Proceedings of the 10th Mediterranean Conference
on Control and Automation - MED2002
Lisbon, Portugal, July 9-12, 2002.

EXPERIMENTAL RESULTS AND IMPLEMENTATION ISSUES OF AN ITERATIVE LEARNING CONTROLLER FOR SOFT LANDING OF AN ELECTROMECHANICAL VALVE ACTUATOR

Katherine Peterson, Wolfgang Hoffmann, Anna Stefanopoulou

Mechanical Engineering Dept., University of Michigan
e-mail: (kspeters,wolfh,annastef)@umich.edu

Keywords: Engine control, electromechanical actuators, observers, singular perturbations, iterative learning control

Abstract

Numerous studies and papers show that variable valve timing has the potential to significantly enhance the automotive engine. Improvements in fuel economy, emissions, and performance can be achieved, as well as cylinder deactivation, and the elimination of the throttle and external exhaust gas recirculation. Actuation employing electromagnets has been proposed as a replacement to the traditional camshaft to achieve variable valve timing. Unfortunately, the resulting ElectroMechanical Valve actuators suffer from large impact velocities between the valve, valve seat, and the actuator itself due to the motion of the valve, leading to excessive noise and wear on the system. Impact velocities comparable to those achieved by a camshaft are required before the system can be used in production vehicles. An Iterative Learning Controller (ILC) is applied to achieve this performance. This paper discusses the implementation issues associated with the ILC and presents experimental results.

1 Introduction

Global fuel resources and environmental considerations are pressuring the automotive industry to produce more fuel efficient vehicles. Throughout the last twenty years significant improvements have been made by replacing mechanical subsystems with their electronic counterparts. Digital control of spark timing, idle speed bypass valve, throttle, and air/fuel ratio have led to more fuel efficient vehicles. Despite research showing that variable valve timing has the potential to contribute to higher fuel economy, such a system has yet to be implemented. While several manufactures have implemented variable profile camshafts to achieve a degree of flexibility in valve timing, complete independent control of valve timing remains elusive and is being actively researched.

A promising solution to achieve variable valve timing is the ElectroMechanical Valve (EMV) actuator, shown in Figure 1.

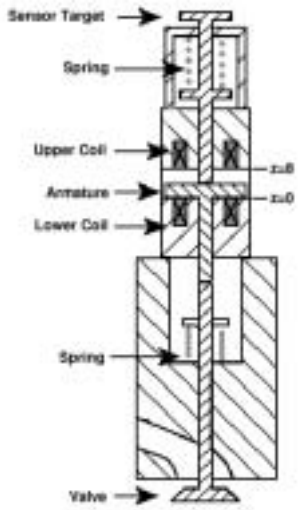

Figure 1: Schematic of the ElectroMechanical Valve (EMV) actuator

The EMV actuator governs the opening/closing of the valve through the forcing of a set of springs and electromagnets. Initially the armature is held in one of the two extreme positions by an electromagnet, causing the spring on that side to be more compressed than the opposing spring. When the voltage across the magnetic coil is reduced to zero, the difference in the spring force drives the armature across the 8 mm gap, thereby causing the valve to either open or close. As the armature nears the other extreme position the electromagnet is activated to catch and hold the armature in place.

While effective in ensuring opening/closing of the valve, the EMV actuator suffers from large impact velocities between the armature, valve, and valve seat leading to excessive noise and wear on the system. In addition, transition times need to be both quick and consistent to avoid collision with the piston and variability in trapped mass. Before the actuator can be implemented in production vehicles a solution to these problems needs to be found.

Due to the repetitive nature of the motion with fixed ini-

tial conditions, an Iterative Learning Controller (ILC) is well suited for this application. Although repetitive control approach is applied in [6] the resulting controller works well when the reference signal is periodic. This is not the case in engine variable valve timing applications because a new opening process depends on the driver signal and the engine management controller. The ILC, uses the position error from the previous transitions to modify the input of the next transition in order to improve the tracking of a given trajectory. By choosing a sufficiently smooth trajectory, as the tracking error goes to zero, so should the impact velocity.

The system model and control difficulties are summarized in Section 2. Section 3 presents an overview of ILC theory. Modifications due to implementation issues are presented in Section 4. Experimental results are found in section 5.

2 Model and control Difficulties

2.1 System model

A nonlinear system model is developed in [7] and later extended to include impact dynamics in [5]. The model consists of four states: the armature distance from the catching coil, z, the armature velocity, v, the current in the lower coil, i_l, and the current in the upper coil, i_u. To achieve fast release, a reverse pulsing technique is developed in [7] to drive the holding current to zero very quickly. Thus, the holding current has little influence on the motion of the armature. By assuming that valve openings and closings are identical, the four state system is redefined as the following three state system:

i	Catching coil current [A]
z	Distance from the catching coil [mm]
v	Armature velocity [m/s]

The resulting equations of motion are:

$$\frac{di}{dt} = \frac{V_c - ri + \chi_1(i,z)\,v}{\chi_2(z)} \quad (1)$$

$$\frac{dz}{dt} = 1000v \quad (2)$$

$$\frac{dv}{dt} = \frac{1}{m}\left(-F_{mag}(i,z) + k_s(4-z) - bv\right) \quad (3)$$

where

$$\chi_1(i,z) = \frac{2k_a i}{(k_b+z)^2} \quad (4)$$

$$\chi_2(z) = \frac{2k_a}{1000(k_b+z)} \quad (5)$$

$$F_{mag}(i,z) = \frac{k_a i^2}{(k_b+z)^2} \quad (6)$$

written compactly as

$$\frac{dx}{dt} = f(x, V_c) \quad (7)$$

$$x = \begin{bmatrix} i & z & v \end{bmatrix}^T \quad (8)$$

$$y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} x \quad (9)$$

where V_c is the catching coil voltage [V], r is the resistance [Ω], m is the system mass [kg], k_s is the spring stiffness [N/mm], b is the damping [Ns/m], F_{mag} is the magnetic force due to the catching coil [N], $\chi_1 v$ is the back EMF [V], and χ_2 is the inductance [H], and y is the output.

2.2 Control difficulties

The main difficulties stem from a lack of control authority over the armature motion away from the catching coil and the current near the catching coil. For large distances from the catching coil the magnetic force is much less the spring force, as shown in Figure 2. Therefore the system response is dominated by the spring force and the controller has no authority over the armature motion at large distances. Near the catching coil the magnetic force does have influence over the armature motion, but manipulation of equation (1) reveals that the input voltage has little influence on the current dynamics. To demonstrate this consider the current dynamics:

$$\frac{di}{dt} = 1000\left(\frac{(V_c - ri)(k_b+z)}{2k_a} + \frac{iv}{(k_b+z)}\right). \quad (10)$$

Note that $k_b \ll 1$ and z is approaching zero. Then replacing the term $k_b + z$ with the vanishing parameter $\varepsilon = k_b + z$, results in

$$\frac{di}{dt} = 1000\left(\frac{(V_c - ri)\,\varepsilon}{2k_a} + \frac{iv}{\varepsilon}\right). \quad (11)$$

Therefore very near the catching coil, the input, V_c has little effect on the current dynamics. Physically, this is caused by the back EMF and the changing inductance.

To summarize, a well designed feedback controller should:

- Initially raise the current while the input can influence the current dynamics in preparation for when the armature nears the catching coil.
- Apply high voltage at the end of the transition to counteract the changing back EMF and inductance.

In [4, 5] this is accomplished through the use of linear and nonlinear controllers, respectively.

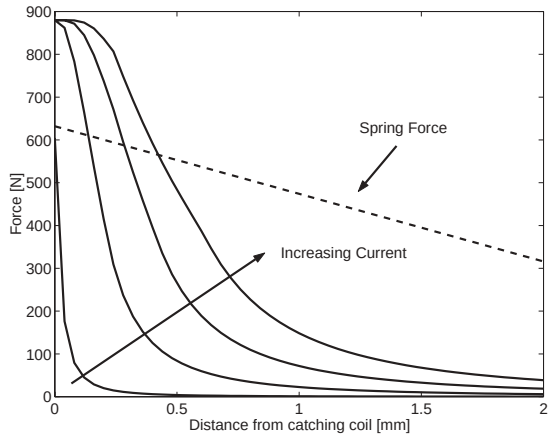

Figure 2: Magnetic force versus position for several values of current

A linear feedback control analysis and design is presented in [4]. A two stage LQR feedback controller is designed and implemented to account for the changing control authority. The first stage of the feedback (which acts for distances greater than 1 mm away from the catching coil) penalizes current much more than position or velocity in the LQR optimization. As the feedback has control authority over current during this time, it is able to raise the current to some nominal catching value, while adjusting slightly for deviations in position and velocity. Near the catching coil (distances less than 1 mm away) the controller switches to the second stage. In the second stage the feedback penalizes position and velocity heavily in the LQR optimization, to ensure that the armature is brought into contact with the catching coil. Current is not penalized in the second stage as it becomes singular perturbed near the catching coil, equation (11), and has been removed via a model reduction [4]. This controller achieves impact velocities with a mean of 0.16 m/s.

The linear control law for V_c exacerbates the loss of control authority by being proportional to the decreasing armature distance (second state). It requires large gains to stabilize the system and this leads to frequent actuator saturation near the catching coil. A more appropriate solution is the nonlinear feedback law:

$$V_c = \frac{K_1}{\gamma + z} v + \frac{K_2}{\beta + z} \tag{12}$$

designed in [5]. Where K_1, K_2, γ, and β are adjustable gains. The input V_c is inversely proportional to the decreasing distance thereby alleviating the effect of the back EMF and changing inductance near the catching coil. Away from the catching coil the input voltage is sufficient to raise the current in preparation for catching the armature. This feedback also achieves impact velocities with a mean of 0.16 m/s.

In [4, 5] an eddy current sensor is used to measure the armature distance, and a nonlinear observer is implemented to estimate the unmeasured states and filter the noisy position measurement. The nonlinear observer is given by

$$\frac{d\widehat{x}}{dt} = f\left(\widehat{x}, V_c\right) + L\left(y - \widehat{y}\right) \tag{13}$$

where $\widehat{x}$ is the estimated state vector $\left[\begin{array}{ccc} \widehat{i} & \widehat{z} & \widehat{v} \end{array}\right]^T$, the gain matrix L is chosen by using a Kalman filter, y is the measured position, $\widehat{y}$ is the estimated position, and the function $f\left(\widehat{x}, V_c\right)$ is the same as equation (7).

To improve the transient response the use of feed-forward compensation from cycle to cycle using an Iterative Learning Controller is investigated in [3]. Manufacturing variations, disturbances, and wear make it impractical to robustly achieve the stringent transient response requirements with a feedback loop. As the valve motion is repeated many times per second (i.e. 50 times per second, when the engine is running at 3000 rpm), the feedback in [3] is augmented with an ILC. Using the tracking error from the previous valve transitions, the ILC modifies the input of the next transition to decrease the tracking error. Here, implementation issues associated with applying an ILC to the EMV actuator are investigated, and experimental results are presented.

3 Iterative Learning Controller

3.1 ILC by singular value decomposition

Iterative Learning Control exploits the repetitive nature of a system to improve the tracking of a desired trajectory, y_d, from cycle to cycle.

Let the input and the output of the system be given by $u\left[k\right]$ and $y\left[k\right]$. Where $u\left[k\right]$ and $y\left[k\right]$ are the k^{th} input and output trajectory respectively. Let us define the mapping $\Gamma : R^n \rightarrow R^n$ as $y\left[k\right] = \Gamma\left(u\left[k\right]\right)$. Using the impulse response sequence $\{h\left[n\right]\}$ of the closed loop discrete linearized system to form the convolution-matrix

$$P = \left[\begin{array}{ccccc} h\left[1\right] & 0 & 0 & \cdots & 0 \\ h\left[2\right] & h\left[1\right] & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ h\left[N-1\right] & h\left[N-2\right] & \cdots & h\left[1\right] & 0 \\ h\left[N\right] & h\left[N-1\right] & \cdots & h\left[2\right] & h\left[1\right] \end{array}\right] \tag{14}$$

that provides a linear approximation of the system near the equilibrium $\Gamma\left(u\left[k\right]\right) \cong Pu\left[k\right]$.

The purpose of the ILC is to find the input, u^*, such that $y_d \cong Pu^*$. A linear ILC is given by

$$u\left[k+1\right] = Su\left[k\right] + E\left(y_d - y\left[k\right]\right) \tag{15}$$

where the matrix S is the weight on the previous input and the matrix E is the weight on the previous error. S and E need to be selected such that

$$u^* = \lim_{k\to\infty} u[k]. \tag{16}$$

As suggested in [1, 3], let S and E be given by

$$S = I, \text{ and } E = \frac{1}{\sigma_o} R L^T \tag{17}$$

where R and L contain the right and left singular vectors of the singular value decomposition, $P = L\Lambda R^T$. Λ is a diagonal matrix whose elements are the singular values of the convolution-matrix, σ_i, arranged in decreasing order, $\sigma_o \geq \sigma_1 \geq \cdots \geq \sigma_i \geq \cdots \sigma_{N-2} \geq \sigma_{N-1}$.

Assuming $y[k] \cong Pu[k]$, equation (15) is rewritten as

$$\begin{aligned} u[k+1] &= Su[k] + E(y_d - Pu[k]) && (18)\\ \Rightarrow R^T u[k+1] &= \left(I - \frac{1}{\sigma_o}\Lambda\right) R^T u[k] + \frac{1}{\sigma_o} L^T y_d && (19) \end{aligned}$$

Applying the linear transformation $v[k] = R^T u[k]$, and $\mu = L^T y_d$ results in

$$v[k+1] = \left(I - \frac{1}{\sigma_o}\Lambda\right) v[k] + \frac{1}{\sigma_o}\mu \tag{20}$$

which can be rewritten into N separate equations

$$v_i[k+1] = \left(1 - \frac{\sigma_i}{\sigma_o}\right) v_i[k] + \frac{1}{\sigma_o}\mu_i \;\forall\; i \in [0,\, N-1]. \tag{21}$$

Thus our choice of S and E yields a de-coupled learning algorithm. The convergence of the ILC is governed by the eigenvalues of the N scalar equations in (21). Each eigenvalue is given by $\lambda_i = \left(1 - \frac{\sigma_i}{\sigma_o}\right)$. Convergence is guaranteed for each scalar equation if $|\lambda_i| < 1$. As $\sigma_o \geq \sigma_i > 0$, $|\lambda_i| < 1$ is ensured. Solving equation (21) recursively yields

$$v_i[k] = \left(1 - \frac{\sigma_i}{\sigma_o}\right)^k v_i[0] + \frac{1 - \left(1 - \frac{\sigma_i}{\sigma_o}\right)^k}{\sigma_i}\mu_i \tag{22}$$

which converges to

$$\lim_{k\to\infty} v_i[k] = v_i^* = \frac{1}{\sigma_i}\mu_i. \tag{23}$$

Since the convergence rate is determined by $1 - \frac{\sigma_i}{\sigma_o}$ and v_i^* is proportional to σ_i^{-1}, output components that require small inputs are learned faster than output components requiring large inputs, which is useful in helping to avoid actuator saturation.

Transforming equation (23) back into y_d and u we find

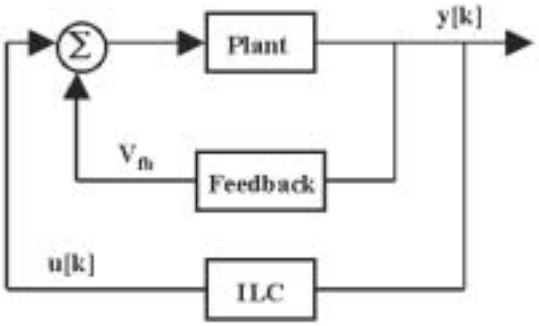

Figure 3: The iterative learning controller as applied to the EMV actuator

$$R^T u^* = \Lambda^{-1} L^T y_d$$

which after noting that $R^T = R^{-1}$ and $L^T = L^{-1}$ yields

$$u^* = R\Lambda^{-1} L^T y_d. \tag{24}$$

After convergence the plant output is equal to the desired trajectory:

$$\begin{aligned} y &= Pu^* \\ \Rightarrow y &= L\Lambda R^T u^* \\ \Rightarrow y &= L\Lambda R^T R\Lambda^{-1} L^T y_d \\ \Rightarrow y &= y_d. \end{aligned}$$

So in essence the iterative learning controller is finding the plant inverse from cycle to cycle to achieve tracking of the desired trajectory.

3.2 ILC as applied to the EMV actuator

In this paper, the ILC is applied to the EMV Actuator as shown in Figure 3. Where the output, y, is the armature distance from the catching coil, and the input, u, is a pure feed-forward voltage.

To properly derive the linear model from the nonlinear system (7), let

$$V_c = V_{fb} + V_{ff} \tag{25}$$

where V_{fb} is the feedback voltage from the nonlinear controller (12), and V_{ff} is the feed-forward voltage, u, from the ILC as explained in section 3.1. Substituting into equation (1), the closed loop current dynamics are

$$\frac{di_{cl}}{dt} = \frac{V_{fb} + V_{ff} - ri + \chi_1 v}{\chi_2} \quad (26)$$

$$\frac{di_{cl}}{dt} = \frac{\frac{K_1}{\gamma+z}v + \frac{K_2}{\beta+z} + V_{ff} - ri + \chi_1 v}{\chi_2} \quad (27)$$

The A_{cl} and B_{cl} matrices of the closed loop linear system

$$\frac{dx}{dt} = A_{cl}x + B_{cl}V_{ff} \quad (28)$$

$$y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} x \quad (29)$$

are given by

$$A_{cl} = \begin{bmatrix} \frac{\partial}{\partial i}\left(\frac{di_{cl}}{dt}\right)\Big|_{x=x_{eq}} & \frac{\partial}{\partial z}\left(\frac{di_{cl}}{dt}\right)\Big|_{x=x_{eq}} & \frac{\partial}{\partial v}\left(\frac{di_{cl}}{dt}\right)\Big|_{x=x_{eq}} \\ 0 & 0 & \frac{\partial}{\partial v}\left(\frac{dz}{dt}\right)\Big|_{x=x_{eq}} \\ \frac{\partial}{\partial i}\left(\frac{dv}{dt}\right)\Big|_{x=x_{eq}} & \frac{\partial}{\partial z}\left(\frac{dv}{dt}\right)\Big|_{x=x_{eq}} & \frac{\partial}{\partial v}\left(\frac{dv}{dt}\right)\Big|_{x=x_{eq}} \end{bmatrix} \quad (30)$$

and

$$B_{cl} = \begin{bmatrix} \frac{\partial}{\partial V_{ff}}\left(\frac{di}{dt}\right)\Big|_{x=x_{eq}} \\ 0 \\ 0 \end{bmatrix} \quad (31)$$

where $x_{eq} = \begin{bmatrix} i_e & 0 & 0 \end{bmatrix}$ is the equilibrium state, and i_e is the value of current such that $F_{mag}(i_e, 0) = 4k_s$.

Therefore the input voltage depends on the feed-forward and feedback as shown in the block diagram in Figure 3. The convolution matrix, P for the ILC, is formed by the impulse response of the discrete version (with a sampling rate 20kHz) of equations (28), (29), (30), and (31).

4 Implementation issues

This section presents modifications to the ILC to compensate for nonlinearities, noise, and computational load.

4.1 Computational limitations

The iterative learning controller needs to run real time within an event (consecutive valve opening or closing) so the computational load is constrained by our processing capabilities. In this section we present how the computational load can be reduced by careful consideration of the performance objective and actuator authority.

The main purpose of the ILC is to achieve soft landing of the armature against the catching coil. As such, it only needs to achieve tracking of the desired position at the very end of the armature transition. Let us partition the output space, $y[k] = Pu[k]$, into

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} u \quad (32)$$

where y_1 and y_2 are the initial and final portions of the position trajectory output respectively. The distinction between the two portions is defined by the distance that corresponds to magnetic force higher than the spring force. The magnetic field becomes strong even for the equilibrium (low) current values for distances less than 0.3 mm. The output space partition is then defined as:

- y_1 be the portion of the position trajectory output for distances greater than 0.3 mm away from the catching coil.
- y_2 be the portion of the position trajectory output for distances less than 0.3 mm away from the catching coil.

Therefore to achieve tracking of only y_2 the mapping employed by the ILC is reduced to $y_2 = P_2 u$. The computational load is further decreased by taking into account the control authority issues raised in Section 2.2.

The input space $y_2 = P_2 u$ can now be separated as

$$y_2 = \begin{bmatrix} P_{21} & P_{22} & P_{23} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (33)$$

where u_1 contains the input signals applied while the armature is greater than 3.5 mm away from the catching coil, u_2 contains the input signals applied while the armature is between 3.5 mm and 0.3 mm away from the catching coil, and u_3 contains the input signals applied while the armature is less than 0.3 mm away from the catching coil. Similarly, P_{21} maps the effect of inputs applied while the armature distance is greater than 3.5 mm away from the catching coil to y_2, P_{22} maps the effect of inputs applied while the armature distance is between 3.5 mm and 0.3 mm away from the catching coil to y_2, and P_{23} maps the effect of inputs applied while the armature distance is less than 0.3 mm away from the catching coil.

The significance of these regions in the input space can be described as follows. First, since the input has little effect on the current dynamics for small distances, equation (11), there is no need for the iterative learning controller to modify the input voltage at small distances from the catching coil, $y < 0.3$ mm. Second, as the magnetic force is much less than spring force at large distances, there is also no reason why the iterative learning controller should modify the input voltage when the armature is very far away from the catching coil, $y > 3.5$ mm.

From the union of these sets, the iterative learning controller only adjusts the feed-forward voltage between 3.5 mm and

0.3 mm away from the catching coil. At a sampling rate of 20 kHz this corresponds to approximately 30 samples, which is the maximum size that our processing board can implement in real time. To keep the matrices square, the size of y_2 is set to be 30 samples.

The final mapping employed by the ILC is thus $y_2 = P_{22}u_2$. The weighting matrices S and E of equation (15) are determined by the singular value decomposition of P_{22}.

Due to sampling and measurement noise, it is not possible to have u_2 start and end exactly at 3.5 mm and 0.3 mm away from the catching coil. Nor is it possible to have y_2 start exactly at 0.3 mm away from the catching coil. To clarify, thus, the computations we introduce additional notation based on the discrete sampled events. Let us index the k^{th} iteration of the input voltage, u_2, which has length 30 as

$$u_2[k][j_u] \text{ for } j_u \in (0, 30), \tag{34}$$

and define $u_2[k][0] = 0$. For distances greater than 3.5 mm away from the catching coil j_u is held at zero. After the controller detects a position less than 3.5 mm away from the catching coil, j_u increases incrementally with the sampling rate.

Similarly, let us index the k^{th} iteration of the output, y_2, which has length 30 as

$$y_2[k][j_y] \text{ for } j_y \in (1, 30) \tag{35}$$

where $y_2[k][j_y]$ stores the measured output at the current time step. For distances greater than 0.3 mm away from the catching coil j_y is held at 1. Thus, the controller keeps rewriting the value of $y_2[k][1]$ at each time step. After the controller detects a position less than 0.3 mm away from the catching coil, j_y increases incrementally with the sampling rate. Therefore the control captures approximately the last 0.3 mm of the transition.

There exists some variation in where exactly u_2 and y_2 begin and end due to the sampling rate and noise. However, j_u is always allowed to reach 30, even if this means that the ILC will modify the voltage for distances less than 0.3 mm away from the catching coil.

4.2 Selection of desired trajectory

By truncating the portion of the trajectory we wish to track, we have inadvertently introduced another complication. The desired trajectory, y_d, must be a "smooth extension" of y_1 for it to be feasibly tracked by y_2. Specifically, the position and velocity at the start of y_d must match the position and velocity at the end of y_1.

To satisfy this constraint let

$$y_d[k][j_y] = y_2[k][1]\, e^{\frac{v[k][1]}{y_2[k][1]}(j_y - 1)TS} \tag{36}$$

where $v[k][1]$ is the velocity of the armature when the sample $y_2[k][1]$ is recorded and TS is the sampling time.

A small negative value is chosen for $v[k][1]$that will ensure that $y_d$is decaying exponentially and thus as the tracking error decreases so should the impact velocity. The fixed $v[k][1]$ value does not guarantee a smooth trajectory but it is a simple solution for the case of zero air flow disturbances. Ideally, we could use a velocity sensor for an accurate value of the initial velocity or we could use the estimated velocity from the observer in (13).

4.3 Noise compensation

To prevent the ILC from attempting to compensate for sensor noise, the measured trajectory needs to be filtered. While the observer does filter the output, it does not completely eliminate the noise. Again, the repetitive nature of the system is exploited to improve the filtering.

Instead of updating every cycle, the ILC is modified to update every third cycle. The tracking error is thus determined by the difference between the desired trajectory and the average of the previous three measured trajectories. Equation (15) is now given by

$$u[n+1] = Su[n] + Ey_{error} \text{ for } n = 3k \tag{37}$$

$$u[n+1] = u[n] \text{ for } n \neq 3k \tag{38}$$

where

$$y_{error} = y_d - \frac{1}{3}\sum_{j=0}^{2} y[n-j]. \tag{39}$$

5 Experimental results

The following experiments are conducted to gauge the effectiveness of the ILC. If well tuned, the nonlinear feedback given by equation (12) achieves impact velocities with a mean of 0.16 m/s. The output of the well tuned nonlinear feedback matches the desired trajectory closely, as shown in Figure 4. The free response is also included in Figure 4 for comparison.

In each experiment the feedback is perturbed slightly, and the ILC is used to track the desired trajectory. For consistency with our previous work [4, 5, 7] the position plotted in the following figures is the deviation from the middle position (4 mm distance from the coil).

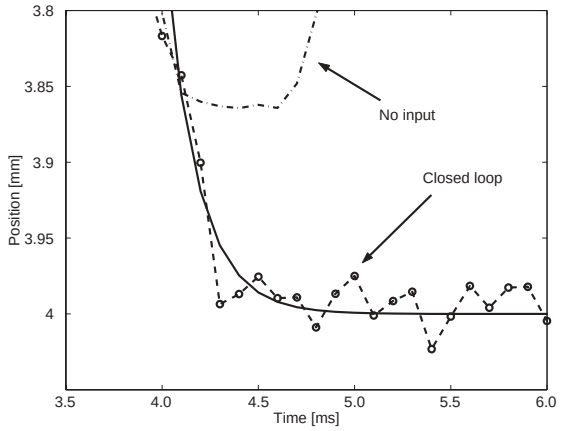

Figure 4: Comparison of the desired trajectory and the output achieved by a well tuned feedback

5.1 Matched perturbation

For the first experiment the feedback is modified between 3.5 mm and 0.3 mm such that the input is held constant during this period. As the ILC operates over the same range, this presents an easy scenario for it to handle.

The average of the 1^{st} set of three trajectories, the average of the 10^{th} set of three trajectories, and the desired trajectory are plotted together for comparison in Figure 5. Initially the tracking error is quite large and the armature bounces against the catching coil. Ten iterations later, the output is much smoother and the error has decreased significantly.

The change in the input voltage between the 1^{st} and 10^{th} iteration is shown in Figure 6. The dotted line represents the voltage that would have been given by the non-modified feedback.

5.2 Detuned feedback

In this section the feedback is perturbed throughout the transition, while the ILC remains limited to acting between 3.5 mm and 0.3 mm away from catching coil. The gains of the nominal feedback are adjusted such that the new feedback, V_{fb}^{+}, is always greater than the nominal feedback. As seen in Figures 7 and 8 the ILC is able to adjust the voltage in seven iterations to achieve better tracking of the desired trajectory. endfigure

If the gains of the nominal feedback are adjusted such that the new feedback, V_{fb}^{-}, is always less than the nominal feedback, a problem occurs. If V_{fb}^{-} is significantly less than the nominal feedback the armature is no longer brought into contact with the catching coil, and returns to the mid position. At which point the system needs to be re-initialized to get the armature moving again. If V_{fb}^{-} is slightly less than the nominal feedback the error is initially very small. As explored in section 5.3, this causes the performance of the ILC to deteriorate.

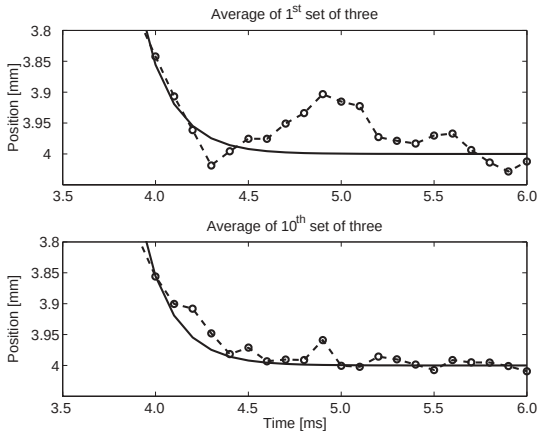

Figure 5: Comparison of the 1^{st} and 10^{th} iteration for the matched perturbation case.

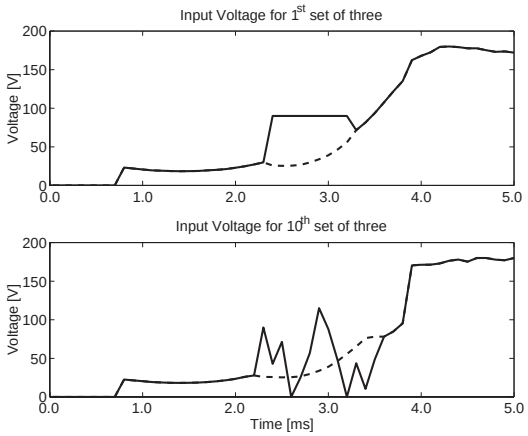

Figure 6: Comparison of the 1^{st} and 10^{th} iteration input voltage for the matched perturbation case.

5.3 Performance deterioration

For each of the previous experiments, once the error between the output and desired trajectory is sufficiently small, the performance begins to deteriorate. This is seen in Figures 9, 10, and 11 which show the output, input voltage, and 2-Norm of the error respectively.

The dynamics of the error exhibit non-monotonic behavior that causes initial convergence, followed with divergence. As described in [2] this is caused when the error at different points along the trajectory converge at different rates. In our case, a large divergence may prohibit landing of the armature against the catching coil, which requires a re-initialization procedure. The convergence/divergence of individual points along the trajectory are plotted in Figures 12 and 13. From these plots it appears that before the error diverges at the 10^{th} iteration:

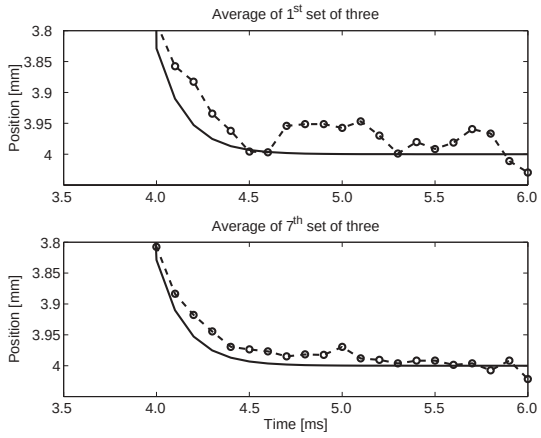

Figure 7: Comparison of the 1^{st} and 7^{th} iteration for V_{fb}^{+}

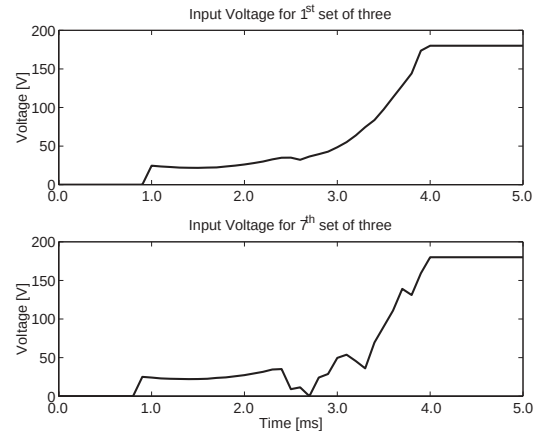

Figure 8: Comparison of the 1^{st} and 7^{th} iteration input voltage for V_{fb}^{+}

1. Points at the beginning of the trajectory are not necessarily convergent.
2. Points in the middle of the trajectory converge slowly.
3. Points at the end of the trajectory converge quickly.

Due to the significant nonlinearities in the system near the catching coil the convolution matrix is probably not accurate enough to ensure monotonic behavior at all points in the trajectory.

5.4 Impact velocity

While tracking improves in sections 5.1 and 5.2, it is important that the impact velocity decreases as well. As plotted in Figure 14, the impact velocity decreases, but the ILC does not achieve impact velocities as small as those achieved in [4, 5]. Perhaps if the ILC can be improved to avoid the performance deterioration seen in section 5.3, the impact velocity would decrease further.

6 Conclusion

While the Iterative Learning Controller is able to improve the tracking performance of the ElectroMechanical Valve Actuator, its effectiveness is limited by several factors:

1. Control Authority
2. Processing Capability
3. Measurement Noise

The methodology presented here attempts to account for these problems, but does not overcome each one completely. However, experimental results do show a decrease in the tracking error. Future work will investigate the development of a more accurate convolution matrix to avoid the convergence followed by divergence phenomenon.

Acknowledgements

Support is provided by the National Science Foundation under contract NSF-ECS-0049025 and Ford Motor Company through a 2001 University Research Project.

References

[1] Doh, T.Y., Moon J.H., Jin, K.B., and Chung, M.J., "An iterative learning control for uncertain systems using structured singular value". In Proceedings of the 2nd Asian Control Conference, Seoul Korea, July 1997. 61

[2] Huang, Y.C., Longman, R.W., "The source of the often observed property of Initial convergence followed by divergence in learning and repetitive control". Advances in astronautical sciences, Vol: 90 pages: 555-572.

[3] Hoffmann W., Stefanopoulou A., "Iterative Learning Control of Electromechanical Camless Valve Actuator," Proceedings American Control Conference, pp.2860-2866, June 2001.

[4] Peterson, K., Stefanopoulou A., Haghgooie M., Megli, T. "Output Observer Based Feedback for Soft Landing of Electromechanical Camless Valvetrain Actuator", to be presented at the American Control Conference 2002.

[5] Peterson, K., Stefanopoulou A., Wang Y., Haghgooie M., Megli, T. "Nonlinear self tuning control for soft landing of an Electromechanical Valve Actuator", To be presented at the IFAC on mechatronics 2002.

[6] Tai, C., Stubbs, A., Tsao, T.C., "Modeling and Controller Design of Electromagnetic Engine Valve". Proceedings of American Control Conference, pp. 2890-2895, June 2001

[7] Wang Y., Stefanopoulou A., Peterson K., Megli T., Haghgooie M., Modeling and Control of Electromechanical Valve Actuator", SAE 2002-01-1106.

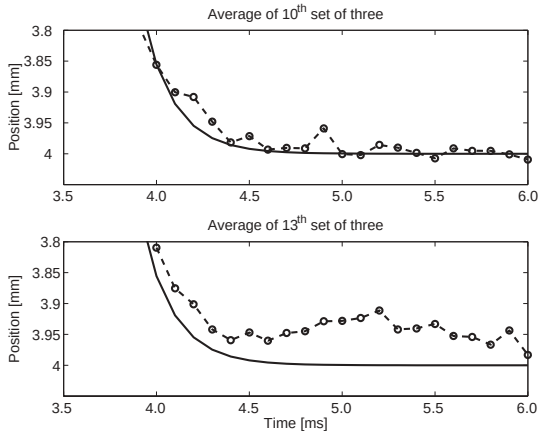

Figure 9: Comparison of the 10^{th} and 13^{th} iteration for the matched perturbation case

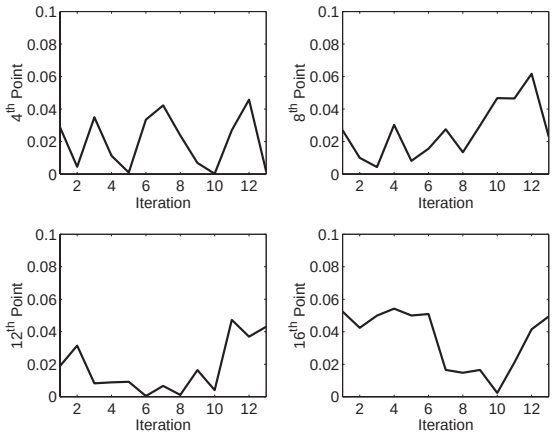

Figure 12: Convergence/Divergence of the 4^{th}, 8^{th}, 12^{th}, and 16^{th} points in the trajectory

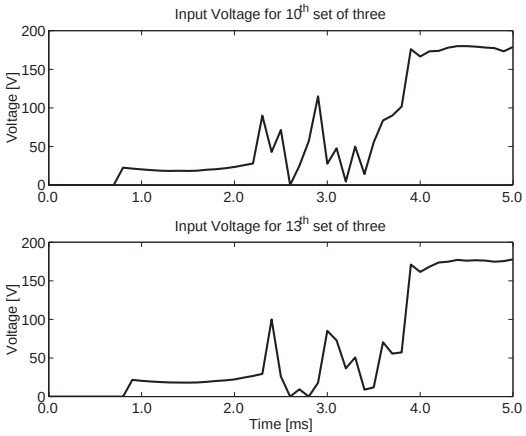

Figure 10: Comparison of the 10^{th} and 13^{th} iteration input voltage for the matched perturbation case

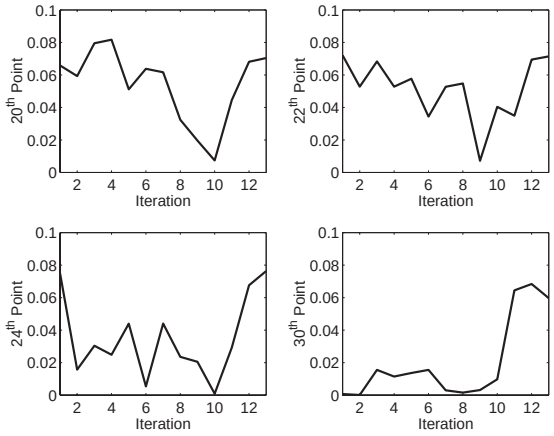

Figure 13: Convergence/Divergence of the 20^{th}, 22^{th}, 24^{th}, and 30^{th} points in the trajectory

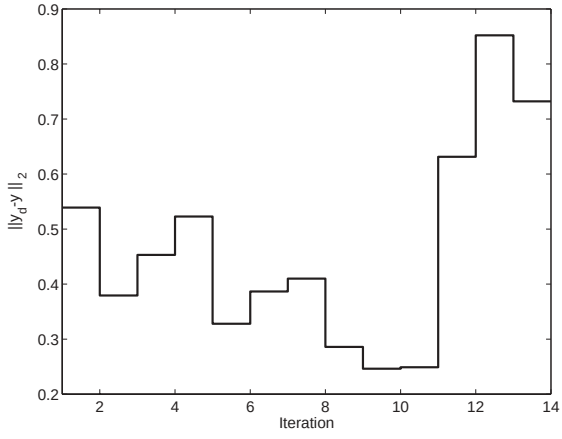

Figure 11: The 2-Norm of the error for the matched perturbation case

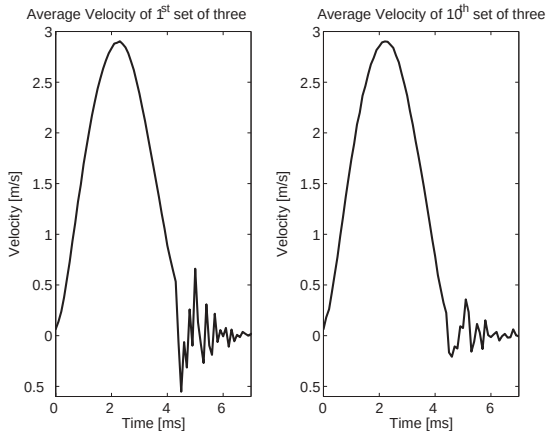

Figure 14: Velocity of the 1^{st} and 10^{th} Iterations