American Journal of Applied Sciences 2 (5): 897-903, 2005
ISSN 1546-9239

Morphing Mechanisms Part 1: Using Iterative Learning Control to Morph Cam Follower Motion

[1]Nonglak Phetkong, [2]Meng-Sang Chew and [3]Richard W. Longman
[1]The National Metal and Materials Technology Center, Patumthani, Thailand 12120
[2]Department of Mechanical Engineering and Mechanics, Lehigh University
Bethlehem, PA 18015 USA
[3]Department of Mechanical Engineering, Columbia University
New York, New York 10027 USA

Abstract: This study introduces the concept of morphing mechanisms. Mechanisms are normally designed for specific operating conditions and once they are built one may wish to use them under different operating conditions. In some cases, a mechanism may be imperfectly fabricated and one would like to get the intended ideal performance. In these cases, instead of designing, fabricating and replacing the original mechanism, the behavior of the existing hardware can be morphed to make it function as if it was the redesigned or re-fabricated system. This concept is illustrated in both simulation and experiments for cam mechanisms. Iterative learning control is used on a cam designed and built using a 2-3 polynomial profile and it is made to function like a cam designed with a 3-4-5 polynomial. Eight cycles of learning are seen to be sufficient to effectively accomplish this morphing of the calm behavior.

Key words: Modern Control, Cam Mechanism, Tracking Error Reduction

INTRODUCTION

Mechanisms are designed to perform a unique motion characteristic. Once they are built the mechanism may be called upon to perform in a different operating regime such as operating at a different speed, or to have some different motion characteristic. Under normal circumstances, this would require the redesign, fabrication and installation of a new cam. This study develops the concept of morphing mechanisms, which morphs the motion characteristics of the original hardware to match those of the new mechanism. This is accomplished through modifying the actuation characteristics of the hardware mechanism.

The concept of morphing can be applied to many classes of mechanisms, including multi-degree of freedom mechanisms such as a five bar, or mechanisms with higher-order pairs such as cams, swash plates and band mechanisms. In this study we illustrate the concepts of morphing using cam-follower systems. The desired morphing can take many forms. One possibility and the one treated in detail in this article, is morphing the motion characteristics from a 2-3 polynomial cam so that it operates as a different cam, such as a 3-4-5 polynomial cam. This example corresponds to morphing a given cam to achieve the operating characteristics of a good cam without actually replacing the original cam. The morphing concept can also be applied when one wants to change the operating speed of the hardware. The cam can be designed for one operating speed, but one can morph its operating characteristics to correspond to one designed for the new desired speed. Yet another application has been to morph the behavior of a cam system that has imperfect fabrication and/or installation errors and make it operate as intended. Morphing mechanisms can be considered as a subset of intelligent mechanisms[1] that have the ability to adapt or respond to changes in the environment or new or non-ideal operating conditions.

A very large class of mechanisms use designs made for a specific operating speed and some form of control system is used that aims to maintain this speed. By using intelligent control concepts to make the control system vary the speed in an appropriate way, one has new freedom to modify mechanism performance. With market competition there is pressure for continually improving the performance of products and this new freedom allows one to improve performance in many situations as described above. What is required is an extra sensor and a modified control algorithm. It is suggested that taking advantage of this new freedom will become important in the future. New generations of mechanisms may be designed deliberately intending to make use of the morphing ability to extend the best performance characteristics of a broad operating environment. The work presented here comes from the doctoral dissertation[2].

CAM MORPHING

Cam mechanisms have been a topic that has been studied for well over half a century. A comprehensive and a long list of articles on cams can be found in the

state-of-art article by Chen[3]. A critical examination of the articles on cam mechanisms shows that many assumptions are used implicitly and explicitly in the process of analysis and design of cam mechanisms. Furthermore, most of the articles are theoretical in nature and there is very little experimental verification of the theoretical developments. This article introduces the concept of cam morphing which frees one from being encumbered by many of the assumptions inherent in prior investigations on cam dynamic systems. The theory behind this concept will be provided along with experimental verification of the concept. We will begin by posing the following problem.

Morphing Cams: Consider the following objective as an illustration: morph a cam mechanism that has a hardware cam designed for a Dwell-Rise-Dwell (DRD) motion using a 2-3 polynomial based on the polydyne approach[4] and make this mechanism behave as if the cam were replaced by a 3-4-5 design. Thus, the cam starts with the output curve as a function of cam angle given by:

$$y_c(\theta) = H_c(3\xi^2 - 2\xi^3) \quad (1)$$

Where, $\xi = \theta / \beta_1$, θ is the cam angle, β_1 is the rise angle and H_c is the maximum cam lift. The morphing is to make the hardware function as if the cam were replaced by another cam satisfying:

$$y^* = H(10\xi^3 - 15\xi^4 + 6\xi^5) \quad (2)$$

Where, y^* is the desired lift curve of the output mass, $\xi = \theta / \beta_1$ and H is the maximum lift of the output mass.

If the cam speed is *constant* throughout its operating cycle, the effective follower lift will exactly trace the cam lift function and the follower lift function will also be a 2-3 polynomial. However, if the cam speed *varies* over its operating cycle, the effective follower lift as a function of time will no longer be a 2-3 polynomial, but will be a different function that depends on how the cam speed varies. Suppose that we want that function to be a 3-4-5 polynomial desired follower lift function (y^*). To achieve this task, we vary the cam speed over the duration of the cycle such that the desired follower lift function is physically realized.

The methodologies used to determine the cam speed as it varies over the duration of the cycle is Iterative Learning Control (ILC). This has been given by Chew and Phan[5, 6] from the perspective of addressing inverse kinematics problems and this application will therefore not be repeated in this study.

Advantages of Morphing Cam Mechanisms: Morphing cams permit the engineer to bypass many of the difficulties and deficiencies inherent in the analysis, design, fabrication, implementation and operation of high-speed cam-follower systems. Some of these include design assumptions, manufacturing errors, parametric errors in the analysis, installation errors, modeling simplifications of the cam system that result in incomplete modeling, unmodelled dynamics of control systems, operating at off nominal design speeds, presence of friction and a whole host of other effects that are difficult to model and quantify.

By morphing the cam we can make high-speed cam-follower systems behave in a way that was originally intended, such as extinguishing the follower residual vibrations, even in the midst of all the unknowns and errors listed above. That is certainly an important accomplishment. However, morphing can do more. It can make the high-speed cam-follower system behave in a way that was *not* originally intended. One such example is operating the high-speed cam at another cam speed other than at its originally designed speed and can still extinguish all the residual vibrations. In other words, by morphing the cam, we can extinguish residual vibrations for *all operating speeds* using the same physical cam.

CAM MORPHING USING ITERATIVE LEARNING CONTROL (ILC)

The morphing is to be accomplished in software, iteratively adjusting the command to the control system that maintains cam speed. If the control system is digital, its microprocessor can perform this function as well. It is to be accomplished without any need to model the system, simply making intelligent iterations with the real world to converge to the desired output behavior. Iterative Learning Control (ILC) and Repetitive Control (RC) are two relatively new, closely related fields that imitate learning by practice in humans[7-9]. They create control systems that learn from previous experience trying to perform a specific command. In ILC, each run is started from the same initial condition, while in RC the command is a periodic function and one learns from the errors observed in the previous periods. In this study we use a simple ILC law to learn both the rise and the return and create a somewhat nonstandard law to handle the dwell sections of the cam.

The simplest form of ILC can be conceptualized as follows: if the output was 2 units too low in the last run or cycle at a certain time step, add 2 units to the command this run, or add some learning gain *G* times 2 units for this run. Since the learning requires recording the output of each run, the implementation is necessarily digital. Most physical systems require one time step between the time the input command is changed and the first time step for which there is a change in the output. This means that one should look at the error in the previous run one time step ahead of the current step when computing the control in the

current run. One can prove that this learning law is guaranteed to converge to zero tracking error for almost all systems, including nonlinear systems[9,10]. However, it can often have very poor learning transients[9]. The usual fix for this problem is to introduce a zero-phase low-pass filter that cuts off the high speed learning[11] and then to raise the cutoff frequency one can introduce a compensator. Perhaps the simplest compensator is a linear phase lead[9,12], which corresponds to simply looking more than one time step ahead of the error in the previous repetition. In this work we use a total of 4 time steps ahead, one of causality and three corresponding to what is called a linear phase lead. There are two simplifying aspects of how the simulations and experiments were run. One is that the sample time is chosen sufficiently long that there was no need for a zero-phase filter to obtain good learning transients. Secondly, no feedback control system was used to control the cam velocity. Instead, the ILC simply learned the voltage history to a DC motor. Thus, the learning law for the rise and the return cam segments is:

$$v^{(i+1)}(kT) = v^{(i)}(kT) + G[y^*((k+4)T) - y^{(i)}((k+4)T)] \quad (3)$$

Here, t = kT is the time of the k^{th} time step, i indicates the i^{th} repetition or cycle, starting with 0 for the first run before learning can start. The symbol y^* is the desired output displacement and $y^{(I)}$ the measured output displacement in repetition myself. The voltage applied to the motor is designated by v. The learning gain G can be made time dependent. The voltage values between times it and (k+1) T are produced by linear interpolation.

During the dwell cam segments, the objective of the learning is different. Instead of a specific rise curve to be followed, the aim is to ensure that the start of the next phase, either return or rise, occurs at the right time. To iteratively accomplish this, the following learning law is used:

$$v^{(i+1)}(kT) = v^{(i)}(kT) + G_d\left[v^* - \left(\frac{\theta^{(i)}(t_{end}) - \theta^{(i)}(t_{start})}{t_{end} - t_{start}}\right)\right] \quad (4)$$

A learning gain G_d is used, the t_{end} and t_{start} are the desired end and start times respectively of the dwelling and $\theta^{(i)}(t)$ is the cam angle at time t in the i^{th} repetition or cycle. This law compares the average velocity from t_{start} to t_{end} in the last cycle and the velocity needed to get the next segment start on time, here $v^* = 2*\pi$ rad/sec and changes the voltage for all dwell time steps by the same constant. The v^* is the cam desired velocity if the dwell angle is correct. There are 20 time intervals during the rise and return segments, 8 intervals during the top dwell and 16 during the first and last dwell segments (a total of 80).

Theoretical Model: The overall nonlinear electro-mechanical system model is shown in Fig. 1. The input to the system is the voltage to the motor and the output of the system is the displacement of the output mass. The cam follower system incorporates *torsion and bending* in the camshaft as well as compliance in the follower.

Simulation Results: The system equations of motion are listed in Appendix A and the simulation parameters are shown in Table 1 in Appendix B. Note that in all the figures, y^* refers to the desired output and y_a refers to the actual output. In addition, v refers to the voltage input, i.e., v_0 refers to the voltage input from the initial cycle and v_1 refers to the voltage input from cycle #1. During the initial cycle (designated by cycle #0), a constant voltage input is applied to the DC motor. The corresponding initial actual output displacement, shown by the dashed line in Fig. 2, greatly deviates from the desired output displacement. In the next two cycles, ILC is applied to the first dwellers in order to get the actual rise segment to start at the same time as the desired segment, a timing requirement. The learning algorithm during this dwell is given by Eq. 4. The actual output displacement from cycle #2, (dotted line in Fig. 2), starts the rise segment at the same time as the desired output. At this point, the learning process for the first bottom dwell stops and its gain is set to zero.

The next step is to apply ILC to the rise segment and the input modification follows according to Eq. 3. Figure 3 shows results during this learning process, which morphs the cam. The actual output displacements from cycles #3 and 14 are shown to illustrate the morphing process and how they are being morphed as compared to the desired output displacement function. At the end of the learning process, the actual rise segment closely tracks the desired rise segment. The two lines (solid and dotted) are basically on top of each other. Once this goal is achieved for the rise segment, the morphing process for this segment terminates. We then begin the process of learning the top dwell segment.

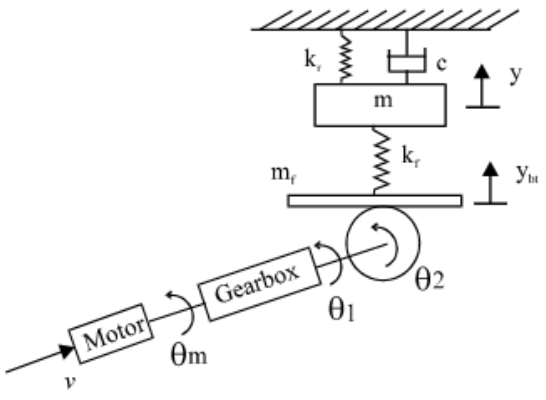

Fig. 1: An Overall Electromechanical System

Table 1: Parameters Used in Theoretical Simulations

k_m (motor torque constant)	0.023 N-m/amp
k_b (motor back emf)	0.0318 V-s/rad
R_m (armature resistance)	3.7 Ohm
I_m (armature and gear inertia)	4.0×10^{-6} N-m s^2
N_g (gear ratio)	0.01
L (armature inductance)	Negligible
S_p (preload on cam)	5.0 N
Number of intervals per cycle	80
Period	2 seconds
Learning gain (G)	First dwell: 500 Rise: 1500 Top dwell: 500 Return: -1500

Table 2: Parameters Used in Experiments

Camshaft length (l)	0.102 m
Camshaft radius (r)	0.00635 m
Output mass (m)	1.02 kg
Follower mass (m_f)	0.025 kg
Cam mass (m_c)	0.2 kg
Follower stiffness (k_f)	1,283 N/m
Return spring stiffness (k_r)	N/A
Cam lift (H_c)	5 mm
Cycle period	2 seconds
Number of intervals in each cycle	80
Learning gain (G)	First dwell: 0.4 Rise: 0.3 Top dwell : 0.5 Return: -0.6

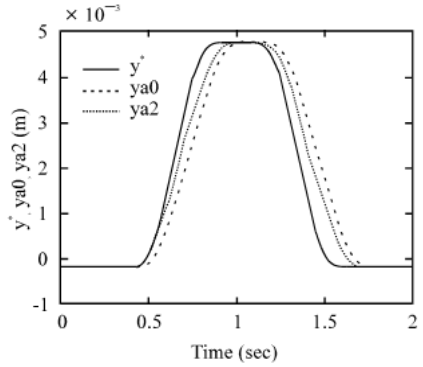

Fig. 2: Desired Output (y^*) and Actual Output Displacements from Cycles #0 and #2

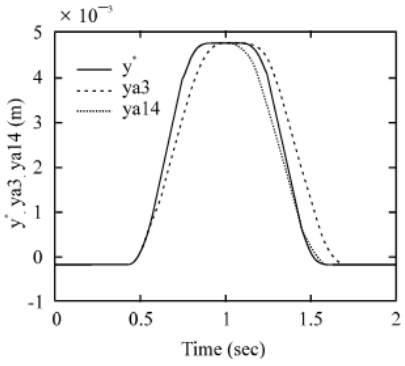

Fig. 3: Desired Output (y^*) and Actual Output Displacements from Cycles # 3 and 14

By closely examining the top dwell segment of the actual output displacement from cycle #14 in Fig. 3, one can see that the actual top dwell ends much earlier than the desired segment. Therefore, the learning algorithm given by Eq. 4 will be applied in order to slow down the motor. Figure 4 shows the results obtained from this learning process. The dotted line represents the actual output from cycle #17, where the top dwell has now been matched with the desired top dwell to an acceptable degree of accuracy. The learning process for the top dwell then stops and the gain is set to zero.

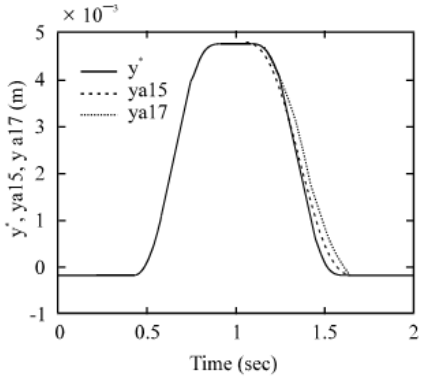

Fig. 4: Desired Output (y^*) and Actual Output Displacements from Cycles #15 and 17

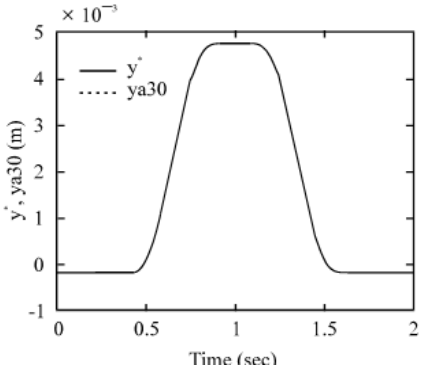

Fig. 5: Desired Output (y^*) and Actual Output Displacements from Cycle #30

The final part of the learning process deals with the return segment of the cam motion. At the beginning of this process, the actual rise is higher than desired. Therefore, the motor needs to speed up so that the actual return segment drops to match the desired return. The final actual output displacement, tracked the desired output displacement very accurately as shown in Fig. 5. Learning control, following the learning law given by Eqs. 3 and 4, has achieved its goal morphing the cam so that the actual output closely tracks the desired output displacement. Figure 6 gives the initial

constant voltage input to the motor and the final voltage input for cycle #30.

EXPERIMENTAL VERIFICATION

This section presents the experimental setup of the cam follower system and the results from a single experiment are presented here. Parameters for the physical experimental system are listed in Table 2 of Appendix B. In constructing the experimental setup, we have tried to maintain the same parameters as those used in the theoretical section. However, randomness and disturbances in the experimental setup cannot be exactly modeled and incorporated by the theory. We shall see that morphing the cam uses ILC will account for all these unknown disturbances as well as unknown errors in the system.

Experimental Setup: The experimental setup, shown in Fig. 7, consists of a cam-follower system, a DC motor, an encoder, a laser diode, a position sensing detector (PSD) and amplifier, a servo voltage amplifier, a power supply and a host personal computer. A *d* SPACE (Digital Signal Processing and Control Engineering) control hardware package is used. It consists of a floating-point processor board, a multi-I/O board and an incremental encoder board all of which are installed in the computer.

First, an initial input voltage is sent from the *d* SPACE Control Desk through the digital-to-analog (D/A) output board to the motor, which drives the system. The output displacement signal is read by the position sensing detector and is sent to the position sensing amplifier and then to the *d* SPACE analog-to-digital (A/D) input board and subsequently stored on the host computer. In the first run (cycle #0), the input is constant. The displacement, y_{a0} of the output mass is measured and the voltage input of the next cycle (cycle #1) is then calculated off line based on the input from cycle #0 and the tracking error between the actual output and the desired output displacements, using Eqs. 3 or 4 as appropriate. This new voltage input for cycle #1 is then sent to the system. The actual output displacement signal, y_{a1}, from cycle #1 is measured and a new voltage input is calculated. This process continues until the actual output displacement tracks the desired displacement to an acceptable level.

Experimental Results: This section presents experimental results morphing the charm of a highly nonlinear cam-follower system using ILC. Figure 8 shows the desired output displacement, which is compared to the actual output displacements (y^*) of cycle #0 (before learning control was applied) and cycle #1. Once the actual bottom dwell has tracked the desired dwell, learning control of this part is stopped and the learning gain is set to zero. Next ILC is applied to the rise segment. Figure 9 shows the actual output displacement from cycle #3, which is compared to the desired output displacement.

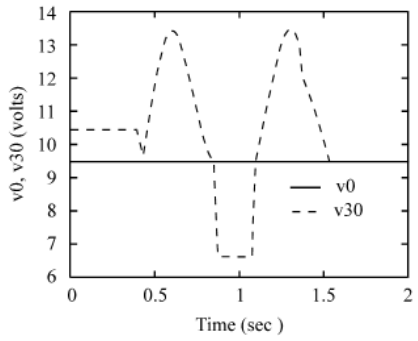

Fig. 6: Initial Voltage Input and Final Voltage Input

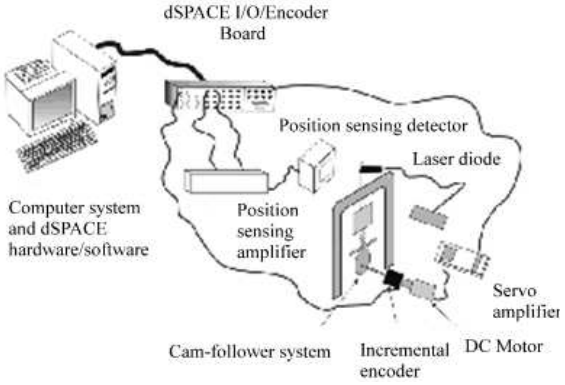

Fig. 7: Overall View of the Experimental Setup

Just as in the theoretical section, the next step is to apply ILC to the top dwell and then the return segments. There is no significant input modification during the top dwell since the actual and desired outputs are almost the same. ILC has slowed down the motor during the return segment so that the actual output tracks the desired output trajectory. Figure 10 shows the final output and the desired output displacements. The two curves are almost the same with some deviation at the last bottom dwellers, where a dip in the actual output is present. This dip is a manufacturing imperfection. The radius of the fabricated cam is not constant as desired and changing the speed of rotation cannot correct this error. Figure 11 shows the initial constant voltage and the final voltage history.

Figure 12 presents a study of repeatability by applying the same constant voltage input six times and the average error for the six runs, using the Root-Mean-Square (RMS) calculation, is shown by the solid line in Fig. 12. No learning process can reach a final error level that is consistently below this repeatability level. Clearly the learning process here has approached this theoretical performance limit. The error is significant during the rise and return segments when there are dynamic effects of the springs, the moving mass, as well as some random friction effects. The final experimental error, shown by the dashed line, is compared to the repeatability error.

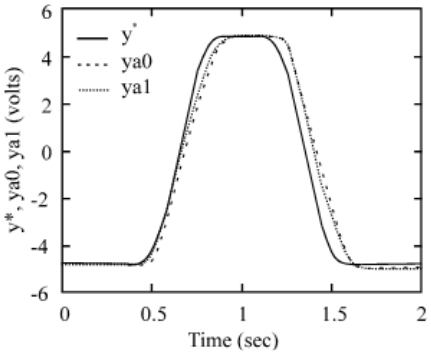

Fig. 8: Desired Output (y^*) and Actual Output Displacements from Cycles #0 and #1

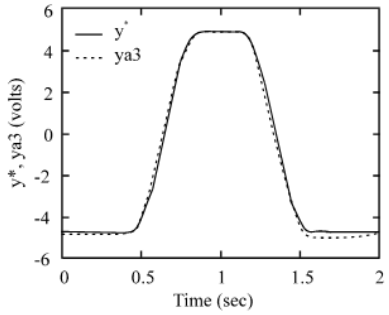

Fig. 9: Desired Output (y^*) and Actual Output Displacement from Cycle #3

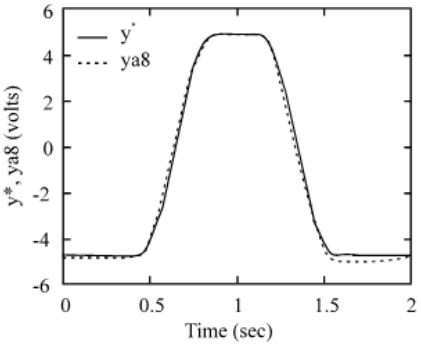

Fig. 10: Desired Output (y^*) and Actual Output Displacement from Cycle #8

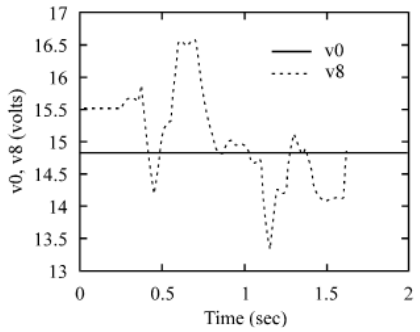

Fig. 11: Voltage Inputs from Cycles #0 and 8

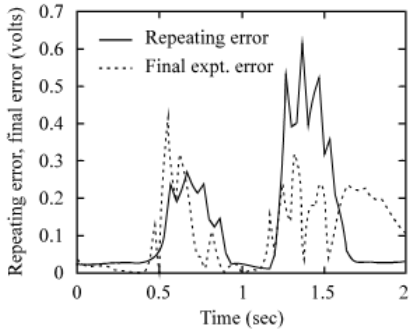

Fig. 12: RMS Values of Repeating and Final Experimental Errors

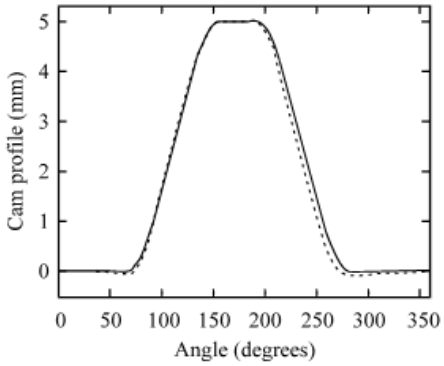

Fig. 13: Deviation of Ideal Cam Profile (Solid Line) and Actual Cam Profile (Dashed Line)

It can be concluded that the experimental results are acceptable since the repeatability error voltage is small relative to the input voltage. As a check for manufacturing errors, Fig. 13 shows the actual cam lift trajectory for this 2-3 polynomial cam and compares it to the ideal 2-3 curve. As mentioned earlier, it is difficult to manufacture a cam to exact specifications, even with the use of high precision NC cutting machinery and ILC can address many manufacturing imperfections.

CONCLUSION

This article presents a theoretical development and experimental verification of cam morphing mechanisms using iterative learning control. ILC is seen to be capable of morphing the hardware follower output motion characteristics so that it tracks the desired trajectory of a different cam design. This can be accomplished in the presence of fabrication and installation errors in the cam-follower system hardware. In addition, the morphing achieves the desired output behavior in spite of bending and torsional deflections, as well as compliance in the follower.

ILC does not require prior knowledge of a model nor knowledge of its parameters. Instead it treats the system as a black box. The processed input is iteratively modified based on prior input and the tracking error between the actual and desired output trajectories.

ACKNOWLEDGMENT

The authors would like to thank Lehigh University and the National Metal and Materials Technology Center of Thailand (MTEC) for their support.

Appendix A: System Equations: The equation of motion of the output mass (*y*-direction) is given as:

$$m\ddot{y} = -k_r y - c\dot{y} - k_f\left(y - y_c(\theta_2) - z\right) \quad (1)$$

Corresponding to torsion in the camshaft, the equation of motion is described as:

$$a_1\ddot{\theta}_2 = a_2 + a_3 + a_4 - a_5 \quad (2)$$

Where:

$$a_1 = I_c + 6m_f H_c \varepsilon\left(\frac{\theta_2}{\beta_1^2} - \frac{\theta_2^2}{\beta_1^3}\right) - \frac{6H_c m_f^2 \varepsilon}{m_c + m_f}\left(\frac{\theta_2}{\beta_1^2} - \frac{\theta_2^2}{\beta_1^3}\right) \quad (3)$$

$$a_2 = k_\theta(\theta_1 - \theta_2) - 6H_c m_f \varepsilon\left(\left(\frac{\dot{\theta}_2}{\beta_1}\right)^2 - 2\left(\frac{\theta_2}{\beta_1}\right)\left(\frac{\dot{\theta}_2}{\beta_1}\right)^2\right) \quad (4)$$

$$a_3 = \varepsilon\left(k_f\left(y - y_c(\theta_2) - z\right) - S_p\right) \quad (5)$$

$$a_4 = \frac{6H_c m_f^2 \varepsilon}{m_c + m_f}\left(\left(\frac{\dot{\theta}_2}{\beta_1}\right)^2 - 2\left(\frac{\theta_2}{\beta_1}\right)\left(\frac{\dot{\theta}_2}{\beta_1}\right)^2\right) \quad (6)$$

$$a_5 = \frac{m_f \varepsilon}{m_c + m_f}\left(k_f\left(y - y_c(\theta_2) - z\right) - S_p - k_z z - c\dot{z}\right) \quad (7)$$

The equation of motion in the bending, *z*-direction, is described as follows:

$$\ddot{z} = -\frac{(a_2 + a_3 + a_4 - a_5)}{a_1} b_1 - b_2 + b_3 \quad (8)$$

where, $$b_1 = \frac{6m_f H_c}{m_c + m_f}\left(\frac{\theta_2}{\beta_1^2} - \frac{\theta_2^2}{\beta_1^3}\right) \quad (9)$$

$$b_2 = \frac{6m_f H_c}{m_c + m_f}\left(\left(\frac{\dot{\theta}_2}{\beta_1}\right)^2 - 2\left(\frac{\theta_2}{\beta_1}\right)\left(\frac{\dot{\theta}_2}{\beta_1}\right)^2\right) \quad (10)$$

$$b_3 = \frac{1}{m_c + m_f}\left(k_f\left(y - y_c(\theta_2) - z\right) - S_p - k_z z - c\dot{z}\right) \quad (11)$$

And the equation of motion of the dynamic system is:

$$\ddot{\theta}_1 = -\frac{N_g^2 R_m T_c}{k_m p_1} - \frac{k_b \dot{\theta}_1}{p_1} + \frac{e_m N_g}{p_1} \quad (12)$$

where, $$p_1 = \frac{R_m\left(I_m + I_c N_g^2\right)}{k_m} \quad (13)$$

REFERENCES

1. Chew, M., R.W. Longman and M.Q. Phan, 2004. Intelligent Mechanisms. Submitted for Presentation at the ASME DETC, Salt Lake, UT.
2. Phetkong, N., 2002. Learning control and repetitive control of a high-speed, nonlinear cam follower system. Ph.D. Thesis. Lehigh Univeristy, Bethlehem, PA.
3. Chen, F.Y. and N. Polyvanich, 1977. A survey of the state of the art of cam system dynamics. Mech. Mach. Theory, 12 : 201-224.
4. Stoddart, D., 1953. Polydyne cam design. Machine Design, 25: 121-135.
5. Chew, M. and M. Phan, 1994. Application of learning control theory two mechanisms: Part 1: Inverse kinematics and parametric error compensation. Machine Elements and Machine Dynamics, 71: 25-32.
6. Chew, M. and M. Phan, 1994. Application of learning control theory two mechanisms: Part 2: Reduction of residual vibrations in high-speed electromechanical bonding machines. Machine Elements and Machine Dynamics, 71: 33-39.
7. Bien, Z. and J.X. Xu, 1998. Editors, Iterative Learning Control: Analysis, Design, Integration and Applications, Kluwer Academic Publishers, Boston.
8. Moore, K. and J.X. Xu, 2000. Guest Editors, Special Issue on Iterative Learning Control. Intl. J. Control, 73: 10.
9. Longman, R.W., 2000. Iterative learning control and repetitive control for engineering practice. Intl J. Control, Special Issue on Iterative Learning Control, 73: 930-954.
10. Longman, R.W., C.K. Chang and M. Phan, 1992. Discrete time learning control in nonlinear systems. A Collection of Technical Papers, 1992 AIAA/AAS Astrodynamics Specialist Conference, Hilton Head, South Carolina, pp: 501-511.
11. Elci, H., R.W. Longman, M.Q. Phan, J.N. Juang and R. Ugoletti, 2002. Simple learning controls made practical by zero-phase filtering: Applications to robotics. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Special Issue on Multidimensional Signals and Systems, Guest Editors: S. Basu and M.N.S. Swamy, 49: 753-767.
12. Wang, Y. and R.W. Longman, 1996. Use of Non-causal digital signal processing in learning and repetitive control. Adv. Astronautical Sci., 90: 649-668.