Optimization and scheduling for automotive powertrains

M. Jankovic, S. Magner
Ford Research & Advanced Engineering
2101 Village Road, MD 2036 SRL
Dearborn, MI 48121-2053, USA

Abstract

Addition of devices intended to improve performance, fuel economy, and emissions of automotive engines makes the tasks of finding optimal settings for the engine parameters and implement the optimal schedules in the engine control unit more difficult. In this paper we present an integrated approach intended to simplify these tasks. The components of the integrated process are 1) engine parameter separation into constrained, dynamic, and instantaneous, 2) extremum seeking, and 3) inverse distance interpolation. The approach is illustrated on the problem of optimizing fuel consumption of a dual-independent variable cam timing engine.

1. Introduction

Modern automotive engine control systems have become increasingly complex in order to handle additional devices or operating modes, such as exhaust gas recirculation (EGR), variable valve timing or lift, intake manifold tuning valve, lean air-fuel ratio operation, etc. These devices are introduced to meet stringent emission standards and progressively lower fuel consumption requirements. The complexity has been exemplified by a proliferation of calibration look-up tables needed to schedule engine variables for optimal performance, emissions, or fuel economy, or govern interactions between them. The look-up tables and functions are obtained after a complex mapping, data regression, and optimization process followed by in-vehicle calibration to fine-tune the powertrain behavior. The complexity increases not linearly, but exponentially with a number of degrees of freedom. An additional degree of freedom typically increases the mapping time and the size of the calibration tables by a factor between 2 (for two position devices) and 3-7 (for continuously variable devices). To manage the complexity, several different approaches have been pursued. Automated calibration/optimization based on the design of experiments (DoE) methods is described in several papers in the collection edited by Edwards et al [4]. Neural network based methods and tools that extract information for engine modeling and optimization from dynamic tests is proposed in Hafner et al [6].

This paper describes an alternative optimization/scheduling process that generates calibration tables in a way that reduces time needed to collect data in the dynamometer cell. In addition, it reduces the size and dimension of the tables and functions implementing the schedules in the engine control unit (ECU) and simplifies the calibration effort. Our approach is based on combining three distinct technologies: (a) extremum seeking (ES) (also called direct search or steepest accent/descent) methods that are developed from the original works of Box and Wilson [1], and Kiefer and Wolfowitz [9] (automotive applications of ES are reported in Dorey and Stuart [2] and Draper and Li [3]), (b) design of experiments (DoE) methods [5, 11], and (c) inverse-distance control strategy for ECU implementation with sparse data sets proposed by Jankovic and Magner [7].

The process to generate ECU calibration assumes that an engine is available in a dynamometer test cell and consists of the following steps:

1. Split engine variables into three groups: constrained variables, instantaneous optimization parameters, and dynamic optimization parameters.
2. Cover the operating envelope of the constrained variables by a grid. For each grid intersection (node) find the optimal combination of the optimization parameters. We propose that this step employs an extremum seeking method because of its efficiency.
3. In the space of dynamic optimization parameters, generate a scatter plot of the parameter combinations that are optimal at constrained variable nodes. Based on this, generate feature points, lines, curves, planes, etc, such that these features contain, or are very close to the optimal points. Using conventional or DoE methods, map the engine to obtain optimal values of the instantaneous parameter variables as functions of constrained and dynamic parameters on the features.
4. In the engine control unit (ECU) strategy schedule dynamic optimization parameters as functions of actual or desired values of the constrained variables.
5. Implement an interpolation system that schedules the instantaneous parameters as functions of current values of the constrained variables and dynamic parameters. An interpolation method that we propose is a combination of the conventional bi-linear interpolation over the grid of constrained variables and inverse distance interpolation method over the features in the dynamic parameter space.
6. Implement compensation systems to mitigate potential undesirable effects of interactions between dynamic parameters during transients (this issue will not be discussed further in this paper; an example of such a system can be found in [8]).

The paper is organized as follows. Steps 1 through 5 are described in more detail in Section 2. An example of the complete process including two approaches to parameter interpolation for a dual-independent VCT engine is given in

0-7803-8335-4/04/$17.00 ©2004 AACC

Section 3. Section 4 contains experimental results with the comparison of the scheduling accuracy of these methods to an "ideal" full-factorial scheduling approach used as a benchmark.

2. Optimization and scheduling

2.1 Classification of engine variables

The first step is to separate engine variables that can be used for optimization from those that are constrained. For example, in most applications, engine speed and torque are constrained variables because they are determined by outside factors such as vehicle speed, transmission gear, and driver's acceleration demand. The EGR rate, spark timing, and/or cam timing can be used to optimize the engine operation for fuel economy, emissions, and/or performance, and are considered optimization parameters.

To relate engine mapping/optimization with subsequent scheduling and interpolation in the ECU during in-vehicle operation, we further subdivide the optimization parameters into instantaneous and dynamic. The instantaneous parameters include those that can be changed (almost) instantaneously relative to the duration of an engine cycle. Such variables may include spark timing and air-fuel ratio. The dynamic optimization parameters are those that may take several engine cycles to assume their desired (optimal) setting. Examples of such variables are EGR rate and (hydraulically actuated) variable cam timing. It may not be clear to which category some variables belong. Often, it is the relative priority of the action performed by the actuator that determines if the variable is instantaneous or dynamic. For example, maintaining air-fuel ratio at stoichiometry is a high priority task in spark ignition internal combustion engines and, thus, air-fuel ratio is considered an instantaneous variable. The air-fuel ratio in a diesel engine is not so tightly controlled (that is, it is controlled by the air-supply system rather than the fuel injectors) and can be considered a dynamic variable. In summary, the set of engine variables is divided into

1. Constrained variables (e.g. engine speed, torque)
2. Optimization parameters, that are further divided into
    a. Instantaneous optimization parameters (e.g. spark)
    b. Dynamic optimization parameters (e.g. EGR, cam timing).

Next, the operating envelope of the constrained variables is covered by a grid corresponding to selected break points. An intersection of the grid lines defines a point (node) where we want to run an optimization experiment. In the case of engine speed and torque, a sketch of the operating envelope and the nodes are shown in Figure 1. Instead of the "full-factorial" optimization over the grid, one can actually set up a central composite design (CCD) or a D-Optimal, V-Optimal, etc. design and get the optimal points on a grid from regressing the data (see Montgomery [11]).

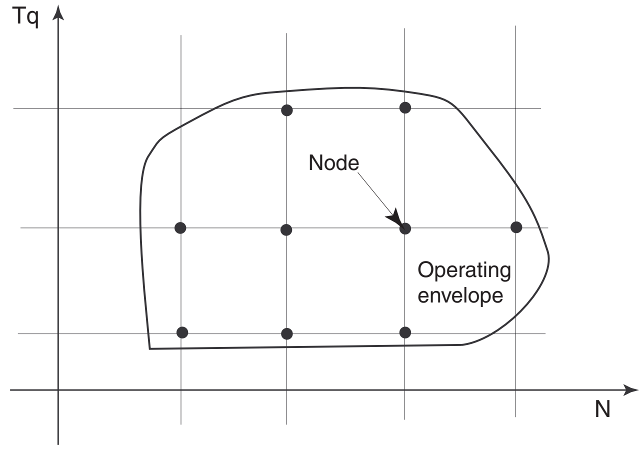

Figure 1. Operating envelope in the speed-torque plane for an internal combustion engine.

2.2 Optimization

In a standard dynamometer test cell set-up the engine speed is typically controlled by the electric motor/generator, while the engine torque is controlled by an electronically controlled throttle (ETC). The torque can also be controlled by either fuel (in lean burn and diesel engines), valve lift (variable valve lift engines) or intake event duration (camless engines) in which case it should be used instead of the ETC. The engine variable that is used to control the torque (for example, throttle position) is constrained and cannot be used as an optimization parameter.

At each speed-torque node, an extremum seeking experiment is conducted to find the optimal combination of the remaining (free) engine variables (optimization parameters) that minimizes a cost function. For example, a choice of the cost is a weighted average, with w_i denoting the weight factors, of the brake specific fuel consumption (BSFC), combustion stability measure such as coefficient of variance of IMEP, and/or hydrocarbons (HC), nitrogen oxides (NOx), and carbon monoxide (CO) emissions:

$$J = w_1\, BSFC + w_2\, HC + w_3\, NOx + w_4\, CO + w_5\, \mathrm{cov}\, IMEP$$

A number of extremum seeking (direct search) methods, described in the literature, can be applied to find the optimal combination of optimization parameters that minimize the cost *J*. These methods include stochastic approximation algorithms [13], Box and Wilson steepest descent [1], persistently exciting finite differences (PEFD) algorithm [14], the method of sinusoidal perturbation [10], etc. What they have in common is that, starting from an initial estimate of the optimum, they vary the optimization parameters and improve the estimate based on the measurements of the variables that make up the cost function. Sometimes, to find a global optimum, the selected extremum seeking method would have to be initialized from several different initial estimates. Engine designer a-priori knowledge can be used as a guide to select an appropriate initial parameter values to reduce the search time and the number of trials.

2.3 Scheduling

Once the optimal combinations are available, the next step is to develop schedules for the optimization parameters. The scheduling of instantaneous and dynamic parameters is dealt with differently. Dynamic parameters are scheduled on constrained variables, while instantaneous parameters are scheduled on both constrained variables and dynamic parameters. The scheduling of instantaneous parameters is often simplified if we introduce a scheduling variable as a proxy for the actual engine torque. In SI engines it is typically engine air-charge or load (normalized air-charge). In diesel engines it can be the amount of fuel injected. By doing this we avoid circular logic in which, for example, spark timing is scheduled on engine torque and torque depends on spark timing.

The ES methods find the optimal parameter values directly without generating more complete engine map such as the one obtained by conventional engine mapping or even DoE. This does not create a problem if all the optimization parameters were instantaneous. At each speed-torque node, we can schedule the optimal set of parameters determined by ES, with each instantaneous parameter a function of speed and torque (or air-charge/load). Between the node points, we can interpolate using the conventional bi-linear (assuming we have only two constrained variables, speed and torque) interpolation that applies to data on a rectangular grid. With sufficiently many break points in speed and torque, a high degree of interpolation accuracy can be achieved. Because the parameters are "instantaneous" they reach their scheduled values sufficiently fast so that no coordination or control action is needed to compensate for their transients.

2.4 Interpolation

An optimization parameter controlled by an actuator that, due to a physical constraint, has a delay or a significant transient time, has to be consider a dynamic parameter. In such a situation any instantaneous parameter is scheduled on speed, torque/load, and on the present value of the dynamic parameter. A simple example is an engine with EGR where the spark timing (which is an instantaneous parameter) is scheduled not only on engine speed and load (a proxy for torque, the constrained variables), but also on EGR (a dynamic optimization parameter).

To implement the above described scheduling one can map the engine and fill the required look-up tables that are then evaluated on-line by the control strategy. By using the results of ES optimization, we can avoid generating a complete engine map, which is a time consuming task. Rather, we propose to map a small fraction of the optimization parameter space (typically a lower dimensional one) and use an interpolation method to achieve good accuracy in transients. Such methods include the inverse distance method described in Jankovic and Magner [7] and its extension proposed in this paper.

3. Example: dual-independent VCT engine

To illustrate the process of mapping, optimization, variable scheduling, and interpolation described above, we consider the problem of optimizing fuel economy of an engine in which the intake and exhaust cam timings can be varied independently (hence dual-independent VCT) while the intake and exhaust event durations are fixed. In this configuration the optimization variables are the intake cam timing represented by the intake valve opening (*ivo*) variable, exhaust cam timing represented by the exhaust valve closing (*evc*) variable, and spark timing (*spark*). In our experimental 3.0L di-VCT engine *ivo* and *evc* are constrained by hardware to $-30 \le ivo \le 30$ and $0 \le evc \le 40$. Hydraulic VCT actuators that are typically used vary the cam timing are relatively slow, sometimes taking 500 ms or more to reach the commanded position. Hence, we classify the variables into

- Constrained variables – speed (*N*) and torque (*tq*).
- Dynamic optimization parameters – intake and exhaust cam timings represented by (*ivo*) and (*evc*)
- Instantaneous optimization parameters – spark timing (*spark*).

3.1 Extremum seeking for di-VCT engine

To find the optimal parameter combinations at each speed-torque node we have successfully tried several ES methods including the stochastic approximation [13], persistently exciting finite differences [14], and a modified version of the Box-Wilson method [1]. The controllers implementing the algorithms have been tested on the 3.0L di-VCT engine in a dynamometer test cell. During the optimization runs the torque and the engine speed were kept constant at the predefined (grid node) values. The optimization parameters (*ivo, evc,* and *spark*), were varied in a search of a combination that provides the best BSFC, possibly subject to a penalty on combustion stability. That is, in this case the minimization criterion used is

$$J = w_1\, BSFC + w_2\, \mathrm{cov}IMEP$$

The algorithms tested found the optimal *ivo, evc,* and *spark* combination in about 15 to 20 minutes. A typical run of an ES controller is shown in Figure 2. Additional details about the algorithms applied, the experimental setup, and the results can be found in [12].

The procedure is repeated at each node point in the operating envelope. In this way we obtained a BSFC optimal schedule of intake, exhaust, and spark timings (denoted here by "SL" for (combustion) stability limited) as functions of the constrained variables:

$$[ivo\_sl, evc\_sl, spark\_sl]^T = F_{SL}(N, tq)$$

Now, the dynamic optimization parameters *ivo* and *evc* are scheduled on the engine speed and desired torque (while the ETC is used to achieve the actual torque equal to the desired). The spark is scheduled on engine speed, load, and actual *ivo* and *evc* to assure accurate spark value during cam timing transients.

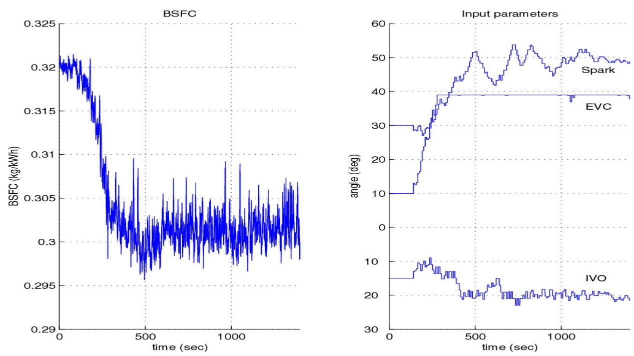

Figure 2: ES controller varies the optimization parameters (right plot) to minimize BSFC (left) at 1500 RPM, 2.62 bar BMEP.

3.2 Multi-point inverse distance

The immediate problem with scheduling spark on actual (current) IVO and EVC is that at this stage only the optimal, i.e. maximum brake torque (MBT), spark value at the optimal (SL) IVO and EVC is known. When the cam timing values are in transition the *spark_sl* value is not appropriate and may differ from the MBT value by as much as 15 to 20 degrees.

Coincidentally, there is a need to have the engine mapped at a few additional cam timing combinations to be able to handle other operating modes/conditions such as extreme oil temperatures and altitude that require use of cam timing and spark schedules different from the SL ones. For cold oil temperature, low oil pressure, etc., the VCT actuators may remain in their "default" (D) position that is usually chosen to provide stable operation. The default position is fixed by hardware design and so the corresponding variables *ivo_d* and *evc_d* are constant. The corresponding spark timing *spark_d* = $F_D(N,load)$ can be obtained by conventional engine mapping. At altitude, limited air density may prevent achieving a prescribed torque at the "best BSFC at sea level" combination of IVO and EVC. Thus, to achieve the desired torque at altitude we use another set of cam schedules called "optimal performance" (OP) which is, in general, different from the sea level tables for a given speed and torque and depend only on engine speed.

$$[ivo\_op, evc\_op]^T = Fn_{OP}(N)$$

On the other hand, the MBT spark at OP cam timing depends on engine speed as well as on load:

$$spark\_op = Fn\_spark_{OP}(N, load)$$

Thus, for each speed-load (or speed-torque) point we can find three (*ivo*, *evc)* pairs from one of the tables *ivo_sl, evc_sl; ivo_op*, *evc_op;* or by using constant *ivo_d, evc_d.* Which of the three is commanded to the VCT actuators depends on the current operating condition (SL (nominal), OP (altitude), or D (default)). Regardless of the mode, the information stored in these tables can be used for spark interpolation. Next we describe a method that interpolates the spark values using the three "points" available.

In [7] the following inverse distance interpolation scheme has been proposed. Given the current engine speed *N*, *load*, *ivo*, and *evc*, we compute the spark timing as follows:

- For current values of speed and load, determine from the cam look-up tables the values of *ivo_sl, ivo_op, evc_sl, evc_op (ivo_d, evc_d* are fixed).
- For current speed and load, determine from the spark look-up tables the values of *spark_sl, spark_op, spark_d*.
- Compute the square "distances" between the above cam timing node points (SL, OP, and D) and the current cam timing position in the IVO-EVC plane offset by the small constant ε (used to prevent division by 0 and selected to be of the order of noise that affects cam timing):

$$d_1 = (ivo - ivo\_sl)^2 + (evc - evc\_sl)^2 + \varepsilon$$

$$d_2 = (ivo - ivo\_op)^2 + (evc - evc\_op)^2 + \varepsilon$$

$$d_3 = (ivo - ivo\_d)^2 + (evc - evc\_d)^2 + \varepsilon$$

- Compute the spark timing as a weighted average of the spark at the three modes, with weights inversely proportional to the squared distances:

$$spark = \frac{\frac{1}{d_1} spark\_sl + \frac{1}{d_2} spark\_op + \frac{1}{d_3} spark\_d}{\frac{1}{d_1} + \frac{1}{d_2} + \frac{1}{d_3}}$$

- The above four steps describe the interpolation in the IVO-EVC direction. Standard bilinear interpolation in the speed-load direction can be used as those points are available in a rectangular grid.

Figure 3 provides a graphical illustration of the look-up tables used by the interpolation method. As discussed above, the default point is fixed by hardware, so *ivo_d* and *evc_d* show no speed/load dependence. OP point, defined by best wide-open-throttle (WOT) torque, depends on engine speed only. Finally, SL point shows full speed, load dependence. Note that only one of these functions/tables is used for scheduling cam timing depending on the operating mode, but for spark interpolation all three pairs are determined and the distance from the current cam timing point computed. Corresponding to each of the SL, OP, or D cam timings, there is a value of spark timing. Finally, the inverse distance interpolation is used to find the spark value at the current (*ivo, evc*) point.

For greater accuracy, additional points can augment the three modal points. For experiments shown in Section 4 we have added a point placed in the middle of the IVO-EVC rectangle, hence, called intermediate point (IM).

3.3 Line feature interpolation

The multi-point inverse distance works well with sparse data, but becomes cumbersome as the number of points increases which may be required for higher accuracy. In

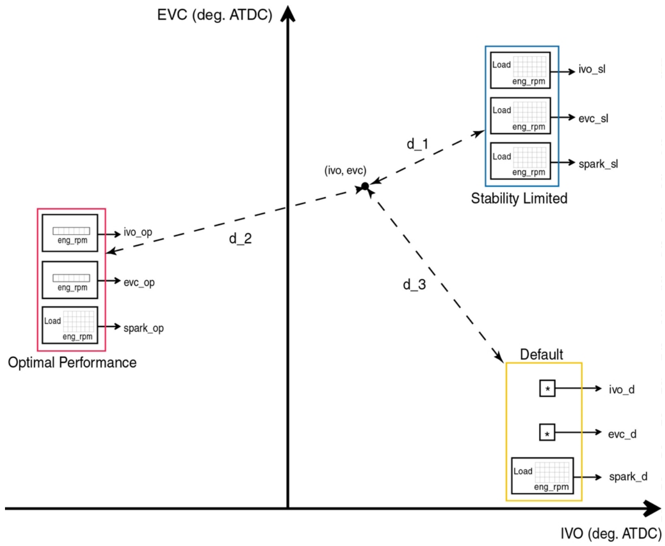

Figure 3: Interpolation with three node points SL, OP, and D in the IVO-EVC plane.

such a situation it makes sense to structure the points into higher dimensional "features." The first step, done at the design stage, is to decide on the features in the IVO-EVC plane. In this particular case the features will be line segments, but they can be curves or, in higher dimensional cases, planes/surfaces. Typically, the optimal IVO-EVC pairs, which can be determined by running ES experiments, change with the engine speed and torque. When these optimal pairs are determined, plotting them creates a scatter plot of the optimal (SL) IVO, EVC parameters. By connecting some (preferably most) of the points we generate line segments in the plane. In general, feature lines depend on the ranges of IVO and EVC determined by hardware design.

Let us associate *ivo* variable with the *x*-axis and *evc* variable with the *y*-axis. Each line segment is parameterized by a parameter, let us call it *s*, so that their expressions take the following form:

$$x = a_x\, s + b_x$$

$$y = a_y\, s + b_y$$

where s takes on values between s_{min} and s_{max}. Next, we map the engine at the feature line segments. Because the resulting mapping space is "rectangular" (in speed, load, and s-parameter) we can use the standard DoE methods for mapping and the conventional multi-linear interpolation in the ECU control strategy for interpolation:

$$spk\_line = Fn\_lookup(N, load, s)$$

Instead of computing distances to feature points, we compute the distance to the line feature (that contains the points) from the current (*ivo*, *evc)* pair, denoted here by (x_0, y_0), to the closest point on the line as shown in Figure 4. The value of the parameter s of the closest point to (x_0, y_0) can be found by minimizing the distance

$$d\_line(x_0, y_0, s) = (x_0 - a_x s - b_x)^2 + (y_0 - a_y s - b_y)^2$$

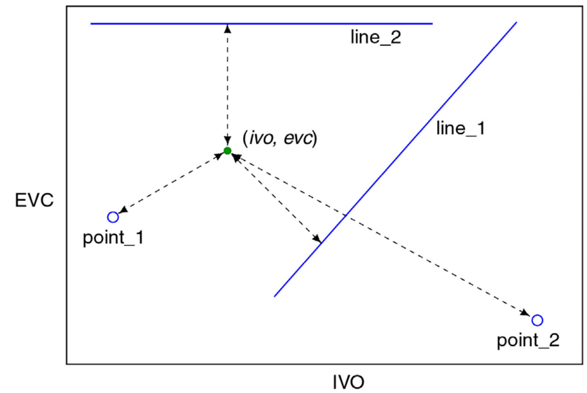

Figure 4. Interpolation based on distances to line segment and point features.

The minimum must satisfy

$$\frac{\partial\, d\_line}{\partial s} = 2(x_0 - a_x s - b_x)a_x + 2(y_0 - a_y s - b_y)a_y = 0$$

Solving the above equation for s we obtain the value of this parameter that minimizes the distance between the point (x_0, y_0) and the line:

$$s^* = \frac{a_x x_0 + a_y y_0 - a_x b_x - a_y b_y}{a_x^2 + a_y^2}$$

If s^* does not belong to the interval between s_{min} and s_{max}, that is if the projection of the point onto the line does not belong to the segment, we set s^* equal to the upper or lower limit as appropriate:

$$\text{if} \quad s^* > s_{\max}, \quad s^* = s_{\max}$$

$$\text{else if} \quad s^* < s_{\min}, \quad s^* = s_{\min}$$

The minimal square distance to the line, denoted by d_line^* is found by substituting s^* into the expression for *d_line*:

$$d\_line = (x_0 - a_x s^* - b_x)^2 + (y_0 - a_y s^* - b_y)^2 + \varepsilon$$

The spark value on the line is obtained by setting s = s^* into the conventional function look-up with three inputs:

$$spk\_line\_j = Fn\_lookup(N, load, s_j^*)$$

The square distance to a feature point (x_p, y_p) is found by using the conventional expression

$$d\_point = (x_0 - x_p)^2 + (y_0 - y_p)^2 + \varepsilon$$

Given the present values of engine speed *(N),* load *(load),* intake cam timing *(ivo),* and exhaust cam timing *(evc),* for each feature point or line segment we compute the distance between the feature (point or line) and the current *(ivo, evc)* value and the spark value on this feature. Finally, the spark timing is found by the inverse distance interpolation:

$$spark = \frac{\displaystyle\sum_{i=1}^{m} \frac{spk\_point\_i(N, load)}{d\_point\_i(ivo, evc)} + \sum_{j=1}^{n} \frac{spk\_line\_j(N, load, s_j^*)}{d\_line_j^*(ivo, evc)}}{\displaystyle\sum_{i=1}^{m} \frac{1}{d\_point\_i(ivo, evc)} + \sum_{j=1}^{n} \frac{1}{d\_line_j^*(ivo, evc)}}$$

4. Experimental Verification

The spark interpolation described in Section 3 has been evaluated experimentally on the 3.0L di-VCT engine. Figure 5 shows the traces of engine torque measured by an in-line torque sensor in the dynamometer test cell (top plot), actual traces of cam timing as represented by *ivo* and *evc* variables (middle plot), and MBT spark timing computed by several different methods (bottom plot), for the three different torque steps. The engine speed was held constant at 1500 RPM by the dynamometer.

The solid blue line in the bottom plots gives the MBT spark timing computed by the full-factorial 4-dimensional look-up table, wherein the spark values are available at the IVO-EVC grid with 10 degrees increment. The full-factorial spark timing is assumed to be the most accurate because it relies on the most detailed data and is used here as a benchmark. It has not been considered for implementation because it requires the most detailed engine mapping and uses the largest amount of computer memory. For example, as implemented, it uses about 3.5 times more memory than the 2-line method and about 9 times more memory than the 4-point method.

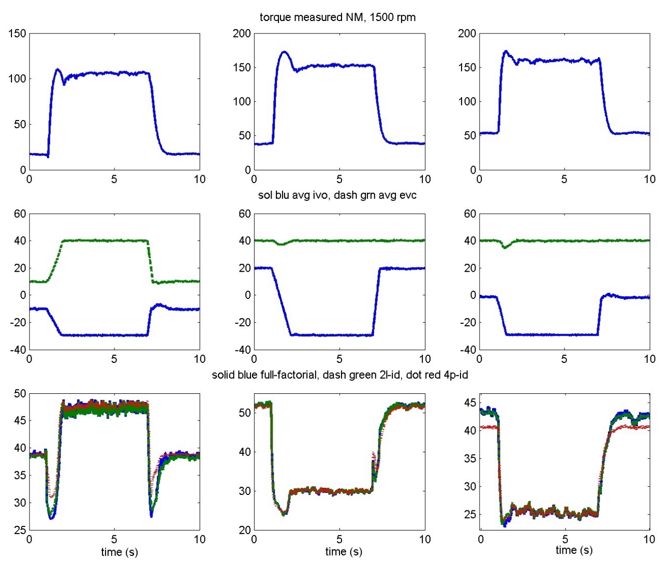

Figure 5. Experimental traces of engine torque (top), intake and exhaust cam timing (middle) and spark timing computed by different methods (bottom).

The dash green line in the bottom plots in Figure 6 is the spark timing computed by 2-line inverse distance method. In these runs it is almost indistinguishable from the full-factorial value. The dash red line shows the spark timing computed by the 4-point method. The accuracy is very good, in particular considering the significant reduction in the number of data points used. Even though the 4-point method is designed to achieve high steady-state accuracy, one can see that in the last test the error remains during steady-state operation. The problem is that the scheduled torque is not equal to the actual one so the SL cam timing, which is correlated with the scheduled torque, may not be in synch with the SL spark, which is correlated to the actual load (torque), if the actual torque is not equal to the desired. Note that the 2-line method is not sensitive to this problem because the lines connect scheduled cam timing points for different torque values and the actual point is likely to fall on a line even in the presence of torque inaccuracies.

References

[1] G.E.P. Box and K.B. Wilson, "On the Experimental Attainment of Optimum Conditions," *Journal of the Royal Statistical Society, Series B*, vol. 13, 1-38, 1951.

[2] R.E. Dorey, G. Stuart, "Self-tuning control applied to the in-vehicle calibration of a spark ignition engine," *Proc. of IEEE CCA*, pp.121-125, Glasgow, UK, 1994.

[3] C.S. Draper, Y.T. Li, "Principles of optimalizing control systems and an application to the internal combustion engine," *ASME Publications*, 415-429, 1954.

[4] S.P. Edwards D.M. Grove, H.P. Wynn editors., *Statistics for Engine Optimization*, Professional Engineering Publishing Ltd., London, 1999.

[5] D.M. Grove, T.P. Davis, *Engineering quality and experimental design*, John Wiley & Sons, NY, 1992.

[6] M. Hafner, M. Weber, R. Isermann, “Model-Based Control Design for IC-ENgines on Dynamometers: The Toolbox OPTIMOT,” *Proceedings of 15-th IFAC Congress*, Barcelona, Spain, July 2002.

[7] M. Jankovic, S. Magner, “Controlling Complex Automotive Engine Configurations for Optimal Performance, Fuel Economy, and Emissions,” *Proceedings of Global Powertrain Congress*, Ann Arbor, MI, Sep. 2003.

[8] M. Jankovic, S. Magner, S. Hsieh, J. Koncsol, “Transient effects and torque control of engines with variable cam timing,” *Proceedings of the American Control Conference*, Chicago IL, June 2000.

[9] J. Kiefer and J. Wolfowitz, "Stochastic Estimation of the Maximum of a Regression Function," *Ann. Math. Stat.,* vol. 23, pp. 462-466, 1952.

[10] M. Krstic, H.H. Wang, "Stability of extremum seeking feedback for general nonlinear dynamic systems," *Automatica*, vol. 36, pp. 595-601, 2000.

[11] D.C. Montgomery, *Design and Analysis of Experiments*, (5^{th} ed.), John Wiley & Sons, NY, 2001.

[12] D. Popovic, M. Jankovic, S. Magner, A. Teel, “Extremum seeking methods for optimization of variable cam timing engine operation,” *Proceedings of the American Control Conference*, Denver, June 2003.

[13] J.C. Spall, "Multivariable Stochastic Approximation Methods," IEEE Trans. On Automatic Control, vol. 37, pp. 332-341, 1992.

[14] A. Teel, “Lyapunov methods in non-smooth optimization, Part II: persistently exciting finite differences," Proc. of 39-th IEEE CDC, Sydney, Australia, Dec. 2000.