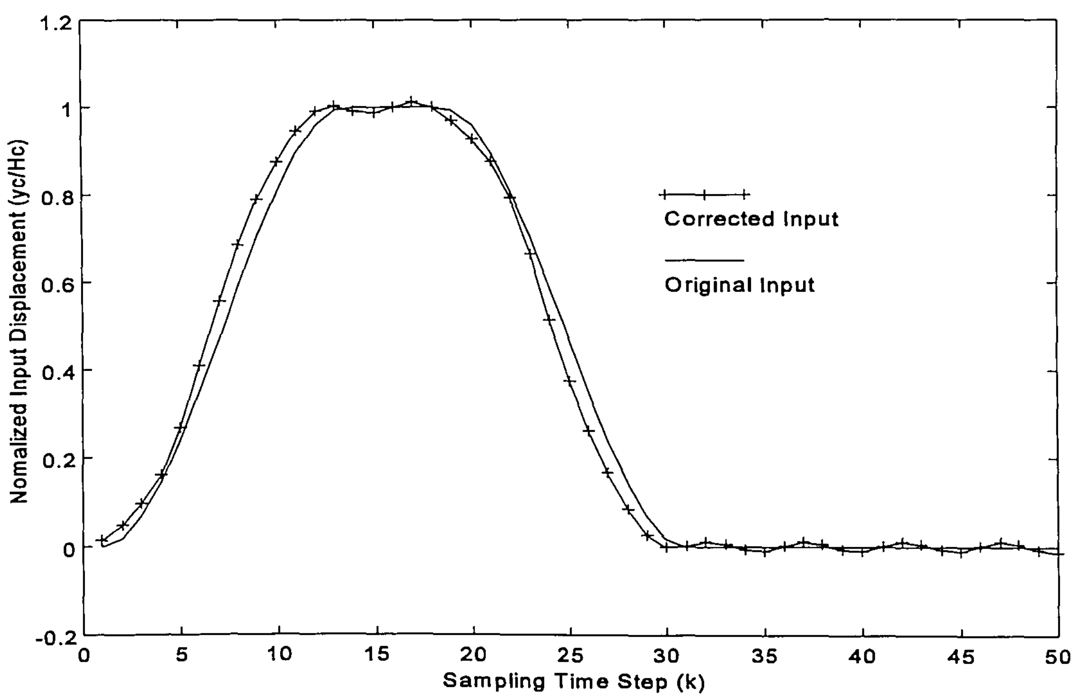


Figure 4.9 Corrected Input Command for Desired Output

4.3.3 Simulation Based on Case 3

The third simulation addresses a disturbance with phase changes at the first step of every repetition, such as :

$$w_j(k) = [\tfrac{H_c}{20} sin(11.1\pi k/p) \;\; 0]^{T} \tag{4.12}$$

This disturbance is applied to the discrete equation Eq.(4.11). However, this disturbance shown in Fig.4.10 which drives the actual vibrational output motion shown in Fig.4.11 has the feature :

$$w_j(p) = w_{j+1}(0) \neq w_j(0) \left\{ \begin{array}{l} = 0 \text{ for } j=1 \\ \neq 0 \text{ for } j > 1 \end{array} \right\} \tag{4.13}$$

wherein the amplitude of the disturbance at the beginning of every repetition is not constant but varies. Control and estimation parameters have been used as in previous simulations. Also the same original input is applied with the phase-shifting disturbance. The response of the output motion without control is plotted in Fig.4.11. In this case, repetitive control is applied up to 100 repetition cycles. However, the small pulse of the output motion at the beginning of every repetition has not been eliminated, as in Fig.4.12a,b. Actual output motion vs. desired motion is plotted in Fig.4.13 where both plots do not coincide at both ends of every repetition. The original input compared to the modified input is given in Fig.4.14.

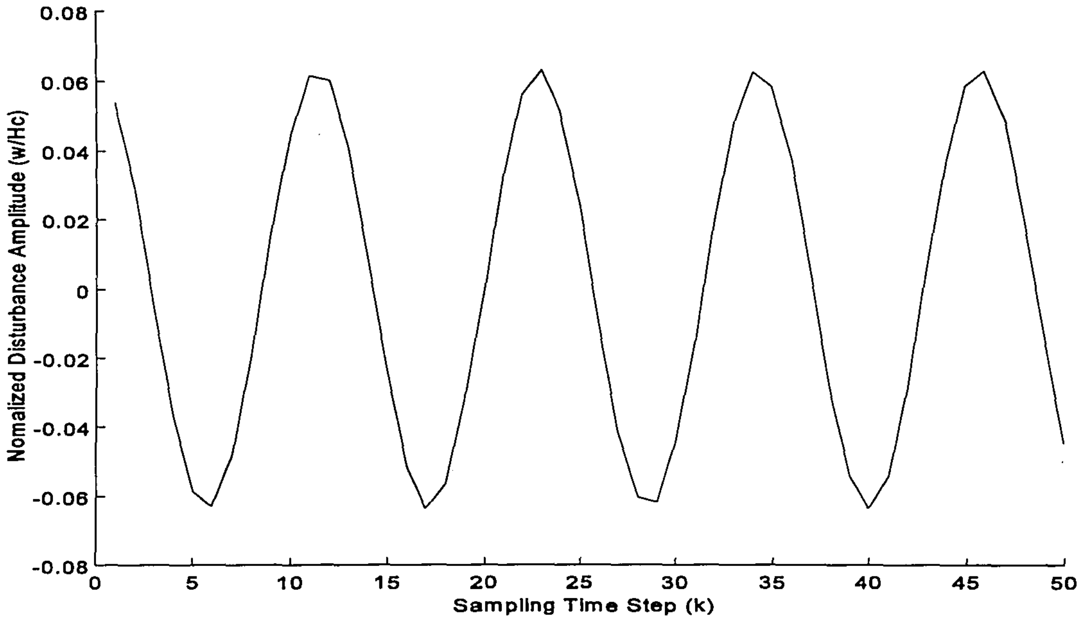


Figure 4.10 Sine Wave Disturbance with Non-Integral Frequency

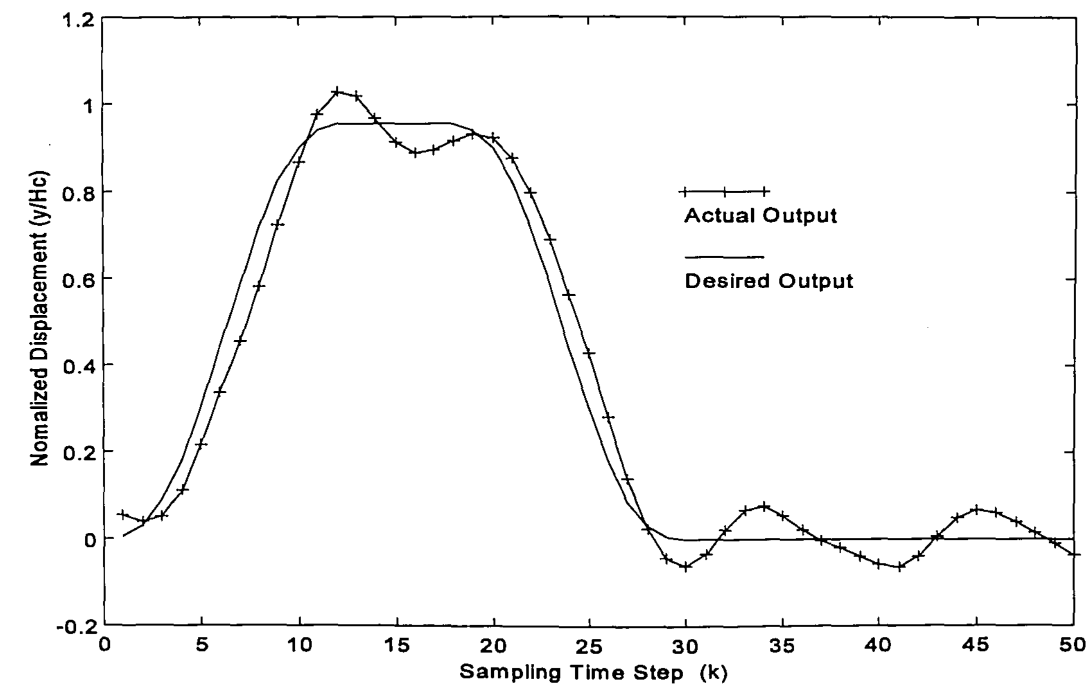


Figure 4.11 Dynamic Motion of the Output Mass without Control

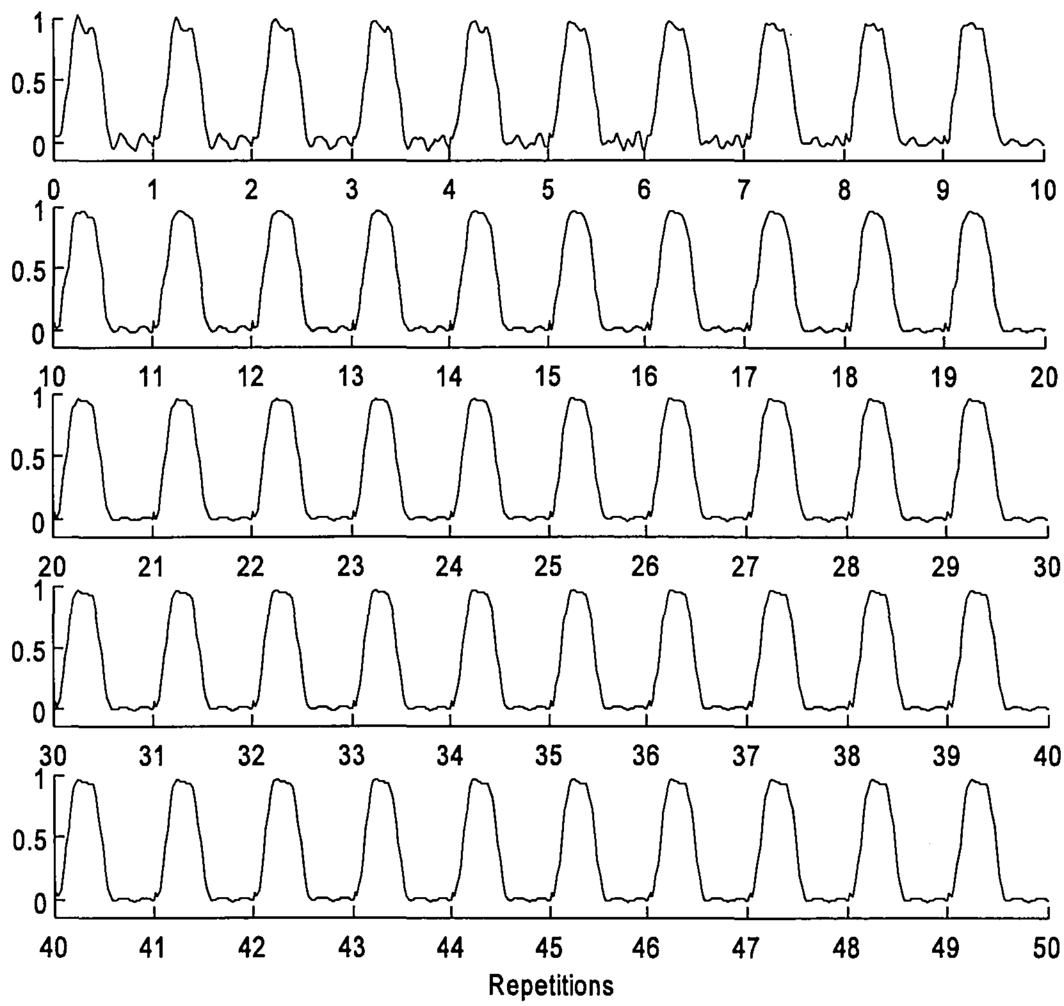


Figure 4.12a Trajectories of Output Motion under Repetitive Control (1~50 Repetitions)

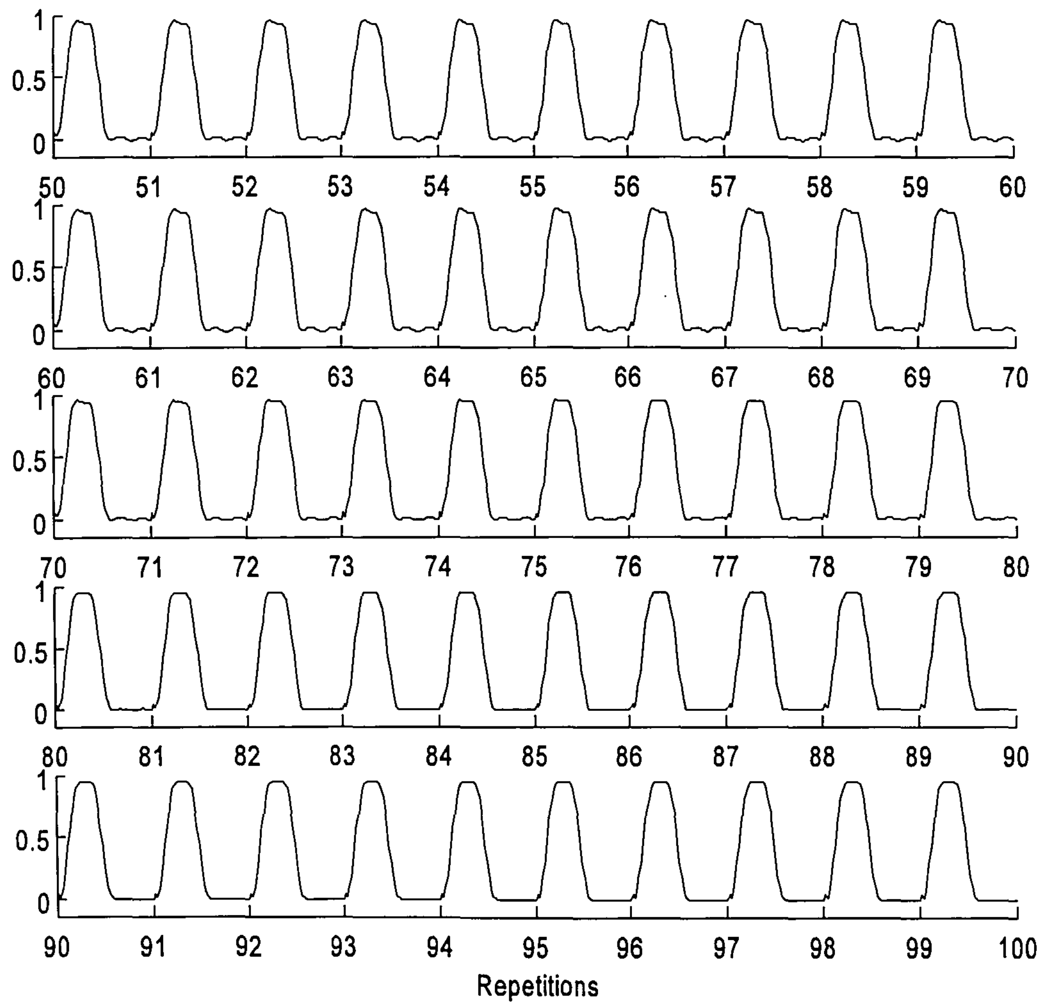


Figure 4.12b Trajectories of Output Motion under Repetitive Control (51~100 Repetitions)

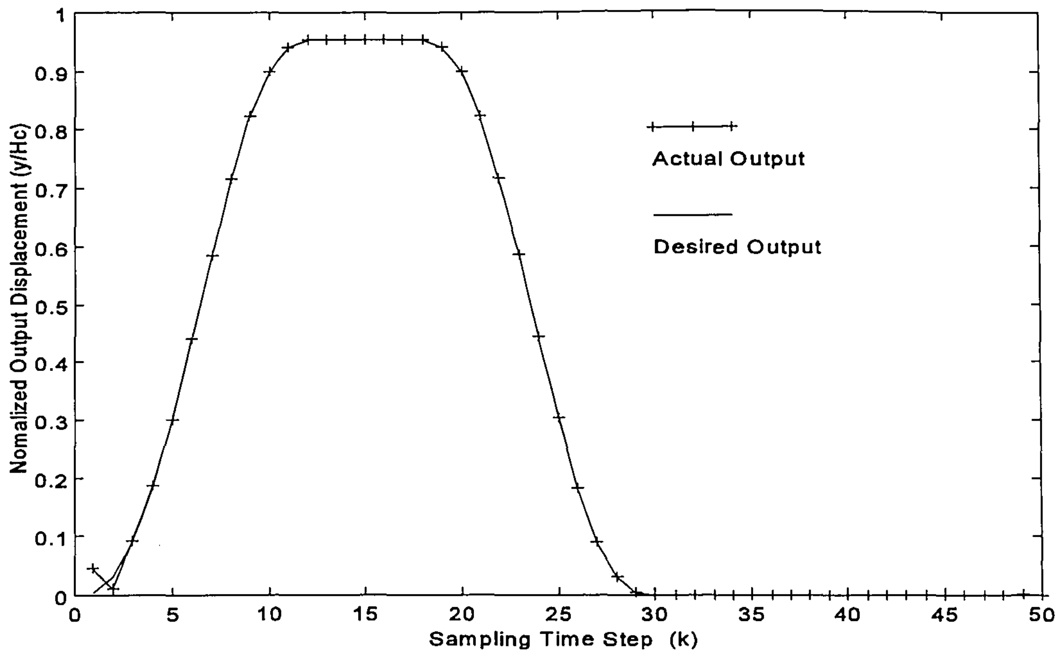


Figure 4.13 Motion of the Output Mass after the Control of 100 Repetitions

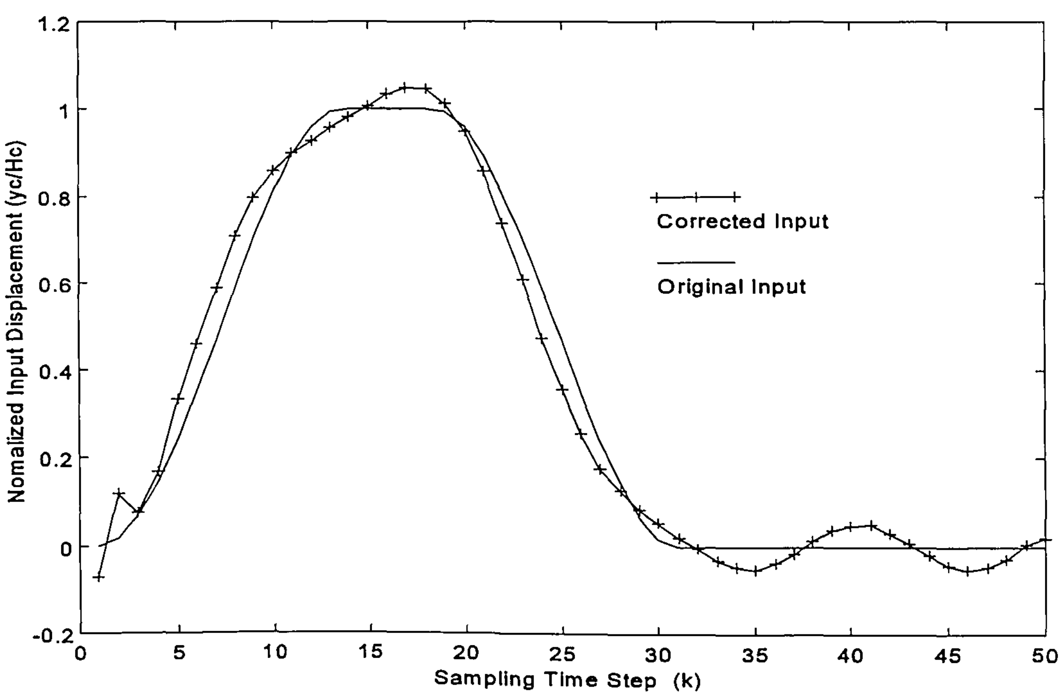


Figure 4.14 Modified Input Command for Desired Output Motion

4.4 Summary of the Results

Through the simulations, repetitive control of the linear follower-output subsystem has shown to guarantee convergence and stability. It is also shown that the estimated parameters in the repetition domain does not always coincide with those of the actual system as long as the estimation provides results in a stable controller. In the case with the constant phase disturbance, the controller recognizes it as a deviation from the linear system. Thus the estimated parameter shows same slight changes even after convergence. The simulation with Case 2 shows that this slightly changing estimation can handle the change in the initial condition at the beginning of a repetition caused only by system vibration.

The simulation with phase shifted disturbances to the system, showed that the drastic fluctuation of the initial condition can not be perfectly controlled with the changing system estimation.

Chapter 5

Piecewise Repetitive Control

The capability of repetitive control algorithm for the linear portion of the cam-follower system is verified in the previous chapter. This chapter presents a modification to repetitive control, namely piecewise repetitive control theory to be applied to the nonlinear problem in the control of entire cam-follower system that includes the DC-motor and the cam shown in Fig.2.1.

In piecewise repetitive control, a repetition is divided into several segments wherein the nonlinearity of the system is reduced sufficiently for stability. Repetitive control is applied only to the segments where the nonlinearity problem is minimized.

In order to apply repetitive control to a nonlinear system segment by segment, it is necessary to define each segment in discrete time domain, since repetitive control has been developed for discrete systems. The following features are assumed for the segments:

Assumption 1. Four segments (rise, dwell-1, return, dwell-2) are defined by the cam angle θ_c as in Fig.2.2. For convenience, these segments are expressed as a, b,c, d in subscript of parameters and variables. θ_a , ..., θ_d is starting angle of each segment, and k_a, ..., k_d is starting steps of each segment.

Assumption 2. Time spent in the process of each segment is constant regardless of repetition number.

Assumption 3. The mean angular velocity of cam ϖ for a repetition is assumed to be always constant.

Assumption 4. k_a, ..., k_d are chosen in such a way that the steps within each of the controllable segments are maximized.

Assumption 5. Repetitive control is applied only to the rise and the return segments.

5.1 Formulation of Piecewise Control

From the Assumptions 2 and 3, the starting steps of each segment are determined from the following relation :

$$k_\eta = \frac{\theta_\eta T}{2\pi \Delta t} \qquad (\eta = \text{a, b, c, d}) \qquad (5.1)$$

where T is the period of system, Δt is sampling time interval, and a, b, c,d represent rise, dwell-1, return and dwell-2 segments respectively. If the value of k_η is not an integer, it is necessary to convert it to an integer by rounding down or rounding up. This decision is made based on the Assumption 4.

According to Assumption 5, it is necessary to design two repetitive controllers, one each for the rise and the return segments. Estimation of the parameter matrix is also made for both these segments. Consider the system equations in jth repetition for both segments as

$$\text{Rise segment} \quad : \quad \underline{\mathbf{Y}}_{aj} = \underline{\mathbf{A}}_{aj}\, \mathbf{x}_j(k_a) + \underline{\mathbf{B}}_{aj}\, \underline{\mathbf{U}}_{aj} \tag{5.2a}$$

$$\text{Return segment} \quad : \quad \underline{\mathbf{Y}}_{cj} = \underline{\mathbf{A}}_{cj}\, \mathbf{x}_j(k_c) + \underline{\mathbf{B}}_{cj}\, \underline{\mathbf{U}}_{cj} \tag{5.2b}$$

where

$$\underline{\mathbf{Y}}_{aj} = [\, \mathbf{y}_j^T(k_a+1)\ \ \mathbf{y}_j^T(k_a+2) \ldots \mathbf{y}_j^T(k_b)\,]^T$$

$$\underline{\mathbf{U}}_{aj} = [\, \mathbf{u}_j^T(k_a)\ \ \mathbf{u}_j^T(k_a+1) \ldots \mathbf{u}_j^T(k_b-1)\,]^T$$

$$\underline{\mathbf{Y}}_{cj} = [\, \mathbf{y}_j^T(k_c+1)\ \ \mathbf{y}_j^T(k_c+2) \ldots \mathbf{y}_j^T(k_d)\,]^T$$

$$\underline{\mathbf{U}}_{cj} = [\, \mathbf{u}_j^T(k_c)\ \ \mathbf{u}_j^T(k_c+1) \ldots \mathbf{u}_j^T(k_d-1)\,]^T$$

and the dimension of unknown parameter matrices are

$$\text{Rise segment} \qquad \underline{\mathbf{A}}_{aj((kb\ -ka\ )xm)},\ \underline{\mathbf{B}}_{aj\ ((kb\ -ka\ )x(kb\ -ka\ ))} \tag{5.3a}$$

$$\text{Return segment} \qquad \underline{\mathbf{A}}_{cj\ ((kd\ -kc\ )xm)},\ \underline{\mathbf{B}}_{cj\ ((kd\ -kc\ )x(kd\ -kc\ ))} \tag{5.3b}$$

Rewriting these system equations in a new augmented matrix as in Chapter 4, we have

$$\underline{\mathbf{Y}}_{\eta j} = \underline{\phi}_{\eta j}\ \underline{\varphi}_{\eta j} \quad (j = 0,1,2, \ldots,\ \eta = \mathrm{a, c}) \tag{5.4}$$

where $\underline{\phi}_{\eta j} = [\underline{\mathbf{A}}_{\eta j}\ \ \underline{\mathbf{B}}_{\eta j}\,]$, $\quad \underline{\varphi}_{\eta j} = [\, \mathbf{x}_j^T(k_\eta)\ \ \underline{\mathbf{U}}_{\eta j}^{\ T}\ ]^T$

Once the initial guesses of $\underline{\phi}_{\eta 0}$ (η =a, c) are given, estimation of the system parameter matrix $\underline{\phi}_{\eta j}$ for the jth repetition are updated by an adaptive control scheme based on least-squares method. the updating formula is the same as Eq.(4.3) but rewritten for both the rise and the return segments.

The design of a piecewise controller using Eq.(5.4) follows along the same procedure presented in Chapter 4. Two repetitive controllers are developed for the control for each for the rise and the return segments. In dwell-1, dwell-2 segments, no control is applied because any control action during the dwell-1,2 portion, will not change the vibrational state of the output.

Introducing the desired output vector $\underline{\mathbf{Y}}_a^*$ and $\underline{\mathbf{Y}}_c^*$ for rise and return segments respectively and assuming some weighting matrices $\underline{\mathbf{Q}}_a$, $\underline{\mathbf{S}}_a$ for the rise segment, $\underline{\mathbf{Q}}_c$, $\underline{\mathbf{S}}_c$ for the return segment, the piecewise repetitive control law for the controllable segments is obtained as

$$\text{Rise} \quad : \quad \underline{\mathbf{U}}_{aj+1} = \underline{\mathbf{U}}_{aj} + \underline{\mathbf{G}}_{aj}\,(\underline{\mathbf{Y}}_a^* - \underline{\mathbf{Y}}_{aj} - \underline{\mathbf{A}}_{aj}(\mathbf{x}_{j+1}(k_a) - \mathbf{x}_j(k_a))) \qquad (5.5a)$$

$$\text{Dwell-1} : \quad \underline{\mathbf{U}}_{bj+1} = \underline{\mathbf{U}}_{bj} \qquad (5.5b)$$

$$\text{Return} \quad : \quad \underline{\mathbf{U}}_{cj+1} = \underline{\mathbf{U}}_{cj} + \underline{\mathbf{G}}_{cj}\,(\underline{\mathbf{Y}}_c^* - \underline{\mathbf{Y}}_{cj} - \underline{\mathbf{A}}_{cj}(\mathbf{x}_{j+1}(k_c) - \mathbf{x}_j(k_c))) \qquad (5.5c)$$

$$\text{Dwell-2} : \quad \underline{\mathbf{U}}_{dj+1} = \underline{\mathbf{U}}_{dj} \qquad (5.5d)$$

where the control gain matrices $\underline{\mathbf{G}}_{aj}$, $\underline{\mathbf{G}}_{cj}$ are given by

$$\underline{\mathbf{G}}_{aj} = (\underline{\mathbf{B}}_{aj}^T \underline{\mathbf{Q}}_a \underline{\mathbf{B}}_{aj} + \underline{\mathbf{S}}_a)^{-1} \underline{\mathbf{B}}_{aj}^T \underline{\mathbf{Q}}_a \qquad (5.6a)$$

$$\underline{\mathbf{G}}_{cj} = (\underline{\mathbf{B}}_{cj}^T \underline{\mathbf{Q}}_c \underline{\mathbf{B}}_{cj} + \underline{\mathbf{S}}_c)^{-1} \underline{\mathbf{B}}_{cj}^T \underline{\mathbf{Q}}_c \qquad (5.6b)$$

For more details about the design of these controller, see Chapter 4.3 and Appendix A.

5.2 Simulation of Piecewise Repetitive Control

In the simulation of piecewise repetitive control, the entire motor driven cam-follower nonlinear system model is used. The following control parameters and the initial guesses are given for the simulation.

p	:	number of steps in a repetition	50
k_a	:	the starting step of rise segment	0
k_b	:	the starting step of dwell-1 segment	13
k_c	:	the starting step of return segment	16
k_d	:	the starting step of dwell-2 segment	31
T	:	Period of system motion	1 *second*

$\underline{\mathbf{A}}_{\eta 0}$, $\underline{\mathbf{B}}_{\eta 0}$ (η = a,c) : Initial estimation of system parameter in rise and return segments

$$\underline{\mathbf{A}}_{a0} = zero_{((kb\text{-}ka)\mathrm{x}m)},\ \underline{\mathbf{B}}_{a0} = \boldsymbol{I}_{((kb\text{-}ka)\mathrm{x}(kb\text{-}ka))}$$

$$\underline{\mathbf{A}}_{c0},= zero_{((kd\text{-}kc)\mathrm{x}m)},\ \underline{\mathbf{B}}_{c0} = \boldsymbol{I}_{((kd\text{-}kc)\mathrm{x}(kd\text{-}kc))}$$

$\underline{\mathbf{Q}}_{\eta}$, $\underline{\mathbf{S}}_{\eta}$ (η = a,c) : Weighting matrices for optimal control in rise and return segment

$$\underline{\mathbf{Q}}_{a} = \boldsymbol{I}_{((kb\text{-}ka)\mathrm{x}(kb\text{-}ka))},\ \underline{\mathbf{S}}_{a} = 0.5\boldsymbol{I}_{((kb\text{-}ka)\mathrm{x}(kb\text{-}ka))}$$

$$\underline{\mathbf{Q}}_{c} = \boldsymbol{I}_{((kd\text{-}kc)\mathrm{x}(kd\text{-}kc))},\ \underline{\mathbf{S}}_{c} = 0.05\boldsymbol{I}_{((kd\text{-}kc)\mathrm{x}(kd\text{-}kc))}$$

$\underline{\mathbf{R}}_{\eta 0}$ (η = a,c) : Initial projection matrix in adaptive control in rise and return segment

$$\underline{\mathbf{R}}_{a0} = e^{30}\boldsymbol{I}_{((kb\text{-}ka+m)\mathrm{x}(kb\text{-}ka+m))}$$

$$\underline{\mathbf{R}}_{c0} = e^{20}\boldsymbol{I}_{((kd\text{-}kc+m)\mathrm{x}(kd\text{-}kc+m))}$$

Other system parameters are given in Table 2.1. The initial condition of the first repetition is $\mathbf{x}_0(0)=[0\ 0\ 0\ 0]^T$. Original input command for the motor has a constant velocity $w = 2/T$ and starts at zero degree. This original input causes the vibration in the output motion shown in Fig.5.1 when no control is applied. Piecewise repetitive control is applied from the second repetition onwards. Fig.5.2 shows the trend of vibration elimination as a function of repetition. The plot of actual output vs. desired output is illustrated in the Fig.5.3 which is free of vibration after the control for 50 repetition cycles. The corrected input command of motor driving the desired output motion is plotted in Fig.5.4.

5.3 Summary of the Simulation

Due to the kinematic nonlinearity of the cam, the repetitive control used in Chapter 4, failed in the control of the cam-follower nonlinear system. In this system, the relationship between the displacement input and that of output shows totally different trends in each segment: it is approximately proportional in the rise segment, it is proportional in a negative sense, in the return segment. In the dwell-1,2 segments, there is no response of the output from the change of input command. Taking theses characteristics of each segments into consideration, repetitive control is applied segment by segment. In the dwell-1,2 segments, no control is applied since the geometry of the cam in these segments does not permit any type of controls. Repetitive control is applied only to the rise and the return segments. The desired output motion is obtained after less than 50 repetitions of piecewise control.

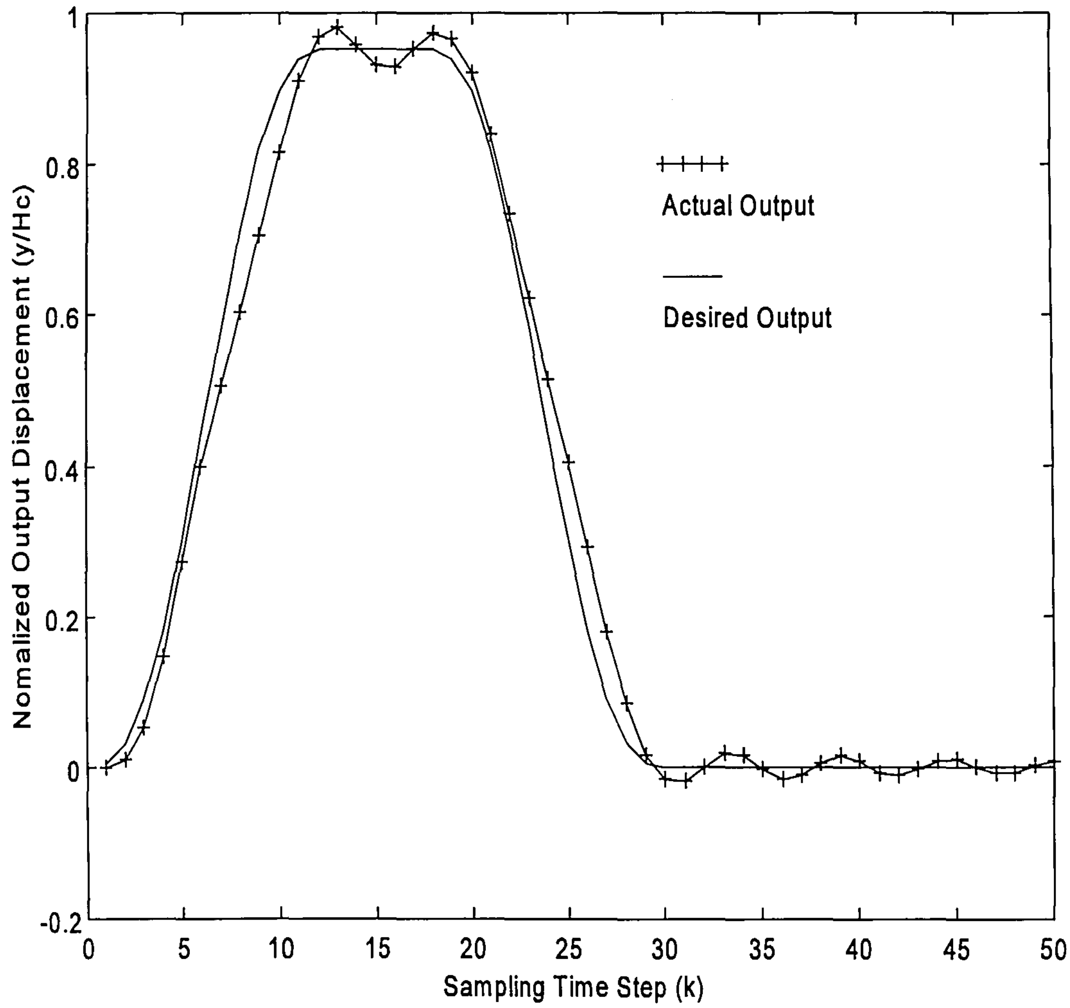


Figure 5.1 Dynamic Motion of the Nonlinear Cam-Follower System (No Control Applied)

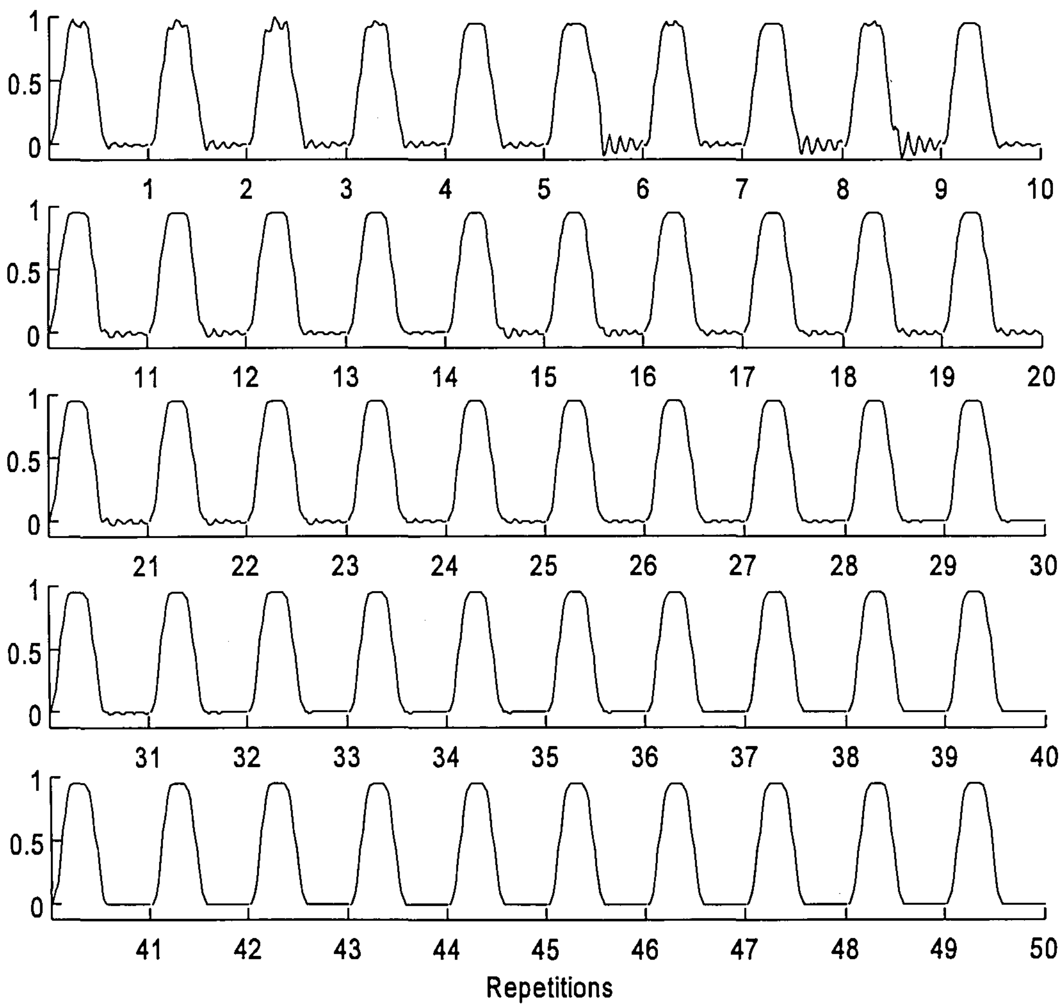


Figure 5.2 Trajectories of Output Motion under Piecewise Repetitive Control

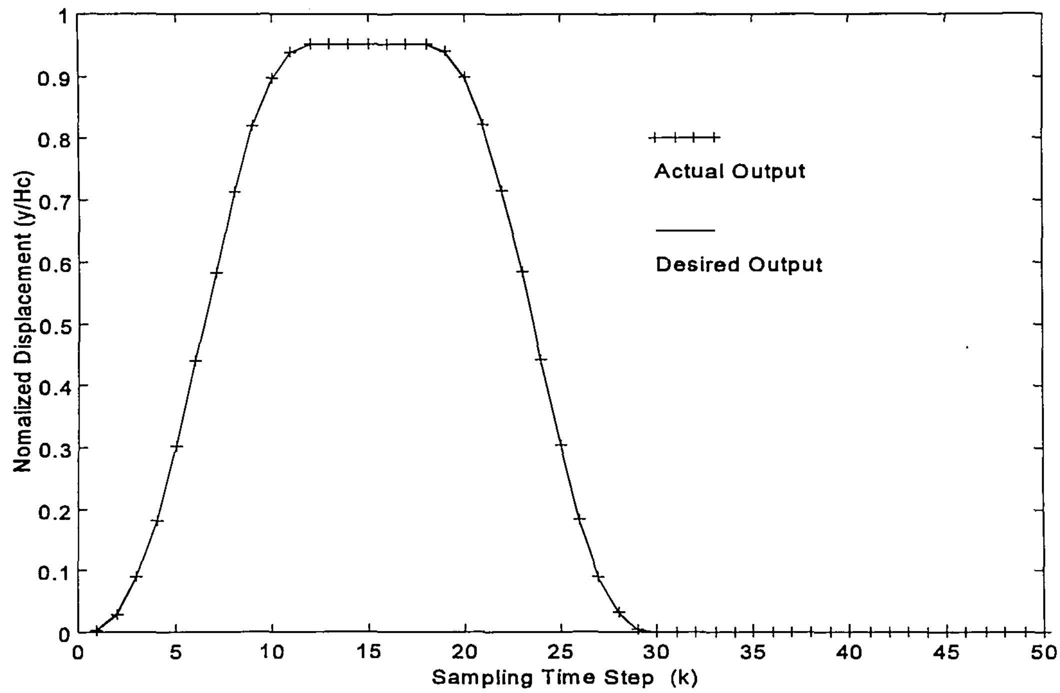


Figure 5.3 Motion of the Output Mass vs. Desired Output Motion after Piecewise Control of the Nonlinear System

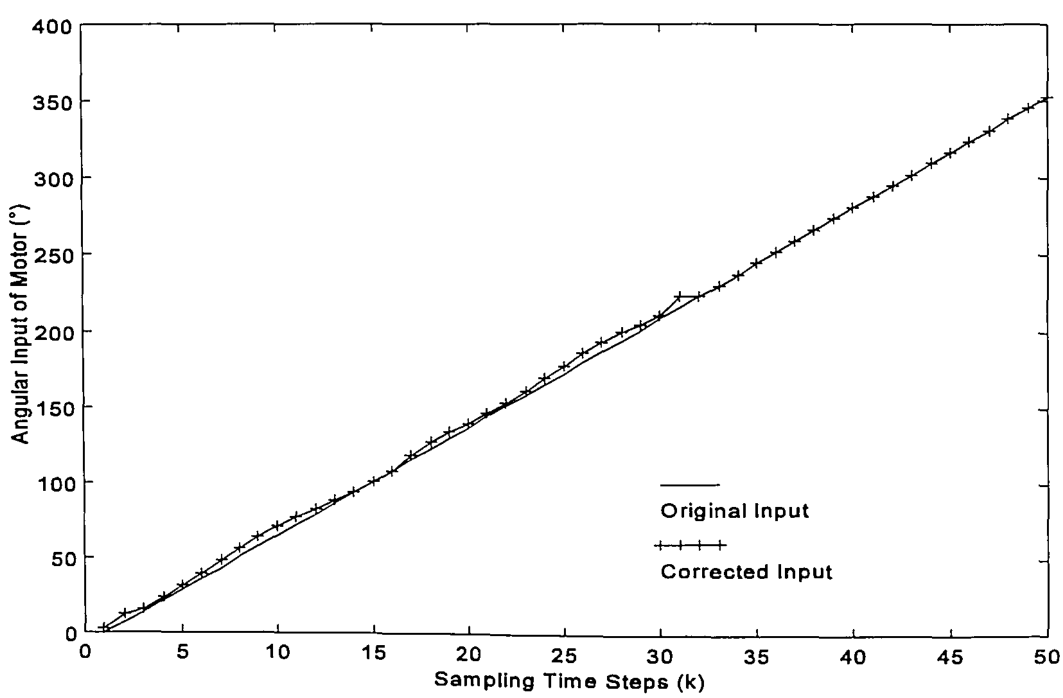


Figure 5.4 Modified Motor Input Command for the Desired Output Motion

Chapter 6

Conclusion

This thesis uses two main approaches in control of a high-speed cam-follower system : an analysis of the dynamic motions in the system and a designing of a repetitive controller. A large part of this research is devoted to the formulation of a proper gain matrix to ensure stability and convergence of the repetitive control procedure. Repetitive control algorithm has been shown to be an integrated tool in the handling of a motor driven cam-follower system for high-speed applications. It has also shown that repetitive control handles nonlinearity and disturbance problems which occur in all real mechanisms and machiniaries.

The repetitive controller introduced in this thesis is developed for a discrete single-input single-output system with single degree of freedom. Input command and output measurements are related only to the angular or linear displacement of actuator and sensors. All system models are converted to the discrete-time system.

From the view point of inverse dynamics, the control gain matrix can be said to be the inverse of the system parameter matrix which can be determined by analysis and linearization of the system. However, due to the instability and lack of robustness, an adaptive control technique has been used instead. The adaptive controller assumes the system parameters and updates the characteristics of the controller from the relationship between the inputs and outputs.

Repetitive control can handle the changes in the initial condition at the beginning of each repetition. It permits the system to proceed towards repeating the operation without stopping even if the system completes a repetition with error, so that the system operation starts from a different initial condition in the next repetition. This is the main advantage of repetitive control theory over the learning control.

This research illustrates the control possibility of nonlinear systems, but does not guarantee it. A nonlinear system which has a periodic output path that has an acceptable deviation from that of a linear system can be controlled without modifying the repetitive control algorithm. A heavy nonlinear system which has a periodic output path showing a totally different trend from linear system can be controlled by dividing the output path into several segments so that the characteristics of the output path in relation to input command become closer to that of a linear system.

References

1. Anderson, D.G. "Cam Dynamics," Production Engineering, Vol.24, Oct., 1963, pp.70-176.

2. Bloom, D., Radcliffe, C.W. "The Effect of Cam Shaft Elasticity on the Response of Cam Driven Systems," ASME, paper 64-MECH-41, 1964.

3. Koster, M.P. "Effect of Flexibility of Driving Shaft on the Dynamic Behavior of a Cam Mechanism," ASME, Journal of Engineering of Industry, Vol.97, No.2, 1975, pp.595-602.

4. Ardayfio, D. "Dynamics of High-Speed Cam Mechanism with Damped Flexible Followers Driven by Flexible Cam Shaft," ASME, pp.76-DET-69, 1976.

5. Hrones, J.A. "Analysis of Dynamic Forces in a Cam Driven System," Trans. of ASME, Vol.70, July, 1948, pp.473-482.

6. Faires, V.M. *Design of Machine Elements*, Macmillan, 4th Edition, 1965.

7. Phan, M., Longman, L.W. "Mathematical Theory of Learning Control for Linear Discrete Multivariable Systems," American Institute of Aeronautics and Astronautics, Inc., Columbia University, New York, NY, 1988.

8. Phan, M., Juang, J.N. "Design of Learning Controllers Based on Auto-Regressive Representation of a Linear System," Journal of Guidance, Control and Dynamics, March-April, 1996.

9. Chew, M., Phan, M. "Application of Learning Control Theory to Mechanisms: Part 1-Inverse Kinematics and Parametric Error Compensation," Machine Elements and Machine Dynamics ASME 1994, DE-Vol 71, pp.25-32.

10. Chew, M., Chuang, J.N. "Minimizing Residual Vibrations in High-Speed Cam-Follower Systems over a Range of Speeds," Trans. of the ASME, Vol.117, March, 1995, pp.166-172.

11. Chew, M., Phan, M. "Application of Learning Control Theory to Mechanisms Part 2: Reduction of Residual Vibrations in High-Speed Electromechanical Bonding Machines," Machine Elements and Machine Dynamics ASME, 1994, Vol.71, pp.33-40.

12. Juang, J.N., Horta, L.G., Robertshaw, H.H. "A Slewing Control Experiment for Flexible Structures," Journal of Guidance, Control and Dynamics, Vol.9, Sept.- Oct., 1986, pp.599-607.

13. Phan, M., Longman, L.W., Juang, J.N. "Indirect Repetitive Control for Linear Discrete Multivariable Systems," Proceedings of the 27th Annual Allerton Conference on Communication, Control and Computing, Monticello, IL, Sept., 1989, pp.867-876.

14. Dubowsky, S., Desforges, D.T. "The Application of Model-Referenced Adaptive Control to Robotic Manipulator," Journal of Dynamic Systems, Measurement of Control, Trans. of ASME, Vol.101, No.3, Sept., 1979, pp.193-200.

15. Phan, M., Juang, J.N., Hyland, D.C. "Neural Networks in Identification and Control of Dynamic System," *Wave Motion, Intelligent Structures and Nonlinear Mechanisms,* Ed. by Guran, A. and Inman, D.J., Published by World Scientific Publishing Co., pp.194-225, 1995.

16. Astrom, K.J., WittenMark, B. *Adaptive Control,* Addison Wesley, 1989.

17. Greenwood, D. *Principles of Dynamics,* Second Edition, Prentice-Hall, Englewood Cliff, NJ, 1988.

18. Ogata, K. *System Dynamics* Second Edition, International Editions, Prentice-Hall International Inc., Englewood Cliffs, NJ, 1992.

19. Ljung, L. *System Identification: Theory for the User,* Prentice Hall Inc., Englewood Cliffs, NJ, 1987.

20. Friedland, B. *Control System Design: An Introduction to State Space Methods,* McGraw-Hill, Inc., 1986.

21. Sung, B.J. "Simulation of Suspension Systems for Ground-Based Validation Testing of Flexible Space Structure," M.S. Thesis, Lehigh University, Bethlehem, PA, Oct., 1995.

Appendix A : Derivation of the Optimized Control Gain

Consider a linear system equations in the repetition domain,

$$\underline{\mathbf{Y}}'_j = \underline{\mathbf{B}}\ \underline{\mathbf{U}}_j \tag{A1}$$

where $\underline{\mathbf{Y}}'_j$ and $\underline{\mathbf{U}}_j$ is the output and input column vector respectively, which have same dimension, and $\underline{\mathbf{B}}_j$ is the system matrix. $\underline{\mathbf{Y}}'^*$ is assumed to be the desired output. The objective of the optimal control is to find out $\underline{\mathbf{U}}_j$ which produces the desired output $\underline{\mathbf{Y}}'^*$ at jth repetition from the history of (j-1)th repetition. Consider the quardradic cost function given by

$$\underline{\mathbf{J}} = \frac{1}{2}\,\underline{\varepsilon}_j^{T}\,\underline{\mathbf{Q}}\,\underline{\varepsilon}_j + \frac{1}{2}\,\delta\underline{\mathbf{U}}_j^{T}\,\underline{\mathbf{S}}\,\delta\underline{\mathbf{U}}_j \tag{A2}$$

where $\underline{\varepsilon}_j = \underline{\mathbf{Y}}'^* - \underline{\mathbf{Y}}'_j$, $\delta\underline{\mathbf{U}}_j = \underline{\mathbf{U}}_j - \underline{\mathbf{U}}_{j-1}$. The error $\underline{\varepsilon}_j$ can be written as

$$\underline{\varepsilon}_j = (\underline{\mathbf{Y}}'^* - \underline{\mathbf{Y}}'_{j-1}) - (\underline{\mathbf{Y}}'_j - \underline{\mathbf{Y}}'_{j-1}) = \underline{\varepsilon}_{j-1} - \delta\underline{\mathbf{Y}}'_j \tag{A3}$$

The differenced output $\delta\underline{\mathbf{Y}}'_j$ is obtained from the differneced input $\delta\underline{\mathbf{U}}_j$ as

$$\delta\underline{\mathbf{Y}}'_j = \underline{\mathbf{B}}\ \delta\underline{\mathbf{U}}_j \tag{A4}$$

Eq.(A2) can be written as

$$\underline{\mathbf{J}} = \frac{1}{2}[(\underline{\varepsilon}_{j-1} - \underline{\mathbf{B}}\,\delta\underline{\mathbf{U}}_j)^T\, \underline{\mathbf{Q}}\,(\underline{\varepsilon}_{j-1} - \underline{\mathbf{B}}\,\delta\underline{\mathbf{U}}_j) + \delta\underline{\mathbf{U}}_j^T\, \underline{\mathbf{S}}\, \delta\underline{\mathbf{U}}_j\,] \tag{A5}$$

$$= \frac{1}{2}[\, \underline{\varepsilon}^T_{j-1}\, \underline{\mathbf{Q}}\, \underline{\varepsilon}_{j-1} - \underline{\varepsilon}^T_{j-1}\, \underline{\mathbf{Q}}\, \underline{\mathbf{B}}\, \delta\underline{\mathbf{U}}_j - \delta\underline{\mathbf{U}}^T_j\, \underline{\mathbf{B}}^T\, \underline{\mathbf{Q}}\, \underline{\varepsilon}_{j-1} + \delta\underline{\mathbf{U}}^T_j\, \underline{\mathbf{B}}^T\, \underline{\mathbf{Q}}\, \underline{\mathbf{B}}\, \delta\underline{\mathbf{U}}_j + \delta\underline{\mathbf{U}}^T_j\, \underline{\mathbf{S}}\, \delta\underline{\mathbf{U}}_j\,]$$

Take a partial derivative about $\delta\underline{\mathbf{U}}_j$ to minimize the $\underline{\mathbf{J}}$ as

$$\partial\underline{\mathbf{J}} / \partial\delta\underline{\mathbf{U}}_j = - \underline{\mathbf{B}}^T\, \underline{\mathbf{Q}}\, \underline{\varepsilon}_{j-1} + (\, \underline{\mathbf{B}}^T\, \underline{\mathbf{Q}}\, \underline{\mathbf{B}} + \underline{\mathbf{S}})\, \delta\underline{\mathbf{U}}_j \tag{A6}$$

Setting the result to zero and solving for $\delta\underline{\mathbf{U}}_j$

$$\delta\underline{\mathbf{U}}_j = (\, \underline{\mathbf{B}}^T\, \underline{\mathbf{Q}}\, \underline{\mathbf{B}} + \underline{\mathbf{S}})^{-1}\, \underline{\mathbf{B}}^T\, \underline{\mathbf{Q}}\, \underline{\varepsilon}_{j-1} \tag{A7}$$

Expanding the differenced input and error, and introducing a gain matrix $\underline{\mathbf{G}}$, Eq.(A7) yields the optimized repetitive control law as

$$\underline{\mathbf{U}}_j = \underline{\mathbf{U}}_{j-1} + \underline{\mathbf{G}}\,(\, \underline{\mathbf{Y}}'^* - \underline{\mathbf{Y}}'_{j-1}\cdot) \tag{A8}$$

where the gain is $\underline{\mathbf{G}} = (\, \underline{\mathbf{B}}^T\, \underline{\mathbf{Q}}\, \underline{\mathbf{B}} + \underline{\mathbf{S}})^{-1}\, \underline{\mathbf{B}}^T\, \underline{\mathbf{Q}}$.

Appendix B : MatLab Codes

The Basic Functions of the Cam Lift

y23p.m

```
function y=ycam(theta)

% The dispacement of yc in function of angular
%displacement 'theta'
% 2-3 Polynomial

global Hc
temp=rem(theta,2*pi);
zeta=temp./(pi/2);

for iter=1:1:length(theta)
 if zeta(1,iter)<90*2/180                        % Rise
   y(1,iter)=Hc*(3*zeta(1,iter)^2-2*zeta(1,iter)^3);
 elseif zeta(1,iter)<126*2/180                   % Dwell
   y(1,iter)=Hc;
 elseif zeta(1,iter)<216*2/180                   % Return
y(1,iter)=Hc*(-10.368+20.16*zeta(1,iter)-11.4*zeta
(1,iter)^2+2*zeta(1,iter)^3);
 else                                  % Dwell
   y(1,iter)=0;
 end
end
```

yd23p.m

```
% First Derivative of yc
function y=ydcam_f(theta)
Hc=0.01;
temp=rem(theta,2*pi);
zeta=temp./(pi/2);
for iter=1:1:length(theta)
 if zeta(1,iter)<90*2/180
   y(1,iter)=Hc*(6*zeta(1,iter)-6*zeta(1,iter)^2)/90;
 elseif zeta(1,iter)<126*2/180
   y(1,iter)=0;
 elseif zeta(1,iter)<216*2/180
y(1,iter)=Hc*(20.16-11.4*2*zeta(1,iter)+6*zeta(1,it
er)^2)/90;
 else
   y(1,iter)=0;
 end
end
```

ydd23p.m

```
function y=yddcam_f(theta)
% Second serivative of yc
Hc=0.01;
temp=rem(theta,2*pi);
zeta=temp./(pi/2);
for iter=1:1:length(theta)
 if zeta(1,iter)<90*2/180
   y(1,iter)=Hc*(6-12*zeta(1,iter))/90^2;
 elseif zeta(1,iter)<126*2/180
   y(1,iter)=0;
 elseif zeta(1,iter)<216*2/180
   y(1,iter)=Hc*(-11.4*2+12*zeta(1,iter))/90^2;
 else
   y(1,iter)=0;
 end
end
```

The Desired Output Function, y345p.m

```
function y=yout(theta)
global H
temp=rem(theta,2*pi);
zeta=temp./(pi/2);
for iter=1:1:length(theta)
 if zeta(1,iter)<90*2/180
y(1,iter)=H*(10*zeta(1,iter)^3-15*zeta(1,iter)^4+6*
zeta(1,iter)^5);
 elseif zeta(1,iter)<126*2/180
   y(1,iter)=H;
 elseif zeta(1,iter)<216*2/180
temp=118.333-338.6879*zeta(1,iter)+383.0399*zet
a(1,iter)^2-211.5999*zeta(1,iter)^3;
y(1,iter)=H*(temp+57*zeta(1,iter).^4-6*zeta(1,iter).
^5);
 else
   y(1,iter)=0;
 end
end
```

Simulations of the Linear System

fig41_44.m (Case 1)

```
% Repettitive Cntrl of Linear Sys
% Adaptive Control with
%   Least Square Method
clear
global H Hc
Kf=315;Ks=15;M=0.1;c=0.5;
Hc=0.01;H=Kf/(Ks+Kf)*Hc;
Ac=[0 1;-(Kf+Ks)/M -c/M];Bc=[0 Kf/M]';
C=[1 0];
[ai,aj]=size(Ac);[bi,bj]=size(Bc);
[ci,cj]=size(C);
%% Control Parameters %%
w=2*pi;prd=2*pi/w;
```

```
p=50;
gain=0.4;
alpha=1;
mxrpt=40;
mxitr=1;
intvl=10;
Agss=zeros(ci*p,aj);
Bgss=eye(ci*p,bj*p).*5;
sz=aj+ci*p;
R=eye(sz,sz).*1e40;
Q=eye(p).*1;
S=eye(p).*0.01;
%%%%%%%%

dt=prd/p;
[A,B]=c2d(Ac,Bc,dt);
for k=1:p
  ystr(k,1)=y345p(w*dt*k);
  ustr(k,1)=y23p(w*dt*(k-1));
  dstb(k,1)=H/20*sin(20*pi*k/p);
end
q1=0;q2=0;
u=ustr;
x=[0 0]';

for jj=1:mxrpt
  jj
  if rem(jj,intvl)==1
    q1=q1+1;
    q2=0;
  end
  q2=q2+1;
  xo=x;
  for k=1:p
    x=A*x+B.*u(k);
    y(k,1)=C*x;
    tm((q2-1)*p+k,q1)=((jj-1)*p+k)/p*prd;
    yy((q2-1)*p+k,q1)=y(k,1);
  end
  for itr=1:mxitr
    k1=1:aj;
    k2=1:ci*p;
    PP(:,k1)=Agss(:,k1);
    PP(:,k2+aj)=Bgss(:,k2);
    UU(1,k1)=xo(k1)';
    UU(1,k2+aj)=u(k2)';
    err=y-PP*UU';
    dnm=alpha+UU*R*UU';
    PP=PP+err*(UU*R)./dnm;
    R=R-(R*UU'*UU*R)./dnm;
    Agss(:,k1)=PP(:,k1);
    Bgss(:,k2)=PP(:,k2+aj);
  end
```

```
du=(Bgss'*Q*Bgss+S)\Bgss'*Q*(ystr-y-Agss*(x-xo)
);
%  du=Bgss\(ystr-y-Agss*(x-xo));
  u=u+gain.*du;
  yold=y;
  if jj==1
    figure(1)
    plot(y.*100)
    hold on
        plot(ystr.*100,'.')
        text(30,0.6,'Desired Output')
        text(30,0.75,'Actual Output')
        plot([30 32 34],[0.8 0.8 0.8])
        plot([30 32 34],[0.65 0.65 0.65],'.')
        ylabel('Nomalized Displacement (y/Hc)')
        xlabel('Sampling Time Step  (k)')
        hold off
  end
end
hold off
figure(2)
plot(y.*100)
hold on
plot(ystr.*100,'.')
text(30,0.6,'Desired Output')
text(30,0.75,'Actual Output')
plot([30 32 34],[0.8 0.8 0.8])
plot([30 32 34],[0.65 0.65 0.65],'.')
ylabel('Nomalized Output Displacement (y/Hc)')
xlabel('Sampling Time Step  (k)')
hold off
pg1=4;
figure(3)
[z1,z2]=size(tm);
for q=1:pg1
  subplot(pg1,1,q)
  axis([tm(1,q)-dt tm(z1,q) -0.15 1])
  hold on
  plot(tm(:,q),yy(:,q).*100)
end
xlabel('Actual Time(second)')
hold off
figure(4)
plot(u.*100,'+')
hold on
plot(ustr.*100,'.')
text(30,0.6,'Reference Input')
text(30,0.75,'Modified Intput')
plot([30 32 34],[0.8 0.8 0.8],'+')
plot([30 32 34],[0.65 0.65 0.65],'.')
ylabel('Nomalized Input Displacement (yc/Hc)')
xlabel('Sampling Time Step  (k)')
hold off
```

fig45_49.m (Case 2)

```
% Repettitive Cntrl of Linear Sys
% Adaptive Control with
%    Least Square Method
% Priodic Disturbance presented
clear
global H Hc
Kf=315;Ks=15;M=0.1;c=0.5;
Hc=0.01;H=Kf/(Ks+Kf)*Hc;
Ac=[0 1;-(Kf+Ks)/M -c/M];Bc=[0 Kf/M]';
C=[1 0];
[ai,aj]=size(Ac);[bi,bj]=size(Bc);
[ci,cj]=size(C);
%% Control Parameters %%
w=2*pi;prd=2*pi/w;
p=50;
gain=0.4;
alpha=1;
mxrpt=50;
mxitr=1;
intvl=10;
Agss=zeros(ci*p,aj);
Bgss=eye(ci*p,bj*p).*5;
sz=aj+ci*p;
R=eye(sz,sz).*1e40;
Q=eye(p).*1;
S=eye(p).*0.99;
%%%%%%%%%%%%

dt=prd/p;
[A,B]=c2d(Ac,Bc,dt);
for k=1:p
  ystr(k,1)=y345p(w*dt*k);
  ustr(k,1)=y23p(w*dt*(k-1));
  dstb(k,1)=H/20*sin(20*pi*k/p);
end
figure(1)
hold on
plot(dstb(1:p)./Hc)
hold off
q1=0;q2=0;
u=ustr;
x=[0 0]';
for jj=1:mxrpt
  jj
  if rem(jj,intvl)==1
    q1=q1+1;
    q2=0;
  end
  q2=q2+1;
  xo=x;
  for k=1:p
    x=A*x+B.*u(k);
    y(k,1)=C*x+dstb(k,1);
    tm((q2-1)*p+k,q1)=((jj-1)*p+k)/p*prd;
    yy((q2-1)*p+k,q1)=y(k,1);
  end
  for itr=1:mxitr
    k1=1:aj;
    k2=1:ci*p;
    PP(:,k1)=Agss(:,k1);
    PP(:,k2+aj)=Bgss(:,k2);
    UU(1,k1)=xo(k1)';
    UU(1,k2+aj)=u(k2)';
    err=y-PP*UU';
    dnm=alpha+UU*R*UU';
    PP=PP+err*(UU*R)./dnm;
    R=R-(R*UU'*UU*R)./dnm;
    Agss(:,k1)=PP(:,k1);
    Bgss(:,k2)=PP(:,k2+aj);
  end
du=(Bgss'*Q*Bgss+S)\Bgss'*Q*(ystr-y-Agss*(x-xo)
);
%  du=Bgss\(ystr-y-Agss*(x-xo));
  u=u+gain.*du;
  yold=y;

  if jj==1
    figure(2)
    plot(y.*100)
    hold on
        plot(ystr.*100,'.')
        text(30,0.6,'Desired Output')
        text(30,0.75,'Actual Output')
        plot([30 32 34],[0.8 0.8 0.8])
        plot([30 32 34],[0.65 0.65 0.65],'.')
        ylabel('Nomalized Displacement (y/Hc)')
        xlabel('Sampling Time Step  (k)')
        hold off
  end
end
hold off
pg1=5;
figure(3)
[z1,z2]=size(tm);
for q=1:pg1
  subplot(pg1,1,q)
  axis([tm(1,q)-dt tm(z1,q) -0.15 1])
  hold on
  plot(tm(:,q),yy(:,q).*100)
end
xlabel('Actual Time(second)')
hold off
figure(4)
plot(y.*100)
hold on
```

```
plot(ystr.*100,'.')
text(30,0.6,'Desired Output')
text(30,0.75,'Actual Output')
plot([30 32 34],[0.8 0.8 0.8])
plot([30 32 34],[0.65 0.65 0.65],'.')
ylabel('Nomalized Output Displacement (y/Hc)')
xlabel('Sampling Time Step  (k)')
hold off
figure(5)
plot(u.*100,'+')
hold on
plot(ustr.*100,'.')
text(30,0.6,'Reference Input')
text(30,0.75,'Modified Intput')
plot([30 32 34],[0.8 0.8 0.8],'+')
plot([30 32 34],[0.65 0.65 0.65],'.')
ylabel('Nomalized Input Displacement (yc/Hc)')
xlabel('Sampling Time Step (k)')
hold off
```

fig49_12.m (Case 3)

```
% Repettitive Cntrl of Linear Sys
% Adaptive Control with
%    Least Square Method
%    Non-repeating Disturbance presented
clear
global H Hc
Kf=315;Ks=15;M=0.1;c=0.5;
Hc=0.01;H=Kf/(Ks+Kf)*Hc;
Ac=[0 1;-(Kf+Ks)/M -c/M];Bc=[0 Kf/M]';
C=[1 0];
[ai,aj]=size(Ac);[bi,bj]=size(Bc);
[ci,cj]=size(C);

%% Control Parameters %%
w=2*pi;prd=2*pi/w;
p=50;
gain=0.4;
alpha=1;
mxrpt=100;
mxitr=1;
intvl=10;
Agss=zeros(ci*p,aj);
Bgss=eye(ci*p,bj*p).*5;
sz=aj+ci*p;
R=eye(sz,sz).*1e40;
Q=eye(p).*1;
S=eye(p).*3.50;
%%%%%%%%%%%%%

dt=prd/p;
[A,B]=c2d(Ac,Bc,dt);
for k=1:p
  ystr(k,1)=y345p(w*dt*k);
  ustr(k,1)=y23p(w*dt*(k-1));
  dstb(k,1)=H/15*cos(8.75*pi*k/p);
end
figure(1)
plot(dstb/Hc)
xlabel('Sampling Time Step (k)')
ylabel('Magnitude of Disturbance')
q1=0;q2=0;
u=ustr;
x=[0 0]';
for jj=1:mxrpt
  jj
  if rem(jj,intvl)==1
    q1=q1+1;
    q2=0;
  end
  q2=q2+1;
  xo=x;
  for k=1:p
    x=A*x+B.*u(k);
    y(k,1)=C*x+dstb(k,1);
    tm((q2-1)*p+k,q1)=((jj-1)*p+k)/p*prd;
    yy((q2-1)*p+k,q1)=y(k,1);
  end

  for itr=1:mxitr
    k1=1:aj;
    k2=1:ci*p;
    PP(:,k1)=Agss(:,k1);
    PP(:,k2+aj)=Bgss(:,k2);
    UU(1,k1)=xo(k1)';
    UU(1,k2+aj)=u(k2)';
    err=y-PP*UU';
    dnm=alpha+UU*R*UU';
    PP=PP+err*(UU*R)./dnm;
    R=R-(R*UU'*UU*R)./dnm;
    Agss(:,k1)=PP(:,k1);
    Bgss(:,k2)=PP(:,k2+aj);
  end
du=(Bgss'*Q*Bgss+S)\Bgss'*Q*(ystr-y-Agss*(x-xo)
);
%  du=Bgss\(ystr-y-Agss*(x-xo));
  u=u+gain.*du;
  yold=y;
  if jj==1
    figure(2)
    plot(y.*100)
    hold on
        plot(ystr.*100,'.')
        text(30,0.6,'Desired Output')
        text(30,0:75,'Actual Output')
        plot([30 32 34],[0.8 0.8 0.8])
```

```
        plot([30 32 34],[0.65 0.65 0.65],'.')
        ylabel('Nomalized Displacement (y/Hc)')
        xlabel('Sampling Time Step  (k)')
        hold off
  end
end
hold off
pg1=5;
figure(3)
[z1,z2]=size(tm);
for q=1:pg1
  subplot(pg1,1,q)
  axis([tm(1,q)-dt tm(z1,q) -0.15 1])
  hold on
  plot(tm(:,q),yy(:,q).*100)
end
xlabel('Actual Time (second)')
hold off
figure(4)
[z1,z2]=size(tm);
for q=pg1+1:q1
  subplot(q1-pg1,1,q-pg1)
  axis([tm(1,q)-dt tm(z1,q) -0.15 1])
  hold on
  plot(tm(:,q),yy(:,q).*100)
end
xlabel('Actual Time (second)')
hold off
figure(5)
plot(y.*100)
hold on
plot(ystr.*100,'.')
text(30,0.6,'Desired Output')
text(30,0.75,'Actual Output')
plot([30 32 34],[0.8 0.8 0.8])
plot([30 32 34],[0.65 0.65 0.65],'.')
ylabel('Nomalized Output Displacement (y/Hc)')
xlabel('Sampling Time Step  (k)')
hold off
figure(6)
plot(u.*100,'+')
hold on
plot(ustr.*100,'.')
text(30,0.6,'Reference Input')
text(30,0.75,'Modified Intput')
plot([30 32 34],[0.8 0.8 0.8],'+')
plot([30 32 34],[0.65 0.65 0.65],'.')
ylabel('Nomalized Input Displacement (yc/Hc)')
xlabel('Sampling Time Step  (k)')
hold off
```

Simulation of the nonlinear system

mcfsys1.m

```
clear all
global a1 a2 a3 a4 a5 a6 ad b1 b2 b3
global too H Hc

% System parameters of Motor-Cam %
Km=0.023;Kp=0.2*500000;Kd=10;Kb=0.0318;Kf=
300;Ks=15;
Ng=0.1;Rm=3.7;Jm=4e-6;Jc=1e-6;Mf=0.025;M=0.
1;
Sp=10;c=0.5;Hc=0.01;H=Kf/(Kf+Ks)*Hc;
Jo=Rm*(Jm+Jc*Ng^2)/Km;
Ro=Rm*Ng^2/Km;
% System parametr of Cam Fllwr %
too=1;
%%% System Coefficients
a1=Kp/Jo; a2=Kb/Jo; a3=Ro/Jo;
a4=Mf; a5=Kf; a6=Sp;
ad=Kd/Jo;
b1=c/M;
b2=(Kf+Ks)/M;
b3=Kf/M/Hc;
```

mcfsys2.m

```
function xdot=mcfsys2b(t,x)
% Dynamic system equation %
global a1 a2 a3 a4 a5 a6 ad b1 b2 b3
global H Hc t1 t2 u1 u2 Tc;
xdot=zeros(4,1);

dt=t2-t1;
ud=(u2-u1)/dt;
u=ud*(t-t1)+u1;
ycd=yd23p(x(1));
Tc=  a4*ycd*ydd23p(x(1))*x(2)^2;
Tc=Tc+a5*ycd*(y23p(x(1))-Hc*x(3));
Tc=Tc+a6*ycd;
xdot(1)=x(2);

xdot(2)=a1*(u-rem(x(1),2*pi))+ad*(ud-x(2))-a2*x(
2)-a3*Tc;
xdot(3)=x(4);
xdot(4)=(-b1)*x(4)-b2*x(3)+b3*y23p(x(1));
```

fig.51_54.m

```
%%% Piecewise Repetitive Control %%%
%%      of Non-Linear System       %%
%          Motor-Cam-Follower        %
```

```
% I   /----\
% I  /|   |\
% I / |   | \
% I/  |   |  \
% I/  |   |   \     /
% I------------------------
% x11  x12  x13  x14   x21 ...

mcfsys1
global too t1 t2 u1 u2

%%% Progarm Control Parameters
p=50;
mxrpt=10;
gain=0.99999;
dmmy=1;
%%% State Space Matrix

C=[0 0 1 0];
prd=1; prd=prd/too;
dt=prd/p;
w=2*pi/prd;

%%% Desired output & reference input %%%
for k=1:p
  tm(k,1)=dt*(k);
  ystr(k,1)=y345p(w*dt*k)/Hc;
  ustr(k,1)=(w*dt*(k-1));
  uc(k,1)=y23p(ustr(k,1));
end
time=tm*too;

%%% Identifying the Regions %%%
to(1)=0;
to(2)=90/180*pi/w;
to(3)=126/180*pi/w;
to(4)=216/180*pi/w;
no(1)=0;
for k=1:p
  if abs(tm(k)-to(2))<dt
    no(2)=k;
  elseif abs(tm(k)-to(3))<dt
    no(3)=k-2;
  elseif abs(tm(k)-to(4))<dt
    no(4)=k;
  end
end

%%% Initial Guesses
ia=no(2)-no(1);
ic=no(4)-no(3);
A1=zeros(ia,4); B1=eye(ia,ia);
A3=zeros(ic,4); B3=eye(ic,ic);
Q1=eye(ia);S1=eye(ia).*0.05;
Q3=eye(ic);S3=eye(ic).*0.05;
R1=eye(ia+4,ia+4).*1e30;
R3=eye(ic+4,ic+4).*1e30;

%%% Cnotrol, Measure & Id.
x=[0 0 0 0]';
x2(:,1)=x;
u=ustr;
for rpt=1:mxrpt

 %%% Rise & Dwell %%%
  i1=[no(1)+1:no(2)]';
  if rpt>1
    BB=(B1'*Q1*B1+S1)\B1'*Q1;
    du1=BB*(ystr(i1)-y(i1)-A1*(x2(:,1)-x1(:,1)));
    u(i1)=u(i1)+gain.*du1;
  else
    u(i1)=ustr(i1);
  end

  for i2=no(2)+1:no(3);
    u(i2)=u(i2-1)+w*dt;
  end
  x1(:,1)=x2(:,1);
  for k=no(1)+1:no(3)
    t1=tm(k)-dt;t2=tm(k);
    u1=u(k);u2=u(k);
    [tt,xx]=ode23('mcfsys2',t1,t2,x);
    [cl,rw]=size(tt);
    x=xx(cl,:)';
    y(k,1)=C*x;
    uc(k,1)=[1 0 0 0]*x;
    proc=[rpt k u(k) uc(k) y(k) ystr(k)]
  end
  x2(:,3)=x;
  phi1=[x2(:,1)' u(i1)']';
  theta1=[A1 B1];
  err=y(i1)-theta1*phi1;
  dnm1=1+phi1'*R1*phi1;
  theta1=theta1+err*phi1'*R1/dnm1;
  R1=R1-R1*phi1*phi1'*R1/dnm1;
  A1=theta1(:,1:4); B1=theta1(:,i1+4);

  %%% Return & Dwell %%%
  i3=no(3)+1:no(4);
  if rpt>2
    BB3=(B3'*Q3*B3+S3)\B3'*Q3;
    du3=BB3*(ystr(i3)-y(i3)-A3*(x2(:,3)-x1(:,3)));
    u(i3)=u(i3)+gain.*du3;
  else
    for k=no(3)+1:no(4)
```

```
      u(k)=u(k-1)+w*dt;
    end
  end
  for i4=no(4)+1:p
    u(i4)=u(i4-1)+w*dt;
  end
  x1(:,3)=x2(:,3);
  for k=no(3)+1:p
    t1=tm(k)-dt;t2=tm(k);
    if k==p
      u1=u(p);u2=u(p)+w*dt*0;
    else
      u1=u(k);u2=u(k);
    end
    [tt,xx]=ode23('mcfsys2',t1,t2,x);
    [cl,rw]=size(tt);
    x=xx(cl,:)';
    y(k,1)=C*x;
    uc(k,1)=[1 0 0 0]*x;
    proc=[rpt k u(k) uc(k) y(k) ystr(k)]
  end
  x2(:,1)=x;
  if rpt>1
    phi3=[x2(:,3)' u(i3)']';
    theta3=[A3 B3];
    err=y(i3)-theta3*phi3;
    dnm3=1+phi3'*R3*phi3;
    theta3=theta3+err*phi3'*R3/dnm3;
    R3=R3-R3*phi3*phi3'*R3/dnm3;
    A3=theta3(:,1:4); B3=theta3(:,i3-no(3)+4);
  end

 %%% Plotting 'y' %%%
  figure(1)
  if rpt==1
    plot(y)
    hold on
    plot(y,'+')
    hold off
  end
  yy(:,dmmy)=y;
  dmmy=dmmy+1;
end
figure(2)
row=mxrpt/10;
clm=10;
for i=1:row
  for j=1:clm
    k=(i-1)*clm+j;
    subplot(row,1,i)
    axis([time(1,(i-1)*clm+1) time(p,i*clm) -0.12
1])
    hold on
    plot(time(:,k),yy(:,k));
  end
  hold off
end
xlabel('Actual time (second)')
figure(3)
plot(y)
hold on
plot(y)
plot(ystr,'.')
text(30,0.6,'Desired Output')
text(30,0.75,'Actual Output')
plot([30 31 32 33],[0.8 0.8 0.8 0.8])
plot([30 31 32 33],[0.65 0.65 0.65 0.65],'.')
ylabel('Nomalized Displacement (y/Hc)')
xlabel('Sampling Time Step  (k)')
figure(4)
plot(u./pi.*180,'.')
hold on
plot(ustr./pi.*180,':')
plot([30 33],[100 100])
plot([30 31 32 33],[60 60 60 60],'.')
text(30,85,'Original Input')
text(30,45,'Corrected Input')
xlabel('Discrete Time Step (k)')
ylabel('Angular Input of Motor (degree)')
hold off
```

Vita

The author was born in Pusan, Korea on November 18, 1970, to Mr. Sun-Duck Chang and Hee-Ja Ahn. He graduated from The Pusan Jung-Ang High School in the winter of 1989 and continued his education at Hanyang University in Seoul, Korea, where he received a Bachelor of Science degree in the Department of Machine Design and Production Engineering in the winter of 1994. The author continued his education at Lehigh University and received a Masters of Science degree in the Department of Mechanical Engineering in the summer of 1996. He plans to continue his study at Pennsylvania State University to obtain a Ph. D. in Mechanical Engineering.

END OF TITLE