Extremum Seeking Methods for Optimization of Variable Cam Timing Engine Operation

Dobrivoje Popović, Mrdjan Janković, *Fellow, IEEE*, Steve Magner, and Andrew R. Teel, *Fellow, IEEE*

***Abstract*—Automotive engines have been increasingly equipped with devices that can vary engine parameters with operating conditions. Optimizing these parameters via standard engine mapping and calibration has become a very time consuming task. The specific problem under consideration in this paper is optimization of intake, exhaust, and spark timings to improve the brake specific fuel consumption of a dual-independent variable cam timing engine. We have explored extremum seeking (ES) as a method to find the optimal setting of the parameters. During ES, the engine is running at fixed speed and torque in a dynamometer test cell, while an optimization algorithm is iteratively adjusting the three parameters. For our purposes, we have modified several algorithms available in the literature. The details about the algorithms and the experimental results they produced are presented and discussed in the paper.**

***Index Terms*—Extremum seeking (ES), gradient search, variable valve timing.**

I. Introduction

A. Motivation and Problem Description

TODAY'S automotive engines come with many devices that can vary engine parameters with operating points (typically specified by engine speed and output torque) to improve emissions, fuel consumption, and/or peak torque. Among these devices are external exhaust gas recirculation, electronically controlled throttle, variable cam timing, variable valve lift, cam profile switching, etc. In contrast, in conventional engines parameters such as cam timing or valve lift are fixed at the design stage.

The standard strategy of an on-board electronic control unit (ECU) is to assign parameter values that are optimal for the current operating point. Only a finite number of optimal parameter values indexed by operating point (speed and torque) are predetermined in the calibration process and stored in lookup tables. For other operating points, optimal parameters are interpolated from the lookup table values. When the number of parameters is relatively low, conventional grid-based engine mapping can be applied to find optimal values to fill the lookup tables. For a fixed preselected speed/torque operating point, the procedure involves dividing the parameter space into a grid, measuring the engine performance for each grid knot, and then selecting the best values out of the tested ones (or alternatively, creating a regression from the data and finding the optimal parameter values from the regressed model). Because the experimental burden increases exponentially with the number of parameters, this procedure becomes very time consuming even in the case of three parameters. In such cases, extremum seeking (ES) can be used to quickly and directly locate optimal parameters. We demonstrate this on a dual-independent variable cam timing (di-VCT) engine.

In a di-VCT engine we can independently change three timing parameters, where the timing (phase) is measured relative to the crankshaft (and piston) position. The parameters are the spark ignition, opening/closing of the intake valves, and opening/closing of the exhaust valves. Because the cam profiles are fixed, so is the duration between timing of opening and closing of a valve; therefore, changing the timing of valve opening changes the timing of valve closing by the same amount. The tuning of these three parameters can bring several benefits. By varying the intake and exhaust cam timing, an appropriate amount of the residual exhaust gas can be recirculated into the cylinder thereby suppressing NOx formation [2]–[4]. Furthermore, this residual contains some unburned hydrocarbons; consequently, retaining it in the cylinder through two combustion cycles also reduces hydrocarbon emissions [5]. Late intake valve closing reduces engine volumetric efficiency, hence, resulting in part-load operation at higher intake manifold pressures and reduced pumping losses [6]. Note that to achieve these benefits, spark timing must be tuned together with the other two parameters, since these two significantly affect the burn rate through varying in-cylinder dilution and turbulence. More details about operation of variable cam timing engines can also be found in [7].

In this work we were interested in optimizing the engine's brake specific fuel consumption (BSFC) alone or in combination with a constraint on combustion stability. We implemented several optimization algorithms, tested each on the engine at different operating points (specified here by engine speed and output torque) to locate the optimal parameters, and compared the results they produced to the results of a more conventional engine mapping process.

By installing an experimental di-VCT engine in a dynamometer test cell and conducting a number of ES experiments at various engine speeds and low to medium torques we were able to show the following.

- The results were repeatable from run to run, algorithm to algorithm.

Manuscript received December 16, 2003. Manuscript received in final form November 18, 2005. Recommended by Associate Editor Y. Jin. This work was supported in part by Air Force Office of Scientific Research (AFOSR) under Grant F49620-00-1-0106 and by Ford Motor Company. A shorter version of this paper appeared at the *2003 American Control Conference,* Denver, CO, 2003.

D. Popović and A. R. Teel are with the Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 USA (e-mail: doca@ece.ucsb.edu; teel@ece.ucsb.edu).

M. Janković and S. Magner are with Ford Research and Advanced Engineering, Dearborn, MI 48121 USA (e-mail: mjankov1@ford.com; smagner@ford.com).

Digital Object Identifier 10.1109/TCST.2005.863660

- These results agreed with the results obtained by engine mapping.
- The algorithms were able to quickly locate a local minimum, taking about 15 min on average.

Higher torques were avoided since the algorithms did not have a built-in procedure that handles engine knock.

Finally, it is worth mentioning that ES methods are exceptionally well suited for situations when detailed knowledge of response surfaces in the parameter space are not needed. For example, at the engine design stage, "mini-maps" needed to evaluate effects of various hardware modifications (e.g., cam profiles, compression ratios) on cycle average fuel consumption or emissions can be efficiently obtained by ES. A second example is the calibration of a control strategy proposed in [8] that does not rely on the grid-based lookup tables for response surface representation.

B. Historical Perspective

ES has a long history and was a popular subject of adaptive control in the 1950s and 1960s. One of the first works on the subject was a classic 1951 paper by Draper and Li [9] and [24]. They introduced the principles of the so-called "Self-optimilizing Control Systems" through one automotive application: an (analog) control system for maximizing output torque of an engine running at constant speed with constant fuel flow by adjusting air flow and spark timing.

Over the years, interest in ES in the automotive field slowly diminished. The work was mainly focused on adaptive spark timing adjustment in vehicles [10], [11]. In this work we show that ES is a viable and fast option for multi-parameter engine optimization in test cells.

There are several existing software packages that include ES, such as the one described in [12] that combines design of experiments (DOE) methods [13] with "local" extremum seeking. In this case, the ES procedure relies on a good initial guess from the DOE part for the optimization. Our concept, however, is an ES tool that does not rely on any prior results of engine mapping (or DOE).

II. Extremum Seeking

ES is an iterative optimization process performed in real time on a physical system. The function being optimized is the steady-state relationship between the system's input parameters and its performance output. This function, denoted here by $f(\cdot)$, is usually called the response map. Since $f(\cdot)$ is not known (otherwise, it could be optimized in a computer), ES algorithms rely only on its measurements to search for the optimum. Starting from some initial parameter values, such an algorithm iteratively perturbs the parameters, monitors the response, and adjusts the parameters toward improved performance. This process runs usually as long as improvement is being produced.

Aside from algorithmic steps, ES software also performs sequencing of input commands, and collection and filtering of measurements. The software usually assigns the parameter values in step increments. Because the map is generated by a dynamical system, the response of $f(\cdot)$ to a change in input parameters can be measured only after some time has passed, usually after large transients die down. In addition, the measurements are usually noisy and require filtering, which takes additional time. For example, with simple averaging we can get

$$f(x) \approx \frac{1}{T_2} \int_{t_0+T_1}^{t_0+T_1+T_2} y(\tau) d\tau. \tag{2.1}$$

Here, the inputs are fixed at x from t_0 to $t_0 + T_1 + T_2$; T_1 is the waiting time, T_2 is the *averaging* or filtering time, and y is the measured performance output. Note that the transient errors can be controlled via T_1, and the noise errors via T_2.

In ES applications, the goal is to locate optimum in minimum time. This is achieved by an appropriate choice and design of algorithms and by their tuning; waiting and filtering times must be manipulated (and thereby, more or less measurement error allowed) together with other algorithm parameters to produce fastest ES [14]. Note that by manipulating waiting time, we control the dynamical interaction between the optimized system and the algorithm. If the waiting time is reduced, the system is not fully settled down at times when a new measurement is taken (and a new command assigned). It can be shown that under broad conditions this interaction remains of small-gain type, which assures that the ES process will remain stable and that the parameters will converge to the optimum [14].

A. Algorithms Used in This Work

Let $x = [x_1\ x_2\ x_3]^\top$ be a vector with entries x_1, x_2, and x_3 corresponding to the parameters of our engine optimization problem: intake cam timing, exhaust cam timing, and spark timing, respectively. The cam hardware constrained (see Section III) x_1 to the interval $[-30, 30]$ degrees after top dead center (ATDC) and x_2 to the interval [0, 40] degrees ATDC. Let the function $f : \mathbb{R}^3 \to \mathbb{R}$ represent the BSFC response map of the engine, which is a function of the three optimization parameters, and is measured in real time.

The applied algorithms iteratively search for optimal parameters using a recursion of the form

$$x_{k+1} = x_k + \alpha_k \cdot \text{step}\,(x_k). \tag{2.2}$$

Here $k = 1, 2, \ldots$ is the iteration number, x_k is the current estimate of the optimal parameters, the positive number α_k is a *step size* or *gain*,[1] and step(x_k) is a vector calculated from estimates of partial or directional derivatives of $f(\cdot)$ at x_k. The algorithms use symmetric two-sided finite differences to estimate partial, or directional (along some vector v) derivatives

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x + \lambda e_i) - f(x - \lambda e_i)}{2\lambda}$$
$$\nabla f(x)^\top v \approx \frac{f(x + \lambda v) - f(x - \lambda v)}{2\lambda} \tag{2.3}$$

respectively. Vector e_i denotes a unit vector with 1 in the ith place, and 0's elsewhere, *dither* λ is a small nonzero constant,

[1] We call α a step size when step is a unit vector. When norm of step varies with measurements, we call α gain, which conforms to the control systems terminology.

and v is a *perturbation* vector. Computation of one finite difference requires two new measurements of $f(\cdot)$.

In this work we applied the simultaneous perturbation stochastic approximation (SPSA) algorithm by Spall [15], and the related Persistently exciting finite differences (PEFD) algorithm used by Teel [16] for nonsmooth optimization problems. SPSA and PEFD use one finite-difference per iteration. We also applied the Box and Wilson steepest descent (BWSD) [17] algorithm, which needs up to three finite-differences per iteration. For various reasons we had to substantially modify the off-the-shelf versions of these algorithms, and we will describe that in detail. Also note that we applied versions of these algorithms suited for our parameter constraints.

Comments on Our Selection of Algorithms: Our expectation was that SPSA and PEFD would have required a comparatively low number of measurements to locate optimal parameters, and hence would have produced fast ES, and this turned out to be the case. This expectation was supported by the experience of the others in a number of SPSA applications [18]. PEFD had not been applied before, but the form we used in this work differs from SPSA only in the way it selects perturbation vectors, so we expected good performances too. The performance standard for us was the classical Kiefer Wolfowitz Stochastic Approximation (KWSA) [19]. KWSA requires estimation of all the gradient components at each iteration, which, in our case requires three finite-differences and six measurements per iteration. SPSA and PEFD use an estimate of a directional derivative (one finite-difference) along the currently selected direction to produce an estimate of the gradient. Spall in [15], theoretically showed that SPSA, at least asymptotically, needs $2n/2 = n$ times fewer measurements than KWSA to locate an optimum, n being the number of optimized parameters. We applied KWSA to our problem and found that it needed fewer iterations than SPSA/PEFD to converge, but significantly more measurements overall. In fact, KWSA was close to three times slower, as predicted by the theory. Another classical algorithm we tried out is the BWSD, which matched the speed of SPSA and PEFD after it was substantially modified.

We also decided against the very popular, though heuristic, Nelder–Mead simplex algorithm [20]. In an n-dimensional problem, this algorithm needs one or two measurements to produce an improved point in a typical iteration; the new measurement points are chosen based on the best $n+1$ of the measurements collected up to that moment (which can be seen as the vertices of a simplex in an n-dimensional space). We dropped this algorithm for two reasons. First, it is not clear how to adapt it to constrained problems, especially when the optimum lies on the boundary. Second, to achieve fast optimization, and successfully cope with noise, the algorithm requires comparatively larger steps between experiments. This is undesirable when the algorithm drives a dynamical system, since it can produce large transients.

Finally, we considered the sinusoidal perturbation algorithm (see, for example, [9, Sec. V], [24], [21]), which has been recently frequently discussed in the control literature. The main reason we did not test this algorithm is that it was not clear how to modify it for the constrained problems (to our knowledge, such a modification still does not exist).

1) SPSA and PEFD: For this application we selected one common form of SPSA and PEFD [see (2.4)] with projection. The original algorithms had to be modified because of certain problems that were preventing them from fast convergence. We will first describe what we selected from the literature and then explain the modification.

At each iteration $k = 1, 2, \ldots$, both PEFD and SPSA implementations select perturbation vector $v_k = [\Delta_{k1}\Delta_{k2}\Delta_{k3}]^\top$ out of the pool of vectors

$$\left\{[-1\;1\;1]^\top, [1\;-1\;1]^\top, [1\;1\;-1]^\top, [1\;1\;1]^\top\right\}.$$

The only difference between the two implementations is that SPSA selects the perturbation vectors v_ks randomly (each vector from the pool is assigned the same probability), and PEFD periodically, in a round robin fashion.[2] These implementations satisfy general requirements for v_k: for the SPSA, the Δ_{ki} components of the vector v_k need to be generated stochastically as independent random variables, usually Bernoulli symmetrically distributed (flipping a coin decides whether $\Delta_{ki} = +1$ or $\Delta_{ki} = -1$), in the case of PEFD, the condition is that the sequence of v_ks is *persistently exciting*, which is satisfied when the sequence periodically spans the parameter space.

Once v_k is selected, the algorithms perturb the current estimate x_k along v_k to collect two measurements of the response map, $f(x_k + \lambda_k v_k)$ and $f(x_k - \lambda_k v_k)$, and then update the estimates by, first, calculating

$$\tilde{x}_{k+1} = x_k - \alpha_k v_k \frac{f(x_k + \lambda_k v_k) - f(x_k - \lambda_k v_k)}{2\lambda_k} \tag{2.4}$$

then, projecting $\tilde{x}_{k+1}$ onto the constrained set (the standard SPSA form [15, eq. (2.2)] can be obtained from (2.4) if we observe that $v_k = [\Delta_{k1}\ \Delta_{k2}\ \Delta_{k3}]^\top = [(1/\Delta_{k1})(1/\Delta_{k2})(1/\Delta_{k3})]^\top$ when $\Delta_{ki} = \pm 1$). Note that in our case, the projection operator $\pi\{\cdot\} : \mathbb{R}^p \to [-30, \ldots, 30] \times [0, \ldots, 40] \times \mathbb{R}$, is a simple truncation. We get

$$x_{k+1} = \pi\{\tilde{x}_{k+1}\}. \tag{2.5}$$

SPSA with projection is analyzed in [22].

We now present the modification. Combining SPSA or PEFD with projection produces an undesirable effect that we can very loosely describe as "bouncing of parameter estimates against the constraints." This effect slows down the convergence when the optimum is located on the constraints, which was the case in our application. The bouncing effect and an appropriate modification for SPSA are detailed in [23]; here we only briefly describe one iteration of the modified algorithm.

1) Given $x_k = [x_{k1}\ x_{k2}\ x_{k3}]$, separate those parameters, x_{ki}, $i = 1, 2, 3$, that are on the constraints from those that are not.
2) Generate $[\Delta_{k1}\ \Delta_{k2}\ \Delta_{k3}]^\top$.
3) Randomly decide whether to perturb just one of the parameters on the constraints, or simultaneously perturb all of the ones that are inside the constrained set, e.g., if

[2] Periodic selection was also discussed in the SPSA context in [15].

$x_{k1} = 30$ or $x_{k1} = -30$ then v_k can be either $[\Delta_{k1}\ 0\ 0]^\top$ or $[0\ \Delta_{k2}\ \Delta_{k3}]^\top$.

4) Update parameters, for $i = 1, 2, 3$, according to

$$\tilde{x}_{(k+1)i} = x_{ki} - \alpha_k \gamma_{ki} v_{ki} \frac{f(x_k + \lambda_k v_k) - f(x_k - \lambda_k v_k)}{2\lambda_k} \tag{2.6}$$

where γ_{ki} is a scaling parameter that depends on x_k and the way the algorithm made a random decision in step 3.

5) Apply projection, i.e., $x_{k+1} = \pi\{\tilde{x}_{k+1}\}$.

In the stochastic context (noisy measurements), a sequence produced by the modified SPSA will converge to a Kuhn–Tucker point with probability one; this was shown in [23]. Similar modification was implemented for PEFD; the analysis of convergence properties of PEFD with projection is a subject for future research.

2) Modified Box and Wilson Algorithm: This algorithm is based on the steepest descent procedure introduced by Box and Wilson in their seminal 1951 paper on process improvement [17]. Each iteration of this procedure has two stages.

1) The negative direction of the gradient is estimated: $v_k = -(\tilde{\nabla} f(x_k)/|\tilde{\nabla} f(x_k)|)$, when $|\tilde{\nabla} f(x_k)| > \varepsilon$; The procedure stops when the gradient, i.e., linear model, cannot adequately represent the function curvature, usually when it becomes too small; "small" is controlled here by the parameter ε.

2) The estimates are updated in direction of v_k in steps of fixed size α as long as there is an improvement: if $f(x_k + \alpha v_k) < f(x_k)$ then $x_{k+1} = x_k + \alpha v_k$, $v_{k+1} = v_k$. Otherwise, $x_{k+1} = x_k$ and the procedure returns to gradient estimation.

The algorithm was modified so that, in the first stage, it uses three two-sided finite differences to estimate the gradient, rather than a more complicated DOE based estimate. Each component was estimated from: $\partial f/\partial x_i(x_k) \approx (f(x_k + \lambda e_i) - f(x_k - \lambda e_i))/2\lambda$, $i = 1, 2, 3$. The algorithm does not use additional measurements to verify that a good estimate of the gradient is obtained, although this was done in the original procedure.

In the second stage, after the first and before each of the subsequent update steps, the third (spark) component of the gradient is re-estimated, which requires two measurements, $\partial f/\partial x_3(x_k) \approx (f(x_k + \lambda e_3) - f(x_k - \lambda e_3))/2\lambda$, and the unit update direction v_{k+1} recalculated. By this, we exploit a property of the performance map: spark sensitivity varies much more with parameters than with two other components of the gradient. As a result, the algorithm performs more second stage updates (with two measurements each) before returning to full gradient estimation (with six measurements). To additionally speed up ES, the central value, $f(x_k)$ is not measured but estimated as the average of two values obtained during the estimation of spark sensitivity: $f(x_k) \approx (1/2)(f(x_k + \lambda e_3) + f(x_k - \lambda e_3))$.

The last implemented modification is first-step line search. If the first step after gradient estimation is not producing an improvement, the next estimate is sought between x_k and $x_k + \alpha v_k$. The algorithm explores $(1/2)(f(x_k + (\alpha/2)v_k + \lambda e_3) + f(x_k + (\alpha/2)v_k - \lambda e_3))$, $(1/2)(f(x_k + (\alpha/4)v_k + \lambda e_3) + f(x_k + (\alpha/4)v_k - \lambda e_3)$, etc., until one of the values, say the one corresponding to $x_k + (\alpha/i)v_k$, shows an improvement; then $x_{k+1} = x_k + (\alpha/i)v_k$; if no improvement was produced after several cuts of α, old parameters are kept: $x_{k+1} = x_k$.

At each update, violations of the constraints are checked and parameters are projected if needed. When the estimates hit a new constraint, the algorithm returns to the first stage and full gradient estimation.

Note that this is a heuristic algorithm, and that rigorous convergence analysis does not exist.

III. Implementation

An experimental di-VCT engine was installed in a dynamometer test cell at Ford Research Laboratory. The hardware configuration consisted of the engine attached by a common shaft to the dynamometer (electric motor/generator) that maintained constant rotating speed. Constant torque was maintained by a controller that actuates the electronically controlled throttle based on measurements from an inline torque sensor. The phase shift of the cam-shafts relative to the crank-shaft was regulated by electro-hydraulic actuators. In this way, the actuators could continuously vary the timings for intake cam opening (ivo) and exhaust valve closing (evc) (ivo could be varied within $[-30, \ldots, 30]$, and evc within $[0, \ldots, 40]$ degrees ATDC).

For control algorithm development, we used combination of Simulink/Stateflow and a dSPACE[3] rapid prototyping system, in conjunction with other computer-aided system engineering and computer-aided control system design tools.

During experimental runs, the dSPACE processor executed the ES code as well as local control loops for intake and exhaust cam timing and electronic throttle. An existing electronic control unit (ECU) capable of running the engine was already available and was used for closed loop air-fuel ratio regulation and to control spark timing and fuel injector pulsewidths. The dSPACE system and ECU communicated through a shared memory board. Fig. 1 depicts the configuration of our setup.

IV. Experimental Results

We bring experimental results in four parts. The first part brings the results obtained by conventional engine mapping, which we performed separately. The second part describes the way we processed the measurements for ES. That part should distinguish the difference between processing of the measurements for ES and engine mapping. In the third part, we present results of ES for the best fuel consumption. The goal is to compare the optimal parameters as located by ES with those obtained by engine mapping. The final part deals with optimization of fuel consumption constrained by combustion quality.

A. The BSFC Response Surface Obtained by Engine Mapping

Before running the ES experiments, we mapped the engine at several operating points specified by fixed speed and output torque. The parameter space of timings for intake valve opening (ivo), and the exhaust valve closing (evc) was spanned by a 35-point grid $\{-30, 20, 10, 0, 10, 20, 30\} \times \{0, 10, 20, 30, 40\}$, and at each point of the grid, a spark-sweep was performed and the engine performance was measured and documented. Spark

[3] Simulink and Stateflow are trademarks of Mathworks Inc. dSPACE is the trademark of dSPACE GmbH.

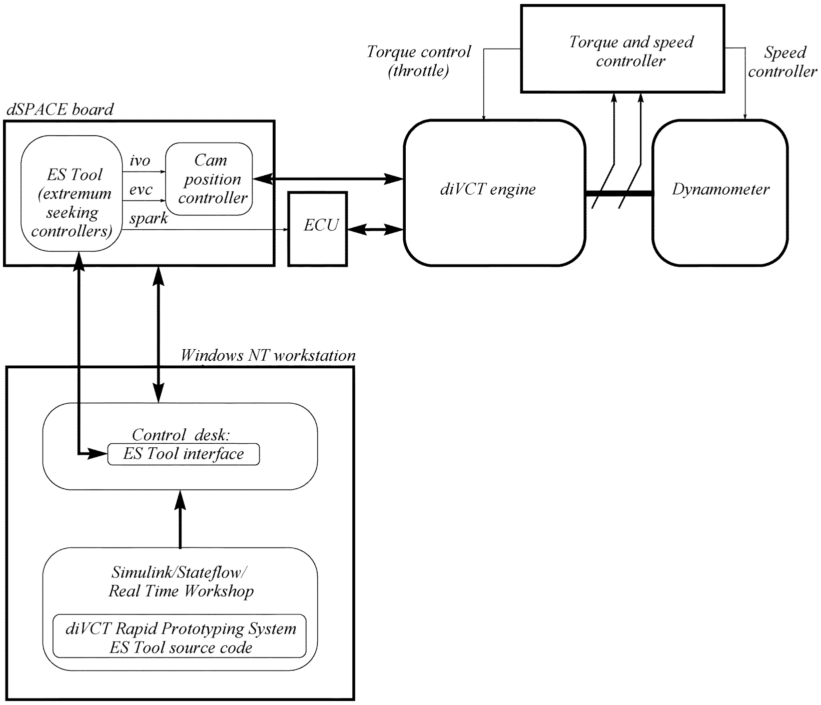


Fig. 1. Configuration of the experimental hardware for ES experiments.

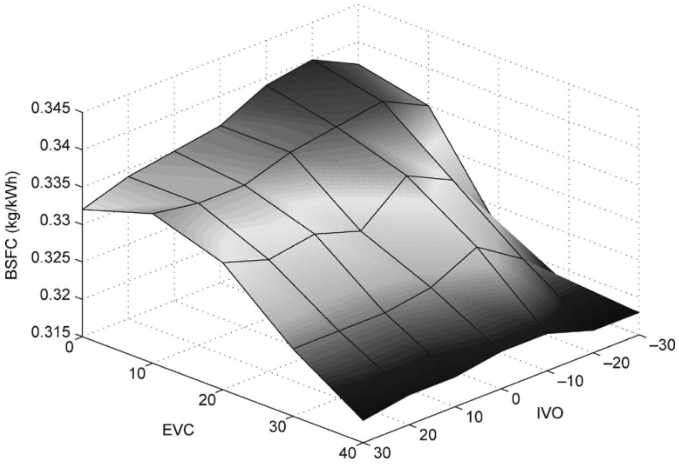


Fig. 2. BSFC for optimal spark versus cam timing. Engine mapping at 1500 rpm and 62.5 N·m. The point $(\mathrm{ivo}, \mathrm{evc}) = (-30, 40)$ has not been mapped due to unstable combustion, and was extrapolated in the figure.

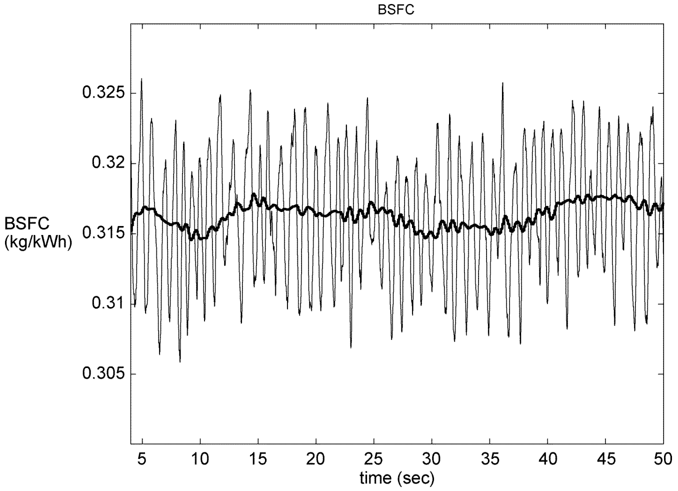


Fig. 3. BSFC signal used for ES. Signal before (plain) and after (bold) the 3 s moving average filter. Engine speed was 1500 rpm, output torque 62.5 N·m.

sweep means measuring engine variables at seven to ten values of spark timing selected around a subjective optimal, maximum brake torque (MBT), spark timing, while the other engine parameters are kept constant. This “full-factorial” mapping process is very time consuming since for each new value of the three parameters, the measurements are averaged for two minutes after the temperature transcients have settled down (which takes typically another 1–2 min).

The response surface in Fig. 2 obtained by the procedure outlined above represents a typical shape of the di-VCT brake specific fuel consumption as a function of ivo and evc. To produce the two-parameter plot, for each pair of cam timing values, the spark timing was set to the corresponding optimum MBT value. Fig. 2 represents measurements taken at 1500 rpm, 62.5 N·m. The BSFC values for points which are not grid knots were interpolated.

We can locate two local minima in the figure. The optimal values for cam timing, $(\mathrm{ivo}, \mathrm{evc})$, are (30, 40) and (−20, 40). Here, the point (−20, 40) borders the region of unstable combustion; measurements at the point (−30, 40) were not taken due to combustion instability (see Section IV-D for more details). Note the following.

- The optimal points are located on the constraint $\mathrm{evc} = 40$.
- At 62.5 N·m, the difference in BSFC among the points with $\mathrm{evc} = 40$ minimal, less than 0.5%. This creates an opportunity to combine BSFC with some other criterion to select the best point. One option is to use emissions, which should be a subject of further work.

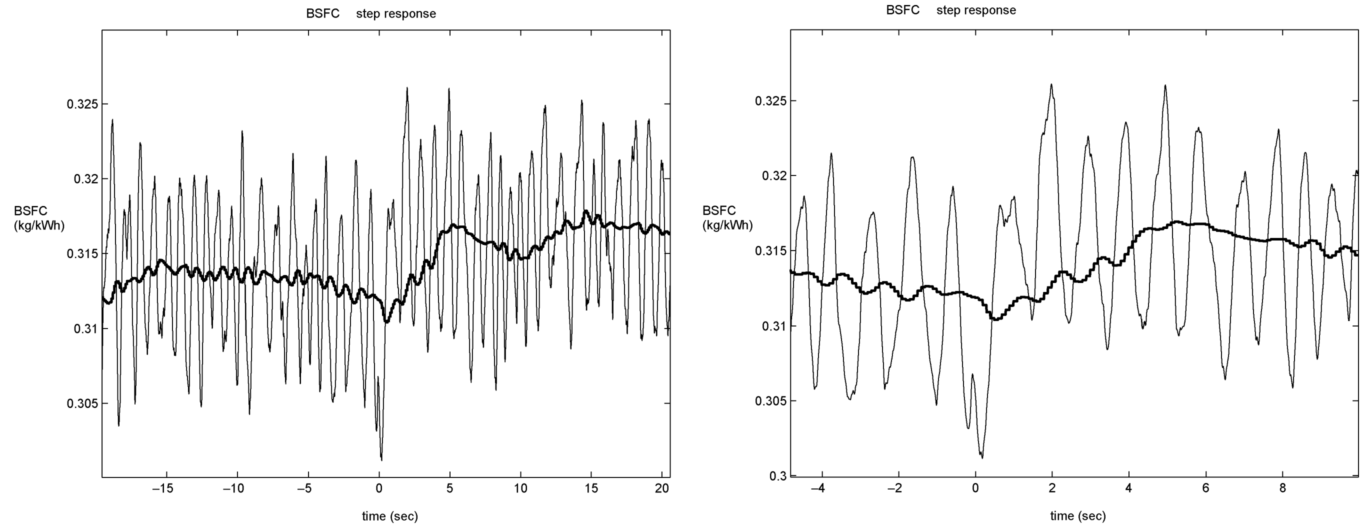


Fig. 4. Response in the BSFC signal to a step change in cam and spark timings. Initial $(\mathrm{ivo}, \mathrm{evc}, \mathrm{spark})$ were (10,10,30) degrees, final (20,20,35). The filtered signal is the bold curve.

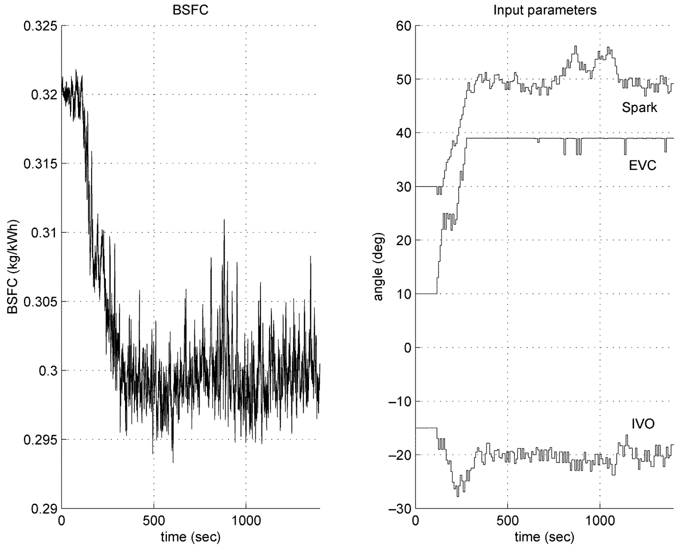


Fig. 5. SPSA, 1500 rpm, 62.5 N·m. Run A. Initial values $(\mathrm{ivo}, \mathrm{evc}, \mathrm{spark}) = (-15, 10, 30)$.

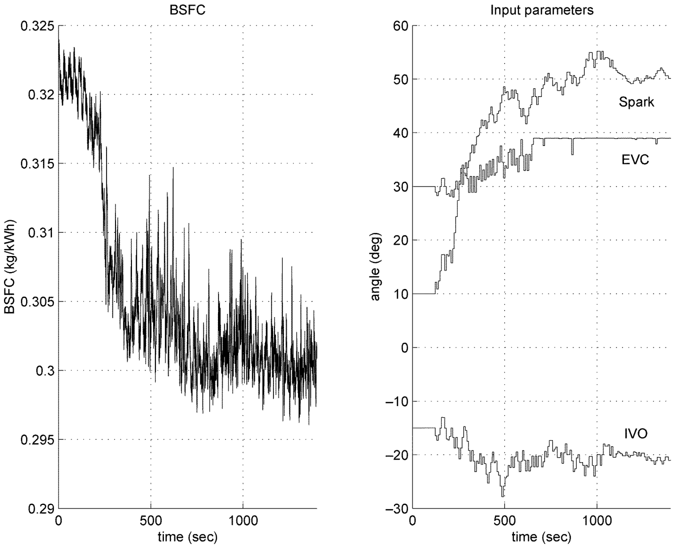


Fig. 6. SPSA, 1500 rpm, 62.5 N·m. Run B. Initial values $(\mathrm{ivo}, \mathrm{evc}, \mathrm{spark}) = (-15, 10, 30)$. Note at $t \approx 1100$, the algorithm gain was reduced to a half of its initial value.

B. BSFC Measurements for ES

To update input parameters, an ES algorithm collects several BSFC measurements per iteration, as described in Section II. Because the value of BSFC is changing during engine transients, the ES algorithm waits for one second after changing parameter settings $(\mathrm{ivo}, \mathrm{evc}, \text{and } \mathrm{spark})$, then averages the BSFC signal for 3 s. We have found that the combination of one second waiting, followed by the three second averaging (4 s per measurement) produced the fastest ES. In contrast, it typically takes several minutes to get one measurement of an engine map by the procedure described earlier. The difference can be explained by the desire in conventional mapping to have accurate readings at all recorded points, while in ES we insist only on the final "optimal" readings being accurate. The intermediate measurements can be inaccurate provided that, on the average, this does not affect the direction of the parameter updates.

The BSFC was calculated from the following formula:

$$\mathrm{BSFC} = \frac{\text{Fuel flow}}{\text{Speed} \cdot \text{Torque}} \tag{4.1}$$

where the three variables were measured with 1-kHz sampling frequency and then passed through a 1.63-Hz low-pass filter. This signal is then passed through a moving average filter (with the 3 s window mentioned above and 10-Hz sampling). The BSFC signal before and after the moving average filter is shown in Fig. 3. There are several points we would like to make about these figures and the BSFC signal, more specifically about disturbances and noise.

- The fuel flow information was calculated from the ECU generated fuel-per-stroke signal. This ECU signal had faster response than the signal coming from the fuel flow

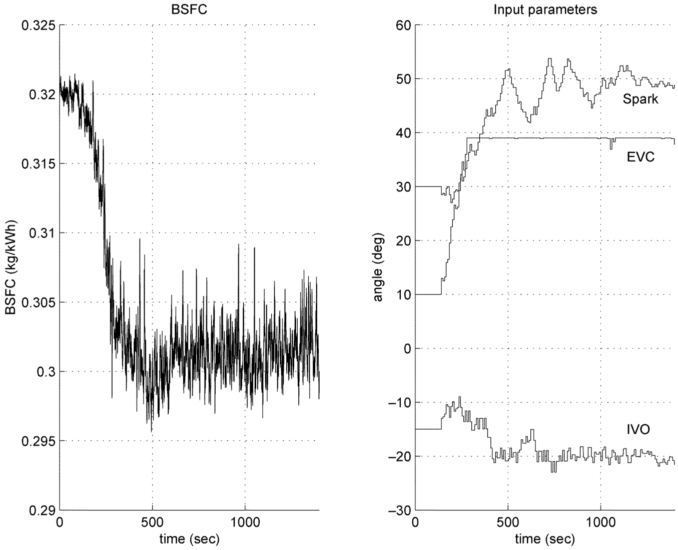


Fig. 7. PEFD, 1500 rpm, 62.5 N·m. Initial values $(\mathrm{ivo}, \mathrm{evc}, \mathrm{spark}) = (-15, 10, 30)$. Note at $t \approx 1200$, the algorithm gain was reduced to half its initial value.

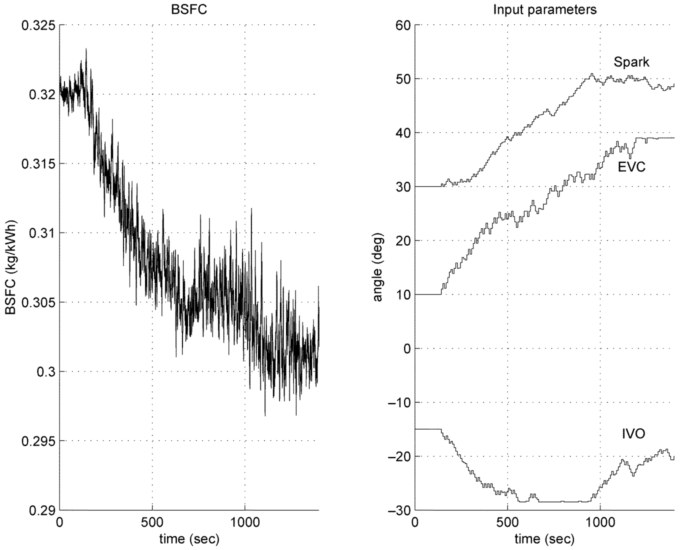


Fig. 8. PEFD, 1500 rpm, 62.5 N·m. Low gain. Initial values $(\mathrm{ivo}, \mathrm{evc}, \mathrm{spark}) = (-15, 10, 30)$. Note that the algorithm gain had a reduced value from the start.

sensor, which was used to get the BSFC during mapping. There was an offset between the two fuel measurements.

- The air/fuel ratio was kept constant by the ECU, which used information from the production heated gas oxygen (HEGO) sensors. The HEGO sensor is of relay type and produces residual oscillations (several percent amplitude, 1 Hz frequency) in the fuel flow, as it can be seen from the unfiltered BSFC signal in Fig. 3. The moving average filter was able to filter most of that out.
- The filtered signal contained a relatively large, low-frequency additive noise component of unknown origin with a period of approximately 25 s. Most of it was filtered out by using finite-differences, (a difference of two measurements that are separated by approximately 4 s acts as a high-pass filter).

Our algorithms were able to cope with these noise problems successfully. However, we expect that use of universal exhaust gas oxygen sensors (which measure continuously) for air/fuel ratio control would reduce the noise and speed up the search process significantly.

Fig. 4 shows a response in the BSFC signal to a step change in cam and spark timings. It can be seen that in about 4 s after the step change, the averaged BSFC approaches its final value.

C. ES Results

This subsection shows results produced by the software based on the algorithms described in Section II. The algorithms were tested at various combinations of engine speed and torque between 1000–2000 rpm and 16–100 N·m (high torques were avoided since the software did not have a built-in procedure to handle spark knock). The results shown correspond to 1500 rpm/62.5 N·m. Each figure brings the evolution of the parameter estimates, together with BSFC, which was filtered by the 4 s moving average filter.

Depending on the initial parameter values, the ES algorithms find one or the other local BSFC minimum:

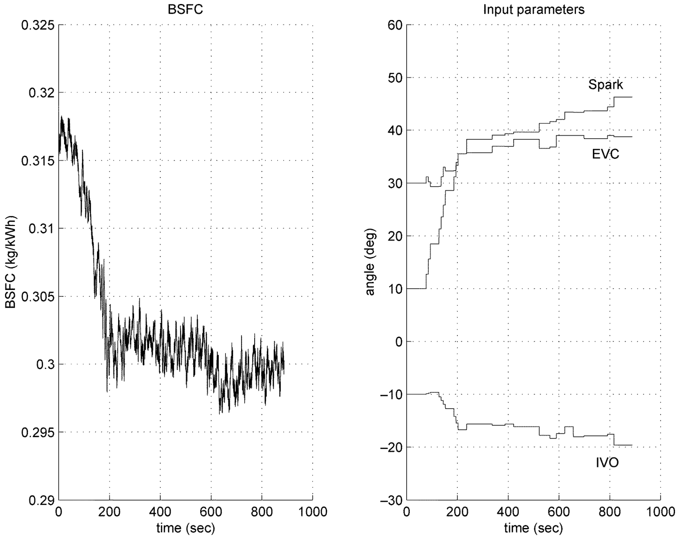


Fig. 9. Modified BWSD, 1500 rpm, 62.5 N·m. Initial values $(\mathrm{ivo}, \mathrm{evc}, \mathrm{spark}) = (-10, 10, 30)$.

- locating the first minimum $(\mathrm{ivo}, \mathrm{evc}) \approx (-20, 40)$: Figs. 5 and 6 for SPSA, Figs. 7 and 8 for PEFD, Fig. 9 for the modified BW steepest descent;
- locating the second minimum $(\mathrm{ivo}, \mathrm{evc}) \approx (30, 40)$, Fig. 10 for SPSA, and Fig. 11 for PEFD.

These figures illustrate also several important points about our algorithms.

- Due to its randomized selection of perturbation vectors, SPSA can produce different paths to the optimum. Hence, the ES times can vary (compare Figs. 5 and 6).
- The PEFD presented here chooses perturbation vectors in a periodic fashion and its results are more repeatable.
- In the experiments shown here, PEFD and SPSA were running with two different values of gain: *high* (1500) and *low* (750). Most of the time, the high gain was used. Running

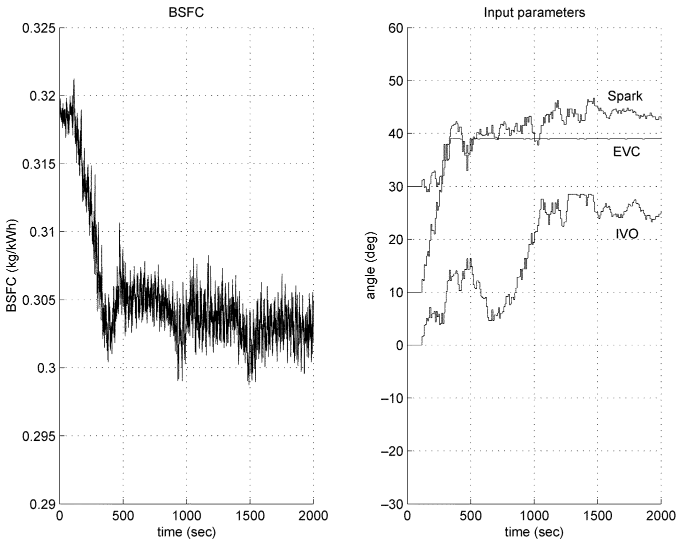


Fig. 10. SPSA, 1500 rpm, 62.5 N·m. Locating the second local minimum. Initial values $(\mathrm{ivo}, \mathrm{evc}, \mathrm{spark}) = (0, 10, 30)$. Notes: the algorithm gain was reduced at $t \approx 1500$.

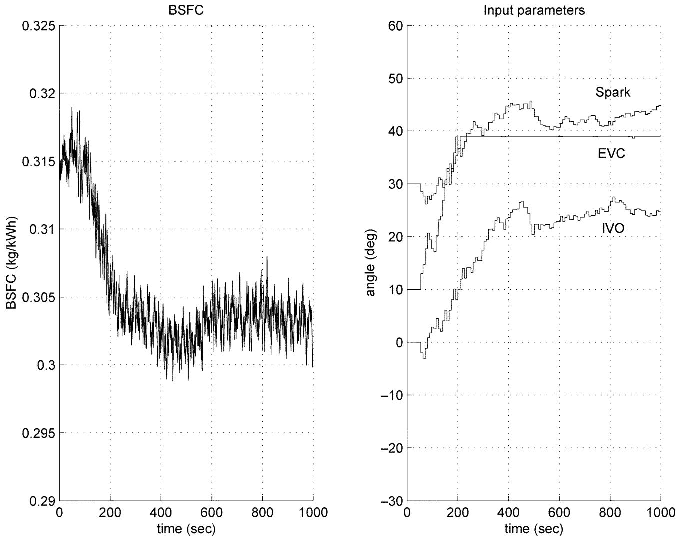


Fig. 11. PEFD, 1500 rpm, 62.5 N·m. Locating the second local minimum. Initial values $(\mathrm{ivo}, \mathrm{evc}, \mathrm{spark}) = (0, 10, 30)$.

the SPSA/PEFD with lower gain produced more robustness to noise and more accurate estimation, but also slower speed of convergence shown in Fig. 8. Our solution was to run the algorithms at high gain until the operator concludes that the estimates oscillate around the optimum; then the gain should be set to low. Using the interactive features of the software, the gain can be reduced online while the algorithm is searching for the optimum, (see Figs. 6, 7, and 10). Comparison in parameter space between runs at low and high gain is shown in Fig. 12.

- During ES, the estimates can approach or enter regions where combustion is unstable. These regions are known from mapping and are marked in Fig. 12.

Additional notes about our experiments are as follows.

- There were small day to day variations in BSFC and they can be seen in the runs from the same initial conditions.
- Figures that represent convergence of the parameters are showing only the estimates x_k, not the actual inputs to the system, which are estimates with superimposed perturbation signals, i.e., $x_k \pm \lambda v_k$ or $x_k \pm \lambda e_i$, needed to compute the finite differences. Some of the oscillations visible in the BSFC signal are the results of these perturbations.

D. Combustion Quality and ES for Best BSFC

At low output torques, optimal values of input parameters of the di-VCT engine may lie in a region with high valve overlap where combustion may not be sufficiently stable. Moreover, the estimates can enter the unstable regions, and the algorithm may start hunting for a local optimum among unstable points. This was more the case for the SPSA and PEFD, which tend to be more aggressive in their search, than for the Box–Wilson algorithm (SPSA and PEFD update their estimates without testing whether the updated estimate produces an improvement, the gain or step size parameter is prescribed in advance. On the other hand, improvement tests are a critical part of the Box–Wilson algorithm). In many cases poor combustion will result in higher BSFC and the algorithms are able to find their way out of the unstable region. However, in a few cases (typically at low torques), the best BSFC is achieved at parameter combinations with insufficient combustion stability.

We would like to force the algorithms to find the optimal BSFC among points with sufficiently stable combustion. One way to avoid unstable combustion is to pre-map parts of the engine operating envelope to find unstable combustion regions and then provide that information to the ES algorithms in some form of prescribed constraints. This, however, would contradict our concept, since we want ES that does not require pre-mapping of the engine.

The approach we tried, amounts to penalizing (artificially increasing) BSFC measurements that are sent to the algorithm if the combustion becomes unstable. Namely, the algorithm bases its search on the following cost function:

$$y(x) = \mathrm{BSFC}(x) + L(x) \tag{4.2}$$

where $L(\cdot)$ is some measure of combustion quality, or as we will use it, instability. This information is usually obtained from cylinder pressure statistics. Because the cylinder pressure and the quality of the combustion are directly reflected in the output torque, which we measured, we decided to use the latter.

Our instability measure was drawn from observing torque variations between two consecutive changes of input parameters. After each input change, some time for the transients to settle was allowed (usually 1 or 2 s) before torque observation started. Two averages of torque measurements were recorded: a long-term torque average taken over the whole observation period, and a short-term average taken over one or two strokes. At the end of the period, the smallest short-term average was divided by the long-term average. The obtained variable is well correlated with LNV values obtained from processing the cylinder pressure information. LNV is defined as the ratio of the lowest to highest values of the indicated mean effective pressure (IMEP) in the cylinder or engine over an observation period.

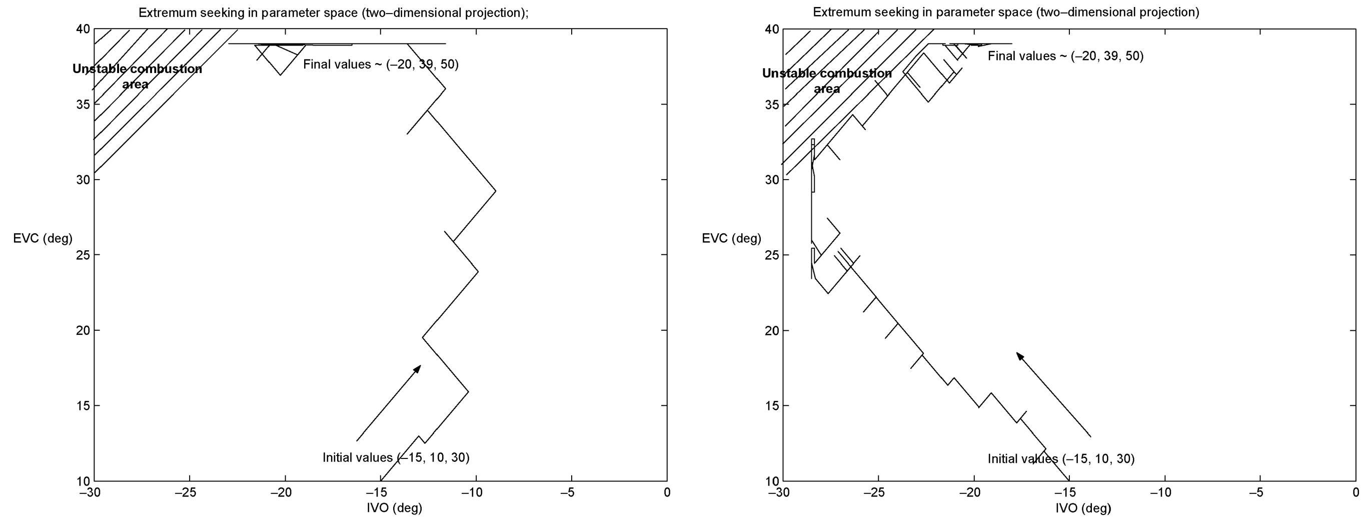


Fig. 12. Comparison in parameter space between two runs (cf. Figs. 8 and 7) of PEFD with high (left) and low (right) gain. The figure on the left represents the more aggressive run with higher gain at the start. The unstable combustion region, known from mapping, is shaded.

Thus, we will here call this variable $\mathrm{LNV}_t(x)$ to emphasize its nature and dependence on input parameters.

Our measure of instability for our cost function was formed as

$$L(x) = \gamma \left(L_{\min} - \mathrm{sat}_{-\infty}^{L_{\min}} \left[\mathrm{LNV}_t(x) \right] \right) \tag{4.3}$$

where γ and $L_{\min}$ are selected positive constants, and the saturation function is defined by

$$\mathrm{sat}_{-\infty}^{L_{\min}} \left[\mathrm{LNV}_t(x) \right] = \begin{cases} \mathrm{LNV}_t(x), & \mathrm{LNV}_t(x) <= L_{\min} \\ \mathrm{L}_{\min}, & \text{otherwise.} \end{cases} \tag{4.4}$$

Note that LNV_t can be negative if the engine misfires. Obviously, our measure of instability increases linearly as the combustion worsens and $\mathrm{LNV}_t(x)$ drops more and more below $L_{\min}$.

In Fig. 13 we show a result of a 62.5 N·m run of PEFD. The reader may compare this to Figs. 5–7. Note that the BSFC measurements are less jittery and that there are several degrees of difference in final spark end ivo values compared to no-LNV penalty runs, the estimates are pushed deeper into the stable region. However, we were still unable to produce successful results at low torques (16 N·m).

Note that in this situation, time scales between the original term, $\mathrm{BSFC}(x)$ and the penalty term, $\mathrm{LNV}_t(x)$, may not be comparable since they use different measurements. In some cases, the penalty term may require more time per measurement and thus it may slow down the convergence. In our case, the LNV was produced in the same time we spent on BSFC averaging. The overall result is that our LNV_t signal at low torques was louder than the BSFC. This seems to be the main reason why the approach did not succeed at low torque. We did not perform experiments with longer torque processing times. In addition, we recorded output torque from a high-speed sensor with a 1-kHz sampling rate, limited by the speed of our controller board. This rate may need to be increased for better results.

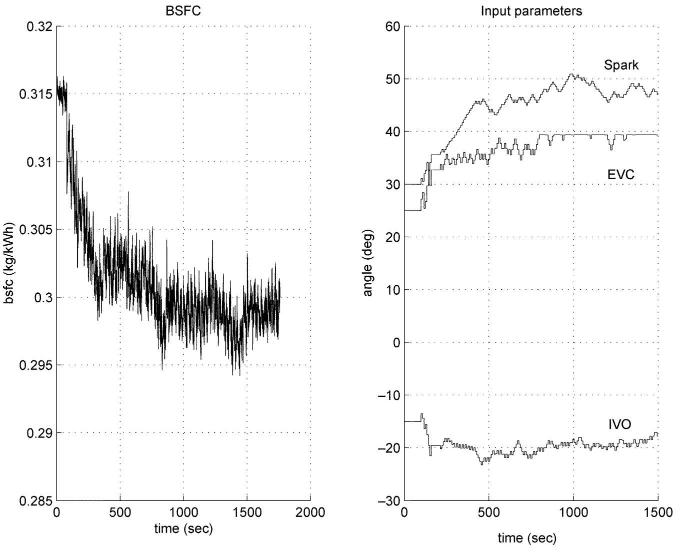


Fig. 13. PEFD, 1500 rpm, 62.5 N·m. LNV_t penalty added to BSFC. Initial values $(\mathrm{ivo}, \mathrm{evc}, \mathrm{spark}) = (-15, 25, 30)$. Compare with Figs. 5–7. Note the several degrees of difference in final spark end ivo values compared to no-LNV penalty runs, estimates are pushed into the stable region.

Finally we note that only very simple statistical processing of the torque information, a simple penalty function, and shorter measurement processing times were employed. Nevertheless, the results indicate that the problem can be successfully solved.

V. Conclusion

The increase in the number of tunable parameters, necessitates new approaches to mapping and calibration of modern automotive engines. One approach to finding optimal combinations of parameter settings is to apply ES algorithms. This paper shows an application of several ES algorithms to optimization of a dual-independent variable cam timing engine. The objective is to find input parameters (intake, exhaust, and spark timings) to minimize fuel consumption. We also performed BSFC optimization constrained by combustion quality. The experimental results demonstrate the feasibility of this approach. Additional work is needed to examine other possible performance criteria that include a penalty on NOx, CO, and HC emissions, etc.

References

[1] D. Popović, M. Janković, S. Magner, and A. Teel, "Extremum seeking methods for optimization of variable cam timing engine operation," in *Proc. Amer. Control Conf.*, Denver, CO, 2003, pp. 3136–3141.

[2] T. W. Asmus, "Perspectives on applications of variable valve timing," presented at the SAE Int. Congr., 1991, SAE Paper 910445.

[3] Y. Moriya, A. Watanabe, H. Uda, H. Kawamura, and M. Yoshiuka, "A Newly developed intelligent variable valve timing system—Continuously controlled cam phasing as applied to new 3 liter inline 6 engine," presented at the SAE Int. Congr., 1996, SAE Paper 960 579.

[4] R. Stein, K. Galietti, and T. Leone, "Dual equal VCT—A variable camshaft timing strategy for improved fuel economy and emissions," presented at the SAE Int. Congr., 1995, SAE paper 950 975.

[5] G. B. Meacham, "Variable Cam Timing as an Emission control Tool," presented at the Nat. West Coast Meeting, 1970, SAE Paper 700 645.

[6] J. H. Tuttle, "Controlling Engine Load by Means of Late Intake-Valve Closing," presented at the Passenger Car Meeting Conf., 1980, SAE paper 800 794.

[7] M. Jankovic and S. W. Magner, "Variable cam timing: Consequences to automotive engine control design," presented at the *Proc. 15th IFAC World Congr.*, Barcelona, Spain, Jul. 2002.

[8] ——, "Controlling complex automotive engine configurations for optimal performance, fuel economy, and emissions," in *Proc. Global Powertrain Congress*, Ann Arbor, MI, Sep. 2003, pp. 208–215.

[9] C. S. Draper and Y. T. Li, "Principles of optimalizing control systems and an application to the internal combustion engine," ASME Publications, Sep. 1951.

[10] P. Scotson and P. E. Wellstead, "Self-tuning optimization of spark ignition automotive engines," *IEEE Control Syst. Mag.*, vol. 3, no. 10, pp. 94–101, Apr. 1990.

[11] R. E. Dorey and G. Stuart, "Self-tuning control applied to the in-vehicle calibration of a spark ignition engine," in *Proc. IEEE Int. Conf. Control Appl.*, Glasgow, U.K., 1994, pp. 121–126.

[12] J. Bredenbeck, "Statistical design of experiments for on-line optimization of internal-combustion engines (in German)," *MTZ Motortech. Z.*, vol. 60, no. 11, pp. 740–744, 1999.

[13] S. P. Edwards, D. M. Grove, and H. P. Wynn, Eds., *Statistics for Engine Optimization*. London, U.K.: Professional Engineering, 1999.

[14] A. R. Teel and D. I. Popovic, "Solving smooth and nonsmooth extremum seeking problems by methods of nonlinear programming," in *Proc. Amer. Control Conf.*, vol. 3, Washington, DC, 2001, pp. 2394–2399.

[15] J. C. Spall, "Multivariate stochastic approximation methods," *IEEE Trans. Autom. Control*, vol. 37, no. 3, pp. 332–341, Mar. 1992.

[16] A. R. Teel, "Lyapunov methods in nonsmooth optimization, part II: Persistently exciting finite differences," in *Proc. 39th Conf. Decision Contr.*, Sydney, Australia, 2000, pp. 118–123.

[17] G. E. P. Box and K. B. Wilson, "On the experimental attainment of optimum conditions," *J. Roy. Stat. Soc. B*, vol. 13, no. 1, pp. 1–38, 1951.

[18] J. Spall, *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Hoboken, NJ: Wiley, 2003.

[19] H. J. Kushner and G. G. Yin, *Stochastic Approximation Algorithms and Applications*. New York: Springer-Verlag, 1997.

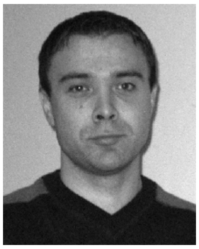
[20] J. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 8, no. 7, pp. 308–313, 1965.

[21] M. Krstic and H. H. Wang, "Stability of extremum seeking feedback for general dynamic systems," *Automatica*, vol. 36, no. 4, pp. 595–601, 2000.

[22] P. Sadegh, "Constrained optimization via stochastic approximation with a simultaneous perturbation gradient approximation," *Automatica*, vol. 33, no. 5, pp. 889–892, 1997.

[23] D. Popovic, A. Teel, and M. Jankovic, "An improved SPSA algorithm for stochastic problems with bound constraints," in *Proc. 16th IFAC World Cong.*, Prague, Czech Republic, Jul. 2005.

[24] R. Oldenburger, Ed., *Optimal and Self-Optimizing Control*: MIT Press, 1966.



Dobrivoje Popović received the B.Sc. degree from the University of Belgrade, Belgrade, Yugoslavia, in 1998, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California, Santa Barbara, in 2000 and 2004, respectively.

He held a postdoctoral research position with the University of California, Santa Barbara. He was also a Senior Research Engineer with the United Technologies Research Center, East Hartford, CT. In 2005, he joined Merrill Lynch Global Equity Trading, New York, NY, where he is currently working on portfolio optimization problems.

Mrdjan Janković (F'04) received the B.S. degree from the University of Belgrade, Belgrade, Yugoslavia, in 1986, and the M.S. and Ph.D. degrees from Washington University, St. Louis, MO, in 1989 and 1992, respectively.

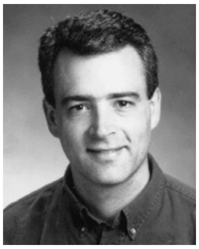
He held postdoctoral teaching and research positions with Washington University and the University of California, Santa Barbara. He joined Ford Research Laboratory, Dearborn, MI, in 1995, where he is currently a Technical Leader in the Powertrain Controls Department. His research interests include automotive engine control, nonlinear control, and time-delay systems. He has coauthored one book *Constructive Nonlinear Control* (Springer-Verlag, 1997) and holds over 20 U.S. patents.

Dr. Janković received the Ford Research Technical Achievement Award and the IEEE Control Systems Society Outstanding Paper Award. He has served on the Editorial Board of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY since 1997.

Steve Magner received the M.S. degree in electronics and computer control systems from Wayne State University, Detroit, MI, in 1990, and the M.S. degree in electrical engineering from the University of Michigan, Lansing, in 1995.

He joined Ford Motor Company, Dearborn, MI, in 1986. Currently, he is a Technical Specialist in Powertrain Research, focusing on software development and control systems in variable valve trains. He has coauthored several papers in powertrain controls and mechatronic systems and has 11 patents.

Mr. Magner received the Ford Research Technical Achievement Award in 2001.



Andrew R. Teel (F'02) received the A.B. degree in engineering sciences from Dartmouth College, Hanover, NH, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1989 and 1992, respectively.

After receiving the Ph.D. degree, he was a postdoctoral fellow at the Ecole des Mines de Paris, Fontainebleau, France. In September of 1992, he joined the Faculty of the Electrical Engineering Department at the University of Minnesota, St. Paul, where he was an Assistant Professor until September of 1997. In 1997, he joined the Faculty of the Electrical and Computer Engineering Department at the University of California, Santa Barbara, where he is currently a Professor.

Prof. Teel has received NSF Research Initiation and CAREER Awards, the 1998 IEEE Leon K. Kirchmayer Prize Paper Award, the 1998 George S. Axelby Outstanding Paper Award, and was the recipient of the first SIAM Control and Systems Theory Prize in 1998. He was also the recipient of the 1999 Donald P. Eckman Award and the 2001 O. Hugo Schuck Best Paper Award, both given by the American Automatic Control Council.