Cycle Adaptive Feedforward Approach Control of an Electromagnetic Valve Actuator

Jimmy Tsai, Charles Robert Koch, and Mehrdad Saif

***Abstract*— An electromagnetic valvetrain on an internal combustion engine can improve the engine thermal efficiency but requires control to achieve soft landing and to avoid excessive wear and noise. Since the valves open and close repetitively, cycle adaptive control can be utilized. A cyclic adaptive feedforward approach controller for automotive electromagnetic valve is presented. This method uses a Nelder-Mead direct search algorithm with the goal of setting constant initial conditions for the landing control. Simulation and testbench results are presented and they show that the approach control works well for disturbances that are slow compared to the valve travel time.**

I. INTRODUCTION

The coupling between crankshaft and engine valve operation presents an area of potential improvement for the internal combustion engine. If this coupling is removed, the engine valve operation can be optimized at different operating conditions using variable valve timing (VVT) [1] [2]. While many variable valve timing systems are available, the promise of improved engine performance, emission and fuel efficiency provides a strong motivation to develop camless valvetrains [3]. Candidate actuators considered for camshaft replacement include hydraulic [4], rotary motor [5], piezoelectric [6], and electromagnetic solenoid actuator [7].

Of all these actuators, the solenoid/electromagnetic valve actuator (EMV) excels in its cost, efficiency, and ruggedness [8]. The EMV actuator considered here consists of two springs, two solenoids, and one shaft that connects a metal armature to the valve (See Figure 1). The springs are preloaded in compression evenly to store energy. When there is no current on either coil of the actuator, the armature rest position is in the middle due to the balanced spring forces. To open or close a valve, each of the two solenoids acts as electromagnet to attract the armature to the respective end of the actuator. The energy needed for travel is mostly recovered because it is stored in the springs. The solenoids are needed only for the additional pull to land and hold the armature.

Without control, the EMV actuator control tends to suffer from excessive valve seating and the resultant premature-wear and acoustic emission [10]. The control problem arises from the low force and low control authority at large airgaps and high inductance and reduced bandwidth at at small airgaps. Specifically in the EMV actuator, the magnetic force drops off inversely proportional with the square of gap distance whereas the system inductance increases with gap distance [11]. Additionally, the exhaust valve has to overcome large pressure disturbances related to valve timing and engine load [12].

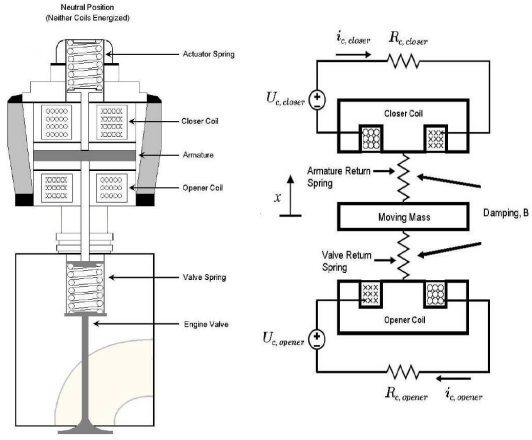

Fig. 1. The EMV actuator : actual (left) and schematic (right) [9]

For effective control, the controller can be divided into two parts [13]. The landing controller is active close to the catching coil where there is enough control authority for it to track a smooth landing trajectory, while the approach controller operates over the remaining trajectory to keep the initial conditions constant for the landing control. The position-velocity plot in Figure 2 shows where the two controllers act. The important position in the plot is the end of approach control, which is also the beginning of the landing control: $x^i_{land} = x^f_{appr} = 2.55$mm.

The superscripts and subscripts i, f, $land$, and $appr$ stands for "initial", "final", "landing control" and "approach control" respectively. Like the dashed line connecting the disturbed to the ideal valve lift in Figure 2, the approach controller compensates any disturbances so that at x^f_{appr}:

$$v^f_{appr} = v^i_{land} = v_d \quad \text{and} \quad i^f_{appr} = i^i_{land} = i_d$$

Published works on EMV actuator control, both on the approach controller and the landing controller, are extensive. A simplified relationship between current measurement with the armature velocity over position ratio is presented in [14] to facilitate "sensorless control", which uses no position/velocity measurement. An LQ optimal controller based on a linearized system model is used in [15]. Later, the sensorless control is improved by adding in take-off and

Jimmy Tsai is with School of Engineering Science, Simon Fraser University, Vancouver, B.C. V5A 1S6, Canada jtsai@sfu.ca

Charles Robert Koch is with Department of Mechanical Engineering, University of Alberta, Edmonton, Alberta T6G 2G8, Canada bob.koch@ualberta.ca

Mehrdad Saif is with School of Engineering Science, Simon Fraser University, Vancouver, B.C. V5A 1S6, Canada saif@ensc.sfu.ca

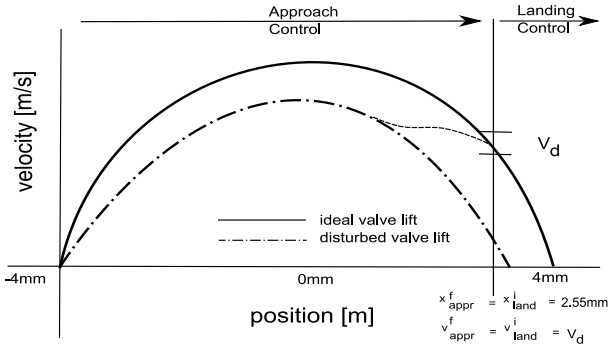

Fig. 2. Control regions for approach and landing control: setting consistent initial condition for the EMV landing control

approach control [16]. Sliding mode controllers for the EMV actuator are presented in [17] and [18]. Repetitive learning control [19], iterative learning control [20], and extremum seeking control [11] [21] all attempt to solve the valve soft seating problem by incorporating cycle-to-cycle information to improve performance. A wide variety of supply voltage, impact velocity, transition time, convergence time, and sensor requirements are given in the literature, so care must be taken when comparing results. The approach controller presented here uses $\pm 42v$ volt supply voltage, a mass-spring with 150 Hz natural frequency (up to 5000∼6000 rpm engine operation), and an LVDT position sensor for feedback.

This paper focuses on the cycle adaptive feedforward approach control of the EMV actuator. The use of feedforward control compensates for the small control force, while cyclical adaptation takes advantage of the repetitive nature of the engine valve operation. For this type of control, it is assumed that the disturbance is much slower than the valve travel time: i.e. the residuals from previous iterations can be used for current iteration. The learning control and the extremum control publications listed above also make the same assumption.

To our knowledge, the feedforward controller presented is novel and has not been presented in the literature. Unlike the learning control strategies, our controller requires no trajectory tracking. Moreover, unlike extremum control which tunes only one variable, the Nelder Mead controller methods can tune multiple coefficients for greater optimization.

II. Model for the EMV actuator

The valve lump-parameter model [13] comes from parameterizing the FEA model in [22].

$$\frac{dx}{dt} = v \tag{1}$$

$$\frac{dv}{dt} = \frac{1}{m}\left[\frac{\lambda_s f'(x)}{f^2(x)}\left[1-[1+i_c f(x)]\,e^{-i_c f(x)}\right]\right] - K_s x/m - Bv/m \tag{2}$$

$$\frac{di_c}{dt} = \frac{e^{i_c f(x)}}{\lambda_s f(x)}\left[U_c - R_c[i_c+i_e]\right] - \frac{f'(x)}{f(x)} i_c v \tag{3}$$

$$\frac{di_e}{dt} = \frac{1}{L_e(x,i_c)}\left[U_c - R_c(i_c+i_e) - R_e(x,i_c)i_e\right] \tag{4}$$

TABLE I

EMV actuator model variable & parameter

symbol	name	symbol	name
U_c	voltage	x	position
i_c	coil current	i_e	eddy current
R_c	coil resistance	R_e	eddy current resistance
v	velocity	K_s	spring constant
m	mass	$f(x)$	$2c_1/(c_2-x)+c_3$
B	damping constant	L_e	eddy current inductance
c_1, c_2, c_3	curve-fit constants		

This model accounts for flux saturation through curve-fitting function $f(x)$ into magnetic force expression. The system eddy current dynamics are accounted by additional RL circuit with constant R_e and L_e. The model parameters are listed in table I.

III. Nelder Mead Simplex Algorithm

The electromagnetic actuator motion is highly repetitive because normal engine operation requires valves to open or close at between approximately 10 to 100Hz. As many of the disturbances and parameters change slowly with respect to the opening and closing time, the information from the previous valve events can be utilized as basis for feedforward control in the current valve event. One way to take advantage of this property is to treat each valve open/close event as a functional evaluation and solve the nonlinear programming problem of the form:

minimize	$F(c)$	with respect to parameter c
subject to	$g_i(c)=0$	i = # of equality constraints
subject to	$h_j(c)>=0$	j = # of inequality constraints

In our cycle adaptive controller, we parameterized input coil current profile $i_c(t)$ by a position-based uniform b-spline $i_c(x(t),c)$, whose coefficient vector c are then tuned by the Nelder Mead algorithm. The vector c is further constrained by c_{ub}, which interpolates the current profile produced under constant maximum voltage input, and c_{lb}, which interpolates the lowest current profile that still enables landing.

$$h_1(c) = c_{ub} - c > 0 \text{ and } h_2(c) = c - c_{lb} > 0$$

In summary, the EMV optimization has no equality constraint $g_i(c)$ and only two input inequality constraints $h_j(c)$ in c_{lb} and c_{ub}, which defines the highest and lowest current profiles that the search must stay within . Lastly, our cost function is defined by a quadratic function of terminal errors at x^f_{appr} (see Figure 2).

$$F(c) = \left(\alpha(v^f_{appr}-v_d)^2 + \beta(i^f_{appr}-i_d)^2\right) \tag{5}$$

α and β are scalar weighting factors.

A. Properties of Nelder Mead algorithm

Nelder Mead simplex method belongs to a branch of nonlinear programming algorithm called direct search, which does not need any derivative information such as gradient or hessian. While other derivative-free methods are available (e.g. pattern search, Rosenbrock's method, Powell's method,

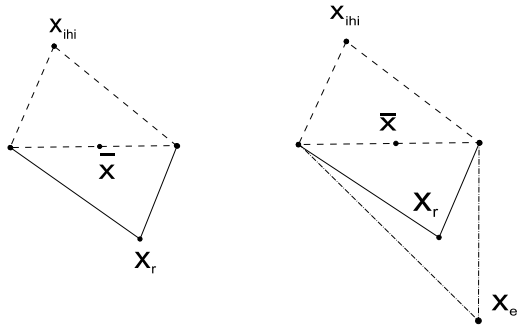

Fig. 3. Nelder Mead algorithm steps: reflection (left) and expansion (right)

etc.), the Nelder Mead is used because of its intuitiveness and computational efficiency [23]. For a detailed discussion on direct search methods, see the survey paper [24].

Instead of taking numerical gradient, the algorithm maintains a "non-degenerate simplex". The definition of simplex is a set of $n+1$ points in n dimensions. i.e., if the input current profile is interpolated by n coefficients, then the simplex should have $n+1$ different current profiles. A simplex is non-degenerate if the vectors connecting any single vertex to the remaining vertices spans the entire space. Non-degeneracy is important because Nelder Mead uses the linear combination of the connection vectors between vertices to search for lower-cost current profiles. During initialization, A simplex is created by perturbing every element of a vertex x_1 by ϵ.

$$\begin{aligned} x_1 &= \left[\begin{array}{ccccc} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \end{array}\right] \\ x_{i+1} &= \left[\begin{array}{ccccc} x_{11} & \dots & x_{1i}+\epsilon & \dots & x_{1n} \end{array}\right] \end{aligned}$$

$i = 1 \dots n$ for all $n+1$ vertices in the simplex.

B. Determining the next trial point

To find the next trial point, we reorder the simplex to separate out the "worst" vertex and move it in the direction of the remaining vertices (represented by their average). This new point is called the reflection point. If the function evaluation at the reflection point is favorable, then we can move the point further toward the same direction for potential improvement. Otherwise, depending on the trial result, the next trial point can be moved back toward the "average" point or even pass the "average" point toward the "worst" point. When the "worst" vertex is replaced by a better vertex, an iteration ends. The goal is to eventually replace the "best" point in the simplex. If nothing better than the "worst" vertex can be found, the vertex itself can be shrunk toward its "best" vertex.

Before describing Nelder Mead in detail, we introduce the tuning parameters ρ, χ, γ, and σ, which are reflection, expansion, contraction, and shrinkage coefficients respectively. They determine the aggressiveness of the algorithm and are constrained by the following rules:

$$0 < \rho < \chi,\ \chi > 1,\ 0 < \gamma < 1,\ \text{and } 0 < \sigma < 1$$

1) **Order**
First $n+1$ vertices are functionally evaluated to separate out the best, the worst, and the second worst vertices.

$$f(x_1) \le f(x_2) \le f(x_3) \dots \le f(x_{n+1})$$

$x_{ilo} = x_1$	best
$x_{inhi} = x_n$	second worst
$x_{ihi} = x_{n+1}$	worst
$\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}$	average (exclude worst)

2) **Reflect** (Figure 3, left)
Then we reflect the worst point toward the average of remaining vertices.

$$x_r = \bar{x} + \rho(\bar{x} - x_{ihi}) = (1+\rho)\bar{x} - \rho x_{ihi} \tag{6}$$

if $f(x_r) < f(x_{ilo})$ then go to expansion step
else if $f(x_r) < f(x_{inhi})$ then replace x_{ihi} with x_r, terminate the iteration.
else if $f(x_r) > f(x_{ihi})$ then go to the contraction step.

3) **Expand** (Figure 3, right)
Compute the expansion point x_e.

$$x_e = \bar{x} + \chi(x_r - \bar{x}) = (1+\rho\chi)\bar{x} - \rho\chi x_{ihi} \tag{7}$$

if $f(x_e) < f(x_r)$ then replace the x_{ihi} with x_e. Otherwise, replace x_{ihi} with x_r and terminate the iteration.

4) **Contract**
if $f(x_{inhi}) \le f(x_r) < f(x_{ihi})$ then go to contract outside step. Otherwise go to contraction side step.

a) *contract inside* (Figure 4, middle)

$$x_{cc} = \bar{x} - \gamma(\bar{x} - x_{ihi}) = (1-\gamma)\bar{x} + \gamma x_{ihi} \tag{8}$$

if $f(x_{cc}) < f(x_{ihi})$, then replace x_{ihi} with x_{cc} in the simplex and terminate the current iteration, else go to the shrink step.

b) *contract outside* (Figure 4, left)

$$x_c = \bar{x} + \gamma(x_r - \bar{x}) = (1+\rho\gamma)\bar{x} - \rho\gamma x_{ihi} \tag{9}$$

if $f(x_c) < f(x_{ihi})$, then replace x_{ihi} with x_c in the simplex and terminate the current iteration, else go to the shrink step.

5) **Shrink** (Figure 4, right)
Evaluate cost function f at the v_i points

$$v_i = x_{ilo} + \sigma(x_i - x_{ilo}) \tag{10}$$

the unordered vertices of the simplex in next iteration will be consisted of $x_{ilo}, v_2, v_3, \dots, v_{n+1}$. Terminate the iteration.

Convergence occurs when either the cost function is reduced sufficiently or when the spacing within the simplex shrinks too much. If the algorithm is stuck at a local minimum, we can reestablish the simplex to reset the search process. Finally, we converted the Nelder Mead algorithm to a finite state machine similar to [25] for real time implementation.

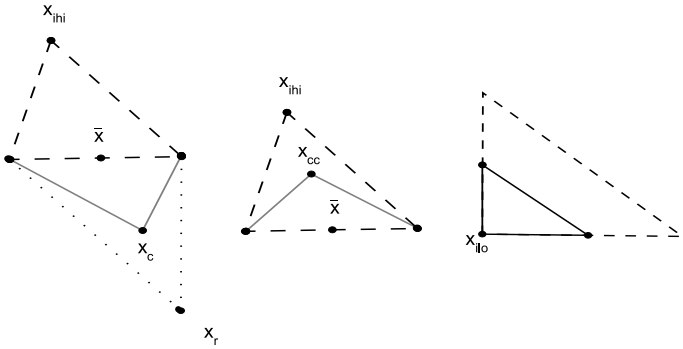

Fig. 4. Nelder Mead algorithm steps: contraction outside (left), contraction inside (middle), and shrinkage (right)

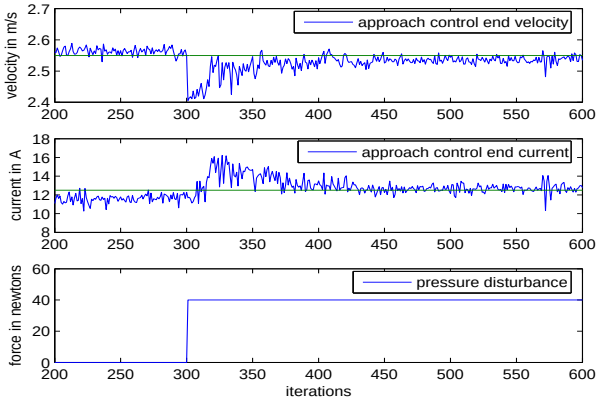

Fig. 5. Step disturbance simulation: approach control end velocity (top), end current (middle), and pressure force (bottom)

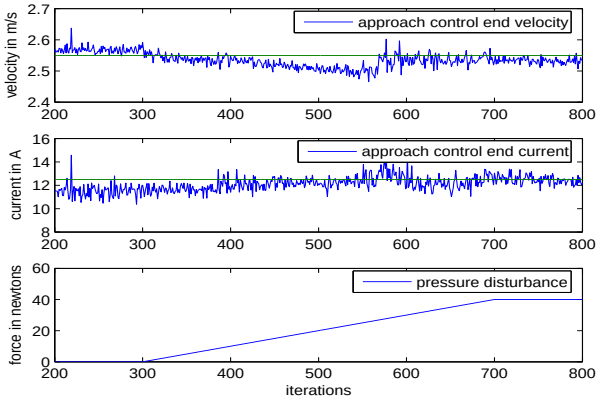

Fig. 6. Ramp disturbance simulation: approach control end velocity (top), end current (middle), and pressure force (bottom)

C. Handling Input Constraints

To handle constraints, a sine coordinate transformation [26] is implemented. Suppose c is the optimization variable constrained by upper-bound c_{ub} and lower-bound c_{lb}. We can solve optimization problem of c by solving the unconstrained problem with z, which is related with c through:

$$c = c_{lb} + (c_{ub} - c_{lb})\frac{(sin(z+1))}{2} \tag{11}$$

By inverting the above equation and checking the necessary condition of arc-sine, an expression for $z(i)$ can be found. Since the output from the sine function only varies between 0 and 1, modifying z will result in changing c within the bounds c_{ub} and c_{lb}. The use of other functions between 0 and 1 are also possible for this type of transformation.

D. Simulation result

To make an assessment of the Nelder Mead algorithm convergence speed, simulation is used to determine the number of iterations required for v^f_{appr} and i^f_{appr} to converge when subjected to a step disturbance. The criteria for convergence is defined to be within $\pm 2\%$ of v_d and $\pm 5\%$ of i_d. In Figure 5, A 40N pressure step increase occurs at step 300 and roughly 100 iterations is needed for both velocity and current to converge. (40N step disturbance is selected to limit the perturbation such that valve-landing during an experiment is ensured.) The response against ramp disturbance of 40N in 400 steps can be seen in Figure 6. Despite simplex vertices becomes obsolete quickly under ramp disturbance, the deviation from velocity setpoint stays within 0.1m/s.

IV. Experimental Setup

The experimental setup is shown on Figure 7. The device on the right is the solenoid valve attached to a workbench and connected to a LVDT position sensor. The device on the left is the H-bridge power electronic that provides three output mode (-42v, 0v, 42v) for PWM output by switching two IGBT power transistors. Integrated in the power electronics are two hall-effect (LEM LA55-P) current sensors. Not shown on the picture are the Sorenson DCS60-18E 1-kW power supply powering the H-bridge and the ± 15V power supply for the sensors. The control software is implemented in C on dSPACE DS1103 hardware at 50Khz. Data monitoring and collection are done on a PC host through dSPACE ControlDesk software program.

V. Experimental Results

A. Unknown disturbance regulation

Even under laboratory condition, the openloop terminal approach velocity v^f_{appr} changes over time due to unknown varying disturbances. The Nelder Mead controller performance over 4000 continuous cycles resulting an average velocity of $v_d = 2.6$m/s and variance of σ_{NM}=2e-5m^2/s^2, which is a reduction by a rough factor of 3.5 compared to openloop variance σ_{OL}=6.9e-5m^2/s^2, as seen in the histogram comparison in Figure 8.

B. Regulate Against Pressure Change

Since the disturbance in section V-A is unknown, it's impossible to quantify how well the the algorithm rejects disturbances. To simulate a known pressure disturbance, the release coil is used to hold back the armature as if

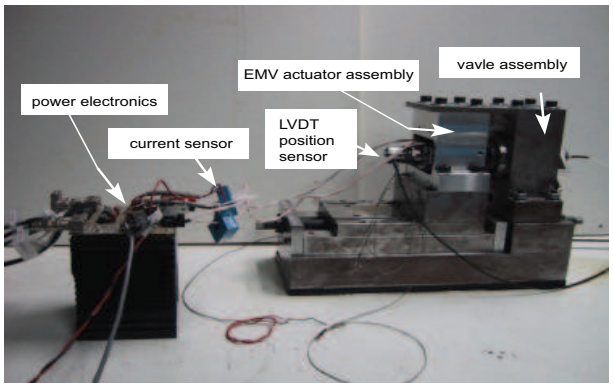

Fig. 7. Electromechanical valve actuator test-bench setup with power electronics and sensors

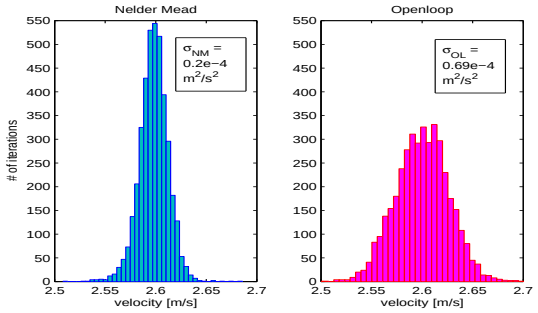

Fig. 8. Histogram of approach control end velocity after 4000 iterations with variances (σ) listed: Nelder Mead (left) and openloop (right)

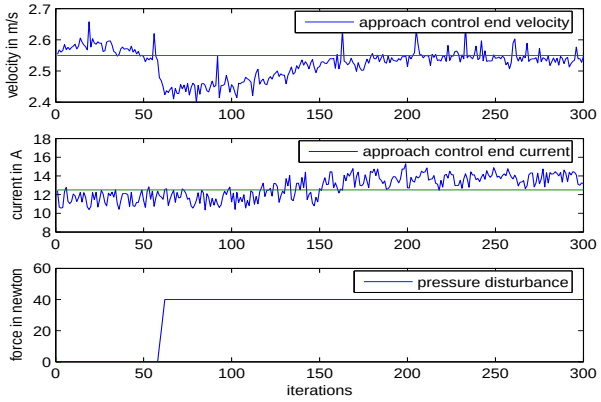

Fig. 9. Step disturbance experiment: approach control end velocity (top), end current (middle), and pressure force (bottom)

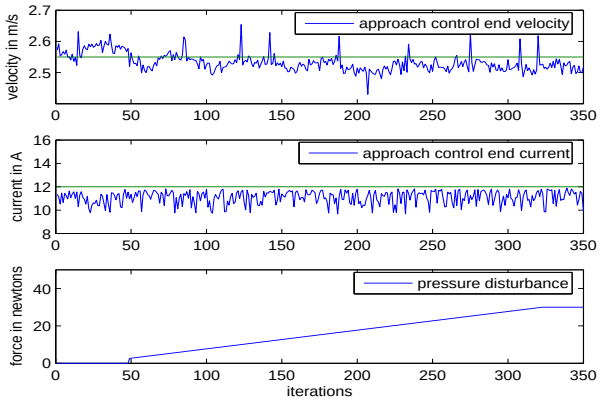

Fig. 10. Ramp disturbance experiment: approach control end velocity (top), end current (middle), and pressure force (bottom)

cylinder pressure is present. The finite element model used for simulating engine back pressure is documented in [22].

1) Step pressure change: For a step pressure increase of 40N, Figure 9 shows that approximately 120 cycles are needed to regulate v^f_{appr} back to within $\pm 2\%$ of the desired velocity v_d. The cycle by cycle values and the disturbances in Figure 9 compared quite well with the same case run in simulation which is shown in Figure 5. Similarly, the simulation and experimental results for step pressure decrease of 40N also contains convergence time of approximately 100 to 150 iterations. However, those figures are only listed in [27] due to the limited space in this article.

2) Ramp disturbance: During ramp disturbances, the controller can be misled by obsolete vertices that carries lower cost value and produces a series of inferior trial points. Fortunately, since the disturbance is assumed to be much slower than the valve travel time, the rate of ramp disturbance is limited. Figure 10 shows the increasing pressure ramp of 30 N over 300 steps does not affect the terminal conditions due to the controller regulation. The experimental results in Figure 10 correspond to the simulation results in Figure 6 and again they match closely. For the negative ramp case, the controller also handles up to the rate of 10N per 100 steps both in simulation and in experiments [27].

C. Adjust Cost Function for Power Consumption

By modifying the cost function with an additional energy term the Nelder Mead controller can be directed to reduce energy consumption as well.

$$F(c) = \left(\alpha(v^f_{appr} - v_d)^2 + \beta(i^f_{appr} - i_d)^2\right) + \gamma \int_0^{t_{end}} i(t)dt$$

Figure 11 shows that by increasing the current weight γ in the cost function $F(c)$, the current integral can be reduced without introducing errors to the v^f_{appr} and i^f_{appr}. Using the modified cost function, the current integral per valve event can reduced from 0.17 to 0.166 ampere-seconds. While current integral is used in $F(c)$ to represent the area underneath the spline curve $i(c)$, other energy terms such as $\int i^2(t)$ should achieve similar energy reduction.

D. Comparison with results from published literature

Because the authors can not find any results that focuses only on the approach control, we have to defer the comparison with the existing controllers after both landing and approach controller are integrated together.

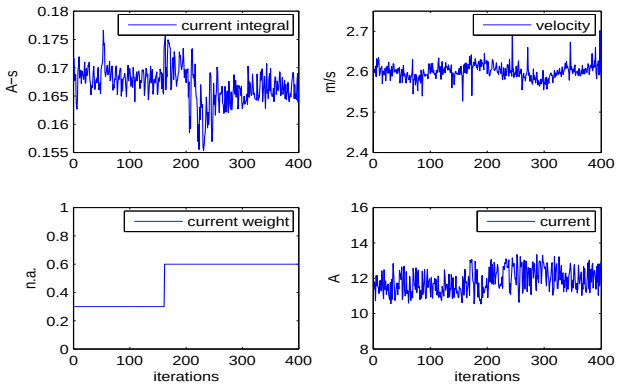

Fig. 11. Cost function experiment with current-integral weight changes: current integral (top left), current weighting in cost function (bottom left), approach control end velocity (top right), end current (bottom right)

VI. CONCLUSIONS AND FUTURE WORK

A. Conclusions

A cyclic adaptive control using Nelder Mead algorithm is used to adapt the input current profile of an electromagnetic valve actuator (**EMV**). The objective is to regulate the current and velocity at the position of 2.55mm, where the approach control ends and landing control starts. By ensuring the landing trajectory starting nearby the desired initial conditions, approach control simplifies the valve-seating problem for the landing controller. The simulation and the experiment result shows that the approach controller can eliminate effect of step pressure change of up-to 40N in 100 to 150 steps and ramp pressure disturbance up to the rate of 10N per 100 steps. Also, the energy consumption can be reduced if an energy term is added to the cost function. In conclusion, solving the EMV approach control problem with a Nelder Mead controller produces satisfactory results in terms of optimization capability, robustness against disturbance, and computational efficiency.

B. Future Work

The feedforward approach controller presented here should be combined with a feedback landing controller (e.g. [13]). The overall controller can then be examined for impact velocity, valve transition time, and bounce. Also, testing with actual engine gas pressure should be performed. Lastly, while our results may suggest integral action for Nelder Mead algorithm, more analysis is required to establish a steady state error bound at the end of EMV approach control.

VII. ACKNOWLEDGMENTS

The authors would like to thank Daimler AG for the donation of the solenoid actuator. Also, the authors would like to thank Auto21 project for providing research funding.

REFERENCES

[1] H. Hong, G.B. Parvate-Patil, and B. Gordon. Review and analysis of variable valve timing strategies - eight ways to approach. *Proceedings of the Institution of Mechanical Engineers, Part D (Journal of Automobile Engineering)*, 218(D10):1179 – 200, 2004/10/.

[2] M. Pischinger, W. Salber, F. van der Staay, H. Baumgarten, and H. Kemper. Benefits of the electromechanical valve train in vehicle operation. *SAE paper 2000-01-1223*, 2000.

[3] M. Schechter and M. Levin. Camless engine. *SAE paper 960581*, 1996.

[4] Z. Sun and D Cleary. Dynamics and control of an electro-hydraulic fully flexible valve actuation system. *Proceedings of the American Control Conference*, 4:3119–3124, 2003.

[5] Parlikar T., Chang W., Qiu Y., Seeman M., Perreault D., Kassakian J., and T. Keim. Design and experimental implementation of an electromagnetic engine valve drive. *IEEE/ASME Transactions on Mechatronics*, 10:482–494, 2005.

[6] C. Weddle and D. Leo. Embedded actuation system for camless engines. *Proceedings of the International Confrence on Adaptive Sturcture and Technologies*, 1998.

[7] C.R. Koch, A.F. Lynch, and R. R. Chladny. Modeling and control of solenoid valves for internal combustion engines. *2nd IFAC Conference On Mechatronic Systems*, 41:317–322, 2002.

[8] Y. Wang, T. Megil, M. Haghgooie, K. Peterson, and A. Stefanopoulou. Modeling and control of electromechanical valve actuator. *SAE paper 2002-01-1106*, 2002.

[9] Soon K. Chung. Flatness-based end-control of a gas exchange solenoid actuator for ic engines. Master's thesis, University of Alberta, 2005.

[10] J. Savage and J. Matterazzo. Application of design of experiments to determine the leading contributors to engine valvetrain noise. *SAE paper 930884*, 1993.

[11] K. Peterson and A. Stefanopoulou. Extremum seeking control for soft landing of an electromechanical valve actuator. *Automatica*, 2004.

[12] R. R. Chladny and C. R. Koch. Flatness-based tracking of an electromechanical vvt actuator with disturbance observer feed-forward compensation. *IEEE Transaction on Control System Technology*, 16:652–663, 2008.

[13] S.K. Chung, C.R. Koch, and A.F. Lynch. Flatness-based feedback control of an automotive solenoid valve. *IEEE Transactions on Control Systems Technology*, 15(2):394 – 401, 2007/03/.

[14] S. Butzmann, J. Melbert, and A. Koch. Sensorless control of electromagnetic actuators for variable valve train. *SAE paper 2000-01-1225*, 2000.

[15] C. Tai and T. Tsao. Control of an elctromechanical actuator for camless engines. *Proc. 2002 American Control Conference*, 2002.

[16] Ch. Gunselmann and J. Melbert. Improved robustness and energy consumption for sensorless electromagnetic valve train. *SAE paper 2003-01-0030*, 2003.

[17] İ. Haskara, V. Kokotovic, and L. Mianzo. Control of an elctromechanical valve actuator for a camless engine. *International Journal of Robust and Nonlinear Control*, 14:561–579, 2004.

[18] Peter Eyabi and Gregory Washington. Modeling and sensorless control of an electromagnetic valve actuator. *Jounal of mechatronics*, 16:159–175, 2006.

[19] C. Tai and T. Tsao. Control of an elctromechanical actuator for camless engines. *Proc. 2003 American Control Conference*, 2003.

[20] Wolfgang Hoffmann, Katherine Peterson, and Anna G. Stefanopoulou. Iterative learning control for soft landing of electromechanical valve actuator in camless engines. *IEEE trans on control system technology*, 11(2):174–184, 2003.

[21] Katherine S. Peterson. *Control Methodologies for Fast and Low Impact Electromagnetic Actuators for Engine Valves*. PhD thesis, University of Michigan, Ann Arbor, 2005.

[22] R. R. Chladny, C. R. Koch, and A. F. Lynch. Modeling automotive gas-exchange solenoid valve actuator. *IEEE trans. on magnetics*, 41(3):1155–1162, 2005.

[23] J.C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.

[24] R.M. Lewis, V. Torczon, and M.W. Trosset. Direct search methods: then and now. *J. Comput. Appl. Math. (Netherlands)*, 124(1-2):191 – 207, 2000/12/01.

[25] F. Sigworth. State-machine simplex minimizer. Availabe on MATLAB Central File Exchange, 2003.

[26] J. D'Errico. Understanding fminsearchbnd. Availabe on MATLAB Central File Exchange, 2005.

[27] Jimmy Tsai. Nelder mead approach control for the electromagnetic valve actuator. Master thesis, Simon Fraser University, December 2007.