MegaSquirt VE Table Determination using Oxygen Sensor Feedback
Bowling/Grippo

The MegaSquirt EFI computer uses a 64-element Volumetric Efficiency (VE) table, organized on an 8x8 table structure, for determining fuel requirements based on Manifold Absolute Pressure (MAP) and RPM. The EFI computer performs a 2-D linear interpolation based on the current RPM and MAP readings, and yields a VE value used in pulsewidth calculations. Determining grid points values can be performed using Oxygen Sensor feedback control values. This document describes a method to accomplish this.

Lets start with the control equations for MegaSquirt:

$$REQ\_FUEL = \frac{36{,}000{,}000 * CID * AIRDEN(100{,}70)}{NCYL * 14.7 * INJFLOW} * \frac{1}{PULSE\_DIVIDE}$$

REQ_FUEL = Computed injector open time in tenths of millisecond.
CID = Cubic Inch Displacement.
AIRDEN = Air density (pounds per cubic feet) at MAP pressure of 100 Kpa, Air Temperature of 70 Degrees F, and Barometric Pressure of 30.00 In HG
NCYL = Number of Cylinders
INJFLOW = Injector Flow Rate in pounds per hour.
PULSE_DIVIDE = injection divide number for number of injections per engine cycle.

The **AIRDEN** equation (used above) is defined:

$$AIRDEN(MAP, Temp) = \frac{0.0391568 * (MAP * 10 - 31.0)}{(Temp + 459.7) * 1728}$$

MAP = Manifold Air Pressure in Kpa
Temp = Air Temperature in Degrees F

Hence, the **REQ_FUEL** value is the amount of fuel (in tenths of milliseconds) required for a **MAP** reading of 100 Kpa, manifold air temp of 70 degrees F, and a barometer of 30.00 In Hg, for one complete filling of one cylinder (Volumetric Efficiency = 100%), without any enrichments.

To introduce the enrichments (and determine required pulsewidth), we use:

$$PW = REQ\_FUEL * VE(RPM, MAP) * MAP * \gamma Enrich + \gamma Accel + InjOpen$$

PW = Injector Pulsewidth in tenths of millisecond
VE = Volumetric Efficiency in percent/100 (from 2-D interpolation)
MAP = Manifold Absolute Pressure in KPa
GammaEnrich = Final Gamma of all enrichments (multiplied factor)
GammaAccel = Acceleration enrichment
InjOpen = Injector open time constant (tenths of milliseconds)

The definition of **GammaEnrich** is:

$$\gamma Enrich = \frac{Warmup}{100} * \frac{O2\_Closed\_Loop}{100} * \frac{AirCorr}{100} * \frac{BaroCorr}{100}$$

GammaEnrich = Total enrichment value
Warmup = After-start and warmup enrichment factor, in percent
O2_Closed_Loop = O2 closed loop adjustment factor in percent
AirCorr = Air Density correction in percent – 100% is defined as air density at 70 degrees F
BaroCorr = Barometer correction factor in percent – 100% is defined as a barometer of 30.00 In Hg

As can be seen, the enrichment values, as well as quantities like **MAP**, **VE**, and **O2_Closed_Loop**, are directly multiplied to the REQ_FUEL value to obtain pulsewidth. Because of this fact, quantities can be rearranged easily, as will be illustrated in a moment.

The closed-loop feedback mode of the oxygen sensor is a very simple loop. MegaSquirt employs a narrow-band single-wire O2 sensor, which produces a voltage of roughly 0.5 volts at a stoich value of 14:7 air-to-fuel ratio. Richer mixtures (lower AF ratio) produce voltages above the 0.5 volt mark, leaner mixtures produce less voltage. What the MegaSquirt control software does is slowly increase the **O2_Closed_Loop** enrichment value until the produced voltage from the O2 sensor swings above 0.5 volts. At this point, the controller starts lowering the **O2_Closed_Loop** value, until the voltage swings below 0.5 volts. The process is repeated over and over, thus the mixture is dithered rich and lean around the 0.5 volt mark.

The value of the O2_Closed_loop variable at the 0.5 volt crossing can be used to determine if the overall fuel values (i.e. VE table entries) are at (or close) to a value of 14.7:1. An **O2_Closed_Loop value** of 100% is the desired target value, and when the closed-loop mode is disabled (during accelerations, low engine RPM, etc) the value is set to 100%. Hence, maintaining the VE table values such that they produce an average **O2_closed_Loop** value of 100% is desirable.

The PC Configuratior application has a PC datalogging mode, which records the current runtime data values ten times per second to a ASCII disc file. These data records contain information such as RPM, MAP, interpolated VE, O2 output voltage, and **O2_Closed_Loop** enrichment value, as well as information on engine status (accelerating, warmup, cranking, etc). The information in this file is sufficient to determine O2 crossing points (0.5 volts), with the corresponding enrichments.

The mode of operation of the VE tuning using O2 sensor crossing points is quite simple - all one has to do is drive the vehicle around while datalogging (using a wide O2 sensor authority range), and then use the data to perform a post-analysis to determine new VE table entry values which maintains 100% closed-loop enrichment

While driving, there will be many 0.5 volt crossing points occurring, as the closed-loop mode maintains the 14.7 AFR value. These transitions will occur at many different RPM and MAP values, depending on the vehicle, road, load, etc. The goal is to drive under many different steady-state conditions, in order to span as much of the VE table as possible.

After taking the data, a post-analysis is performed using the least-squares fit application. This application finds all of the valid O2 transition data records, and discards all other records. Next, the new VE value is determined from the data. Since all enrichments are multiplied together, it is easy to re-arrange the enrichment values. In this case, it is desirable to adjust the **VE** value such that the **O2_Closed_Loop** value is maintained at 100%, hence we operate around the desired 100% operating point for **O2_Closed_Loop**. The relation below illustrates how the VE value can be adjusted to achieve the 100 percent point:

$$\frac{VE_{old}}{100} * \frac{O2\_Closed\_Loop_{old}}{100} = \frac{VE_{new}}{100} * \frac{100}{100}$$

The point to note here is that the transitions occur at various MAP and RPM points, and most likely those points will not fall on one of the grid VE points. In order to utilize all of the data, a least-squares method of linear fitting is used. The method basically takes all computed VE points within a 4-point region (or "square") and uses them to determine the four corner values. In order for the fit to work, at least four VE values are required, with more points being preferable.

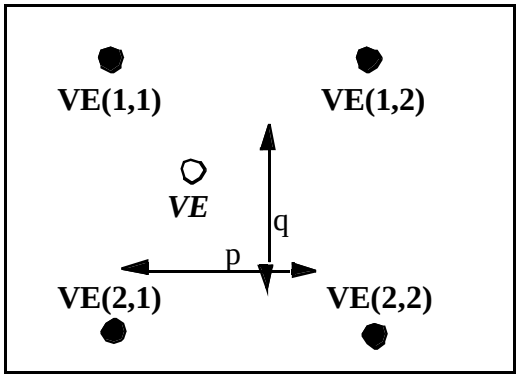
To discuss the least-squares method, a review of the VE interpolation technique is required. The equation for 2-d interpolation is presented as the following:

$$VE = VE_{11}(1-p)(1-q) + VE_{12}(p)(1-q) + VE_{21}(1-p)(q) + VE_{22}(p)(q)$$

Where VE is the interpolated VE value derived by corner values VE11, VE12, VE21, and VE22. Variables p and q are normalized positional values (range 0 to 1) representing RPM and MAP, respectively, and are given by:

$$p = \frac{RPM - RPM1}{RPM2 - RPM1}$$

$$q = \frac{MAP - MAP2}{MAP2 - MAP1}$$



The above variable naming substitutions are substituted with the following to simplify the equations:

A = VE11
B = VE12
C = VE21
D = VE22
(1-p)(1-q) = m
p (1-q) = n
(1-p) q = o
p q = r

So, the relation for the ith measured VE value is given by:

$$VE_i = Am_i + Bn_i + Co_i + Dr_i$$

To perform the least-squares minimization, define:

$$\sum_{i=1}^{nsamples} (VE_i - Am_i - Bn_i - Co_i - Dr_i)^2 = \min$$

To find the minimum, take the derivatives with respect to each determined value and set equal to zero:

$$\frac{\partial}{\partial A} = 2\sum_{i=1}^{nsamples}(VE_i - Am_i - Bn_i - Co_i - Dr_i)m_i = 0$$
$$\frac{\partial}{\partial B} = 2\sum_{i=1}^{nsamples}(VE_i - Am_i - Bn_i - Co_i - Dr_i)n_i = 0$$
$$\frac{\partial}{\partial C} = 2\sum_{i=1}^{nsamples}(VE_i - Am_i - Bn_i - Co_i - Dr_i)o_i = 0$$
$$\frac{\partial}{\partial D} = 2\sum_{i=1}^{nsamples}(VE_i - Am_i - Bn_i - Co_i - Dr_i)r_i = 0$$

Eliminating constants and rearranging in matrix form yields:

$$\begin{bmatrix} \sum m_i^2 & \sum n_i m_i & \sum o_i m_i & \sum r_i m_i \\ \sum m_i n_i & \sum n_i^2 & \sum o_i n_i & \sum r_i n_i \\ \sum m_i o_i & \sum n_i o_i & \sum o_i^2 & \sum r_i o_i \\ \sum m_i r_i & \sum n_i r_i & \sum o_i r_i & \sum r_i^2 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} \sum VE_i m_i \\ \sum VE_i n_i \\ \sum VE_i o_i \\ \sum VE_i r_i \end{bmatrix}$$

Solution of equations can be solved for A, B, C, and D by Gaussian Elimination, or by the use any linear equation solver routine.

Now that one has the solution for four cells (or square), adjacent squares can re-use these results, thus reducing the required computation. For instance, the square immediately to the right of this share the VE12 and VE22 results (B and D), so these can be held fixed and thus the equation set can be set up to solve two unknowns instead of four. This is repeated over the entire VE table range in all directions.

This method of determining VE values will allow one to quickly determine their VE table map based on O2 sensor readings. In addition, this method provides a good check for other enrichment calculations. For example, perform this test during cold weather, and then repeat during warm weather. If the air temperature correction is correct, then the results should be the same. If not, then one has an indication of the magnitude difference, which can lead to refinement in this correction (i.e. relocating the sensor, etc).