MAF Analyzer
By Bowling and Grippo

Rev. 1.01
December 2010

Table of Contents

Introduction

The purpose of this document is to describe the functionality and operation/use of the **MAF Analyzer** application developed by Bowing and Grippo. The program is designed to analyze Mass Air Flow meter (MAF) transfer functions for the purpose of implementation in internal combustion engine control systems (also known as ECUs, ECMs, and Powertrain Controllers).

The use of a MAF sensor in a internal combustion engine control system is primarily to determine the instantaneous rate of mass air entering the intake tract of the engine, in order to determine the amount of fuel required to maintain a targeted fuel/air ratio. The vast majority of modern MAF sensors employ either a hot-wire or hot-film element sensor which is mounted in fashion within the intake air stream. The sensor element is convection-cooled by the intake air stream. An electrical circuit is arranged with the sensor element, providing a controlled voltage/current thru the element. The circuit is designed to maintain the sensor element temperature at a fixed (constant) value, regardless of intake air flow. The amount of "adjustment" in voltage/current to the sensor element to maintain this constant temperature is proportional to the mass air flow rate. The actual presentation of this control adjustment signal is usually in the form of an analog voltage or frequency signal.

In general use, a MAF sensor is tested using a airflow sensing test bench in order to determine the correlation between the electrical output signal and the mass air flow rate (for example, a cylinder head flow bench adapted for MAF sensor flow use). The test facility will flow the MAF sensor at several discrete flow rates and the corresponding voltage or frequency is recorded. The test facility calibration sensor arrangement, used to determine the actual mass air flow rate, is usually an external sensor such as a Laminar Flow Element, Turbine Accelerometer, Pitot Tube, or the air source controlled with sonic air nozzles. The result of the testing is a correlated series of measurement flow rates with corresponding MAF sensor output values. This correlated data represents the MAF sensor transfer function response curve which relates the mass air flow rate to the sensor output signal. Generally a flow test will produce 10 – 20 discrete flow/response data pairs spanning the range of the MAF sensor. The actual range of test will depend on the bandwidth of the sensor and the sensor output signal type. For voltage MAF output the range is generally from 0 volts to 5 volts. For a frequency-based MAF output signal the range can be from a few hundred Hertz to 15 KHz or more. In both cases a higher numerical number generally corresponds to a higher MAF flow rate.

For reporting purposes, a small number of transfer function pairs are usually sufficient. However, for real-time control system applications it is desired to know the transfer analytic curve to a much finer resolution. MAF sensors provide a continuous analog voltage/frequency output, and it is desired to know this response equal to the resolution of the data acquisition means within the engine controller.

With this desire for a finer representation of the MAF transfer function, it is evident that normal sources of test data do not fulfill this requirement. For example, for a 12-bit ADC converter there are 4096 unique voltage/flow data combinations – contrast this to the usual 10 – 20 data points generated from a standard flow test. Clearly there needs to be a method to relate the sparse measurement range to the finer resolution provided by the control system – while maintaining the physical meaning of the sensor output response.

This is where the **MAF Analyzer** comes to the rescue. The **MAF Analyzer** application is designed to determine the best implementation of a mathematical model to the sparse data set quantity, then utilizing this mathematical relation to generate an output response of any range and number of data points.

Mathematical Model

In the year 1914, a paper was published by King describing a physical model for hot-wire anemometers[1]. The model that was derived was based on the energy balance of a wire between heating the wire to a target temperature and the cooling due to thermal loss from mass air flow across the element. The equation by King is expressed as the following:

$$E^2 = bV^{c'}$$

Where **E** is the voltage on the hot wire or film element, V is the mass air flow rate, and ***b*** and ***c'*** are coefficients which describe the physical properties of the wire and environment.

What this equation tells us is that the power supplied to the wire (E^2), used to maintain the wire temperature at a constant temperature, is equal to the mass air flow rate of fluid (air in our case) at some power response shape. Note that this form of the equation zero flow results in zero wire voltage – this is often not the case in the practical world - and we will modify this in a minute to include offsets. But for the moment let's keep focused on this form of the equation and understand its use.

First, to simply the implementation of the King equation model with real-world data, it would be nice to get rid of the voltage squared term on the left. In reality it's not hard to just roll the squared term to the other side – since we are already dealing with an exponent there why not just roll the square into it as well. No problem with that…:

$$E = bV^c$$

See where now we have E on the left, and the new exponent ***c*** = ***c'***/2. Its just math – we can do things like this and not affect the spirit of the meaning. Using this form simplifies the model in that the sensor output E is now in direct measurement units – no need to square the results just to get into a model. Also note that the variable E represents the voltage output of the voltage-type MAFs. For frequency-type MAFs (like GM style) it is possible and legitimate to use the frequency values for the variable E, as long at the correct coefficients ***b*** and ***c*** are used.

OK – you really should be asking yourself the following question "where do the coefficients ***b*** and ***c*** come from"? Glad you asked! The coefficients ***b*** and ***c*** are determined numerically such that the equation form matches the measured data values,

---

[1] King, L.V.: On the convection of heat from cylinders in a stream of fluid, Phys. Trans. R. Soc., London, A214, pp. 342-373, 1914

as best as possible. The word "matches" is the word of the day – in practically all cases the equation will not exactly match all of the data presented to it, but we want the best fit of the equation to match the data. The standard method for achieving this is knows as "least-squares fitting".

Linear least-squares data fitting simply finds the best numerical value for the ***b*** and ***c*** coefficients such that the King equation curve best fits the data, the best it can. It is called least-squares because it compares each of the data values to the actual curve by subtracting and squaring the result, adding up all of the data to a final "goodness of fit" value.

For mathematical equations which are linear or polynomial in nature, the least-squares calculation can be done in one quick step. However, the King equation is not linear (because of the c exponent that is one of the fit parameters) – but it is possible to make the King equation linear and then perform the fit operation. The linearization occurs by taking the log function of both sides of the equation – in our case the equation:

$$E = bV^c$$

Becomes the equivalent linear function:

$$\ln(E) = \ln(b) + c\ln(V)$$

To build up the least-squares fit equation, the King equation is summed over all of the measured data values:

$$\chi^2 = \sum_{i=1}^{n} (\ln(E_i) - \ln(b) - c\ln(V_i))^2 = \min$$

Where $\boldsymbol{X^2}$ (Chi-squared) is the goodness-of-fit result that is to be minimized. OK – so what good is this $\boldsymbol{X^2}$ equation junk to us anyway? Well, if you use this equation and take derivatives with respect to the coefficients b and c and set these to zero, you get an equation set which you solve:

$$\frac{\partial \chi^2}{\partial \ln(b)} = 2 \cdot \sum_{i=0}^{n} (\ln(E_i) - \ln(b) - c\ln(V_i)) = 0$$

$$\frac{\partial \chi^2}{\partial c} = 2 \cdot \sum_{i=1}^{n} (\ln(E_i) - \ln(b) - c\ln(V_i)) \sum_{i=1}^{n} \ln(V_i) = 0$$

Rearranging in matrix form:

$$\begin{bmatrix} n & \sum_{i=1}^{n}(\ln(V_i)) \\ \sum_{i=1}^{n}(\ln(V_i)) & \sum_{i=1}^{n}(\ln(V_i))^2 \end{bmatrix} \cdot \begin{bmatrix} \ln(b) \\ c \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n}(\ln(E_i)) \\ \sum_{i=1}^{n}(\ln(E_i))\sum_{i=1}^{n}(\ln(V_i)) \end{bmatrix}$$

The above matrix is solved using Cramer's rule to yield the b and c coefficients – you could also use matrix inversion, L-U decomposition, and a host of other linear algebra numerical techniques. The method does not matter – the goal is to determine the ***b*** and ***c*** coefficients that best fit the presented data.

OK – remember earlier on we stated that with most MAF sensors, zero mass air flow will not equal a zero output signal? In most cases there will be an offset voltage or frequency coming out of the sensor even when there is no air flow movement. A voltage-out MAF may generate an output voltage of a few tenths of a volt at rest air. A frequency MAF may output a few hundred hertz at no MAF flow rate. This offset is always there, added in with the signal generated from the mass air flow signal circuitry in the MAF. So its important to include the offset in the equation since it is part of the system. Its real easy to do – just throw in an offset term we'll call the ***a*** coefficient:

$$E = a + bV^c$$

So the ***a*** coefficient simply adds to the wire signal from the King equation – it's that easy. And, to determine the ***a*** coefficient may be kinda easy, or a bit more difficult. In fact the offset coefficient determines if the linear least-squares method can be used, or a more complicated non-linear least-squares method needs to be used – here's why…..

If you have the measurement for the MAF output at zero flow rate, then the ***a*** value is just that offset value. And, when performing the linear least-squares fit described above, the offset value ***a*** is subtracted from the MAF output signal data values E_i above before the algorithm is used. What this does is remove the offset effect from the data values before the fit. Later on, when using the King equation to generate the MAF transfer function the offset term is added back in (as in the equation above) to yield the final result.

All this is great and all, but what, if for some chance I do not have MAF data measurements that include the zero flow rate output value? For instance, say I have a data set that was published in a magazine or on the Internet and I want to determine the King equation fit, and there is no data presented for zero flow – how do I handle this? Is there a way to determine the ***a*** coefficient without the zero flow value?

It turns out that it is possible to include the ***a*** coefficient in a least-squares fit – however the approach is quite different. The form of the King equation that includes the ***a*** offset coefficient is no longer linear when you take the log of both sides (you end up with both ***log(a)*** and ***log(b)*** terms which are not unique in the linear solution). In this case, a more numerically-intensive method can be used, called Non-linear least squares.

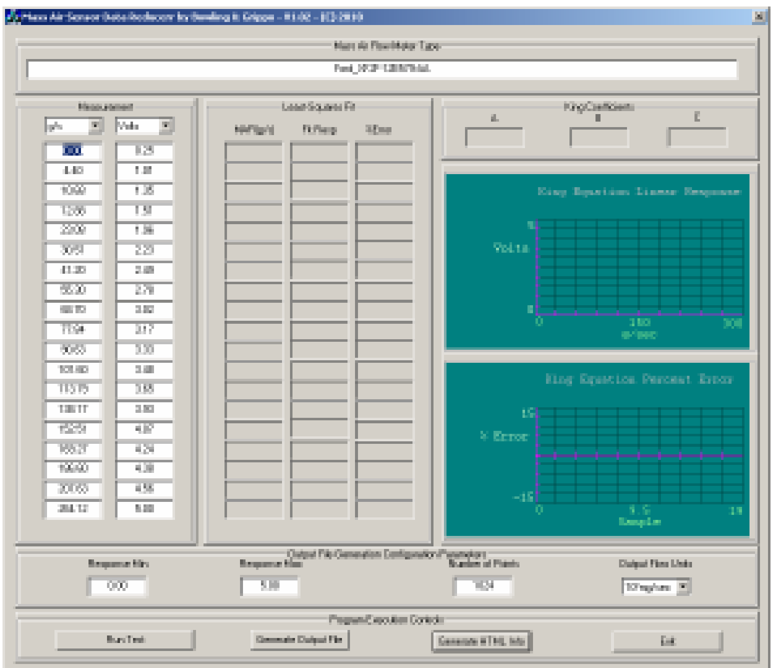
The non-linear least-squares fitting method is an iterative solution where the parameters ***a,b,*** and ***c*** are adjusted such to minimize the sum of the squares between the data and the King equation. We will not go into detail on how the algorithm works, only to say that it uses an optimal gradient-expansion method combined with a linearization of fitting function using Fourier expansions. In simple terms it intelligently adjusts the value of ***a,b,*** and ***c*** to make the X^2 result as small as possible, and goes back and tries again until the result is as good as its going to get.

The choice of computation method used in the **MAF Analyzer** (linear vs. non-linear) depends on the availability of the zero flow value. If there is a zero flow value entered in the first entry box in the **MAF Analyzer** then this value is used as the ***a*** coefficient and the linear method is used. Otherwise the ***a*** coefficient is used in the more general non-linear least squares method. Either method will work equally well.

Program Use and Execution

There is no installer for the **MAF Analyzer** – just simply unzip the contents into an empty directory and double-click the **MafAnalyzer.exe** file to execute. Along with the executable there is a file called *maf_analyzer.cfg*, this file saves the last run and automatically loads upon program startup. If there is no *maf_analyzer.cfg* file the program will open up with no dataset, and the file will be created when a computation run is executed.

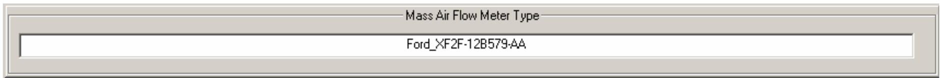
When the **MafAnalyzer.exe** program is executed the first time (from the archive) the screen will look like this:



The default dataset loaded is from a measurement of a Ford voltage-type MAF sensor. The flow measurement test units are grams/sec mass flow rate, with corresponding voltage output.

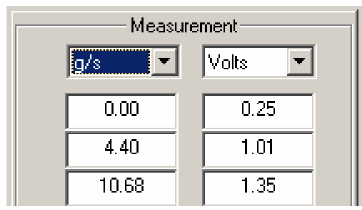
The following paragraphs describe the various controls and their operation:

Title Box – This is a character string name used to describe the test:



The string does not affect program operation; it is just for informational purposes. Note that if there are any spaces in the name it may not load up in the box the next time you run the program, or when you generate Web page output - you may want to use underscores in the name wherever there are spaces.

Measurement Type – There are two pulldown boxes, right above the section where the data is entered:



The left pulldown is the flow type entered into the program. The example above shows g/s which is grams per second. Other selection types are Kg/Hr, Lbs/Hr, and SCFM (Standard Cubic Feet Per Minute). This pulldown describes the data units for the entry boxes directly below it. Note that the program converts the data internally to grams/sec before calculation.

Also note, for **SCFM** units, this is a volumetric flow measurement at standard conditions – and volumetric flow is not the same as mass air flow, since volume alone does not define the mass of the thing (in this case air) taking up the space. To yield mass air flow rate the SCFM value needs to include the air density. Problem here is that all sorts of standards exist to define SCFM, making life more troublesome for all of us. The standard air density value of 1.2 Kg/m3 is used in the conversion for **Maf Analyzer** (29.92 inches Hg, 68 Deg F., 50% relative humidity – as defined by the OMIL R111 standard). If your CFM measurement was at some other standard then the data needs to be corrected for your test environment – you do this by computing the air density for your test and then scaling the SCFM values by the ratio. The generic air density equation is:

$$AD = \frac{\left[(0.348444 \cdot P) - h(0.00252 \cdot \mathrm{T} - 0.020582)\right]}{(273.15 + \mathrm{T})}$$

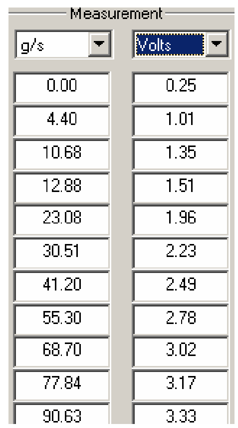
Where AD is the air density for the test (Kg/m3), P is the barometric pressure (mbar), h is the relative humidity (%) and T is the air temperature in degrees C. This leads to the scaling factor SF:

$$SF = \frac{1.2}{AD}$$

What you will do is first determine the AD value for your test environment, then the SF value. Next, you will multiply each of your SCFM values by the SF factor before entering them into the **Maf Analyzer** program.

The right pull down denotes the output type of the MAF sensor – either voltage or frequency. This needs to match the sensor type you are using – its important to set this correctly.

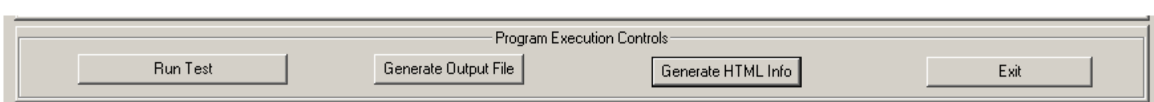
Measurement Entry Boxes – There are 19 measurement entry boxes where you can enter up to 19 data measurement pairs:



You need a minimum of 4 data pairs for the fits to operate, and up to 19 can be used. The data is entered with the slow mass air flow rates at the top and increasing on down. If you have less than 19 data values then for the boxes below then just enter zero in the flow and response boxes below the last entry. The program looks at the flow numbers and checks to see that they are increasing – if the next flow box decreases then it will not use it, or anything below it in the fit calculation. So it is important to make sure the data in entered in numerically-increasing data pairs.

The very first flow box at the top left has significance in program operation: if the value entered here is zero (meaning you have a zero-flow response number) the program wull use the linear least-squares fit method. Any number above zero (you cannot enter a negative flow rate because it would be dumb to do so) means the program will use the non-linear fitting method.

Execution Controls – There are several click-box controls located at the bottom of the screen:

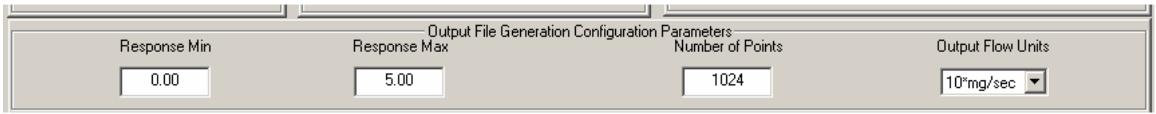


The **Run Test** box does exactly what it indicates – it executes the least-squares fit and displays the results in the output boxes and graphs. You need to click this button after you enter or make any value changes in the input boxes.

The **Generate Output File** click box is used to generate a maffactor.inc file which can be used for MegaSquirt (and other) ECUs as the generated MAF transfer function. There are control value boxes listed above the button which control the operation (see section below).

Selection box **Generate HTML Info** allows one to generate a summary file in HTML format. It lists the information shown on the screen as a formatted HTML document.

Output Control Parameters – When using the Generate Output File (above) there are a few parameters that can be set – here is a screenshot of the options:

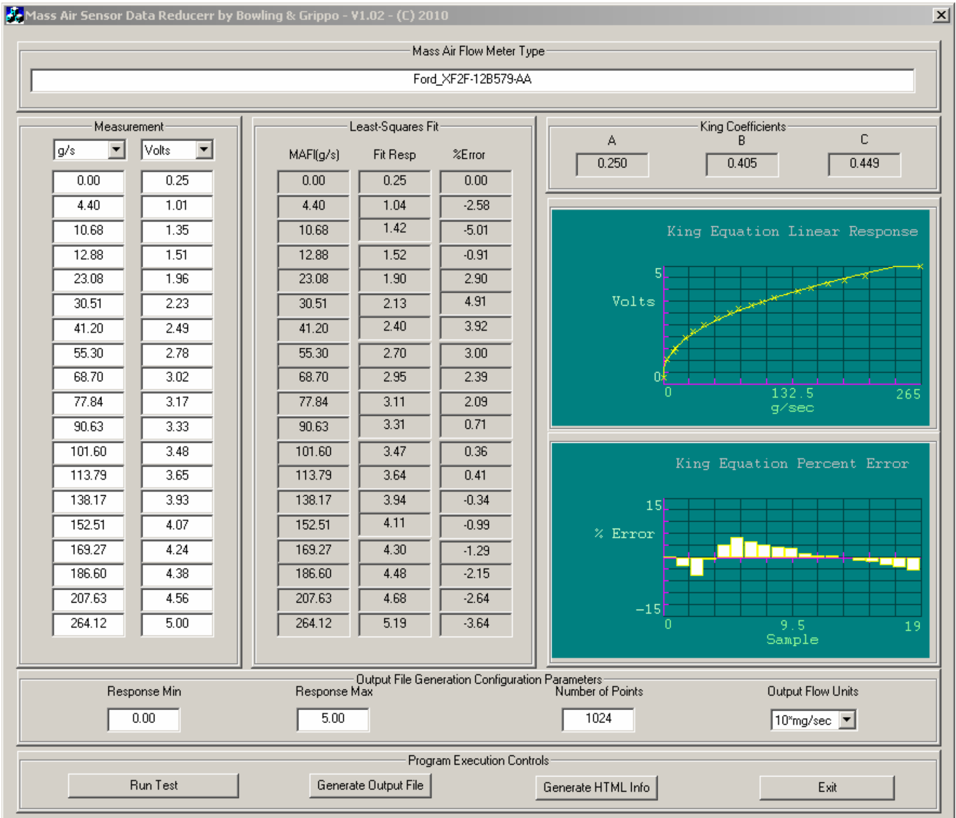


The Response Min and Response Max entry boxes set the minimum and maximum response range that will be generated in the output file. Set these to the range that you require; for voltage outputs this is normally 0 and 5 volts. For frequency MAFs this depends on the range of interest, for example 1.5KHz to 15 KHz for GM LT1 sensors.

The number of points determines the range of numbers – for the example above the minimum is 0 volts up to 5 volts, with 1024 value steps. For MS2 this needs to be 1024.

Output Flow Units specifies the output units to be used. Options are 10*milligrams/sec (default for MS2), mg/sec, g/sec, and kg/hr.

Example Run #1: Ford Voltage MAF (XF2F-12B579AA) – The data set example for the Ford XF2F-12B579-AA MAF sensor measurement is displayed with the results from a **Run Test** operation:

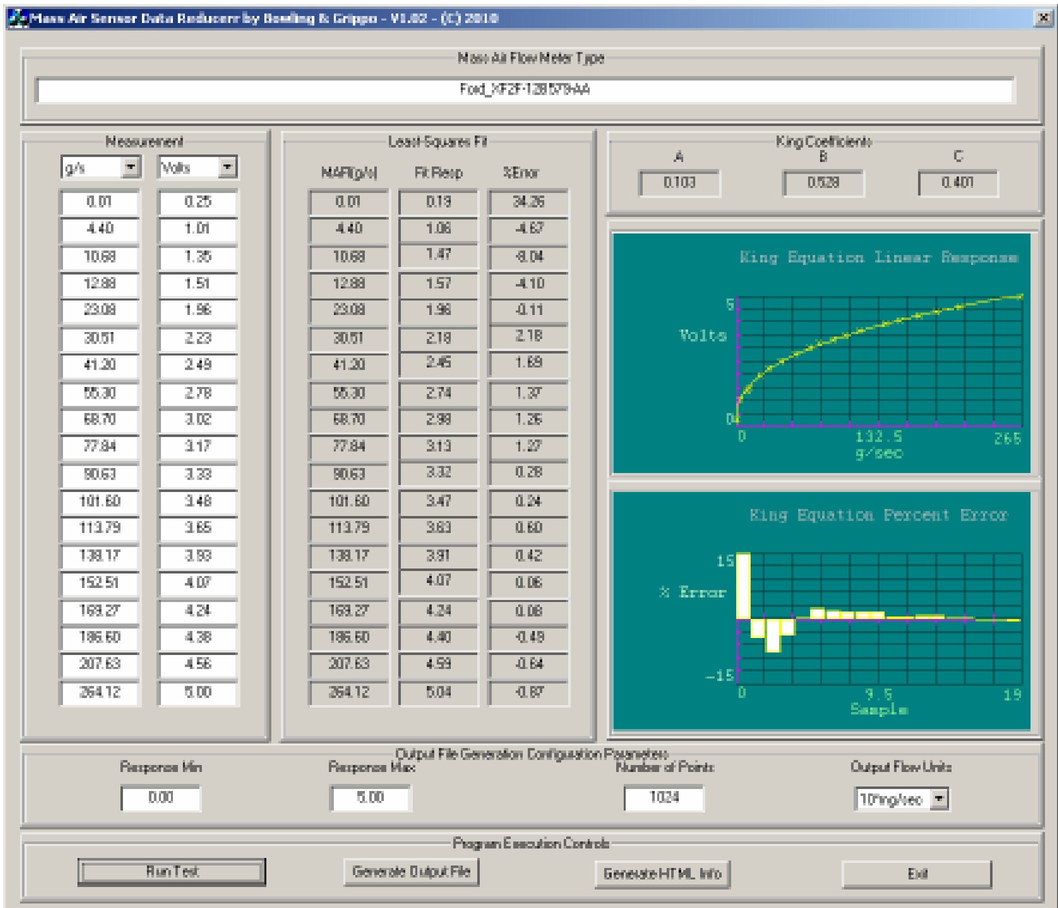


You will see that the boxes in the center are now populated with numbers. This shows the flow in grams/sec (regardless of input range – it always get converted internally to grams/sec before the fit operation). The second row is the fit response; this is the value from the King equation for the corresponding flow. The %Error is the error (percent) of the fit value from the measured data value – this is in indicator of how well this particular data measurement value matches the fit.

To the right of this are the three King coefficients ***a, b,*** and ***c***. When used in the King equation the flow is specified as grams/sec, regardless of the units of the measurement set entered into the program. The response of the program is in either volts or frequency, depending on the data units entered for MAF response.

On the right side there are two plots. The upper plot shows the King function as a continuous curve, with X marks indicating the measured data points. The bottom plot is the percent error values of the fit vs. the measured value; this is a good visual indicator of error for each data value.

One thing to note, in the data set above, the first entry box had a value entered for zero flow with corresponding response. As explained in the previous section the fit is performed with a linear least-squares method, and the ***a*** coefficient is set to this response value (you can see that ***a*** is 0.250, the same as the zero flow response value entered). For an example, in this test, lets set the 0.00 flow number to a small non-zero number, like 0.01; this is basically no flow but will force the program to use the non-linear least squares fit and make the ***a*** coefficient part of the minimization. Here is the result:



Note that the ***a*** coefficient is now 0.103 volts, this was what the non-linear minimization method determined for the best fit. Looking at the percent error plot it is evident that the first point now has a huge error; however, the higher flow rate points have a slightly smaller overall error. This clearly shows that there is variability in the measured data (perhaps systematic); however the overall fit in both cases show general agreement with the King formulism.

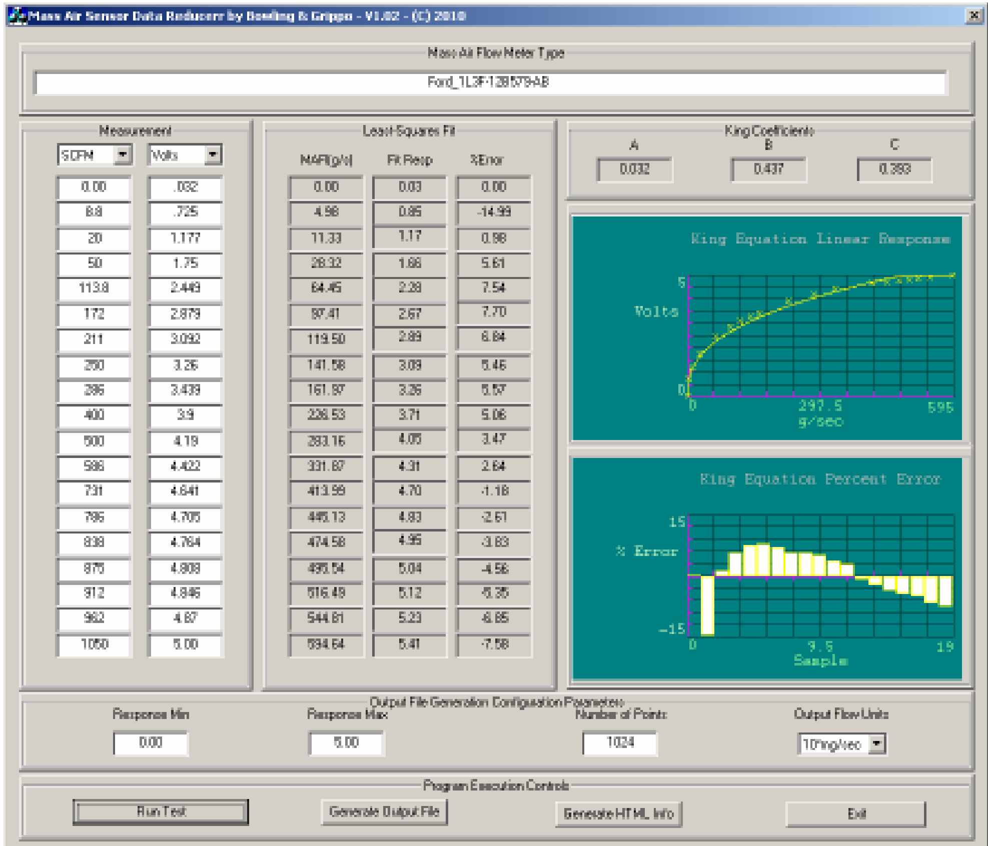
Example Run #2: Ford Lightning Voltage MAF (1L3F-12B579-AB) – Poking around on the Internet, we ran across a table of values that were measured for the 1L3F Ford Lightning 90MM MAF sensor. This is an extremely popular sensor because of its large bore and super-nice high flow rate. The table flow units are in SCFM, without any information on the environment factors like humidity, temperature, barometer, etc. Remember that SCFM is in volume units and the mass of air needs to be included (by way of using air density) which then requires the “test environment” variables. Since none are given, the only thing we can assume is that the test was done at standard conditions, and the default SCFM internal conversion is used (in other words, no correction SF factor is used because we don’t have any information on the test conditions).

Here are the flow measurement data values for the Lightning MAF:

	VDC	SCFM
1	0.0320	0.0
2	0.7250	8.8
3	1.1770	20.0
4	1.7500	50.0
5	2.4490	113.8
6	2.8790	172.0
7	3.0920	211.0
8	3.2600	250.0
9	3.4390	286.0
10	3.9000	400.0
11	4.1900	500.0
12	4.4220	586.0
13	4.6410	731.0
14	4.7050	786.0
15	4.7640	838.0
16	4.8080	875.0
17	4.8460	912.0
18	4.8700	962.0
19	4.9490	1003.0
20	5.0000	1050.0

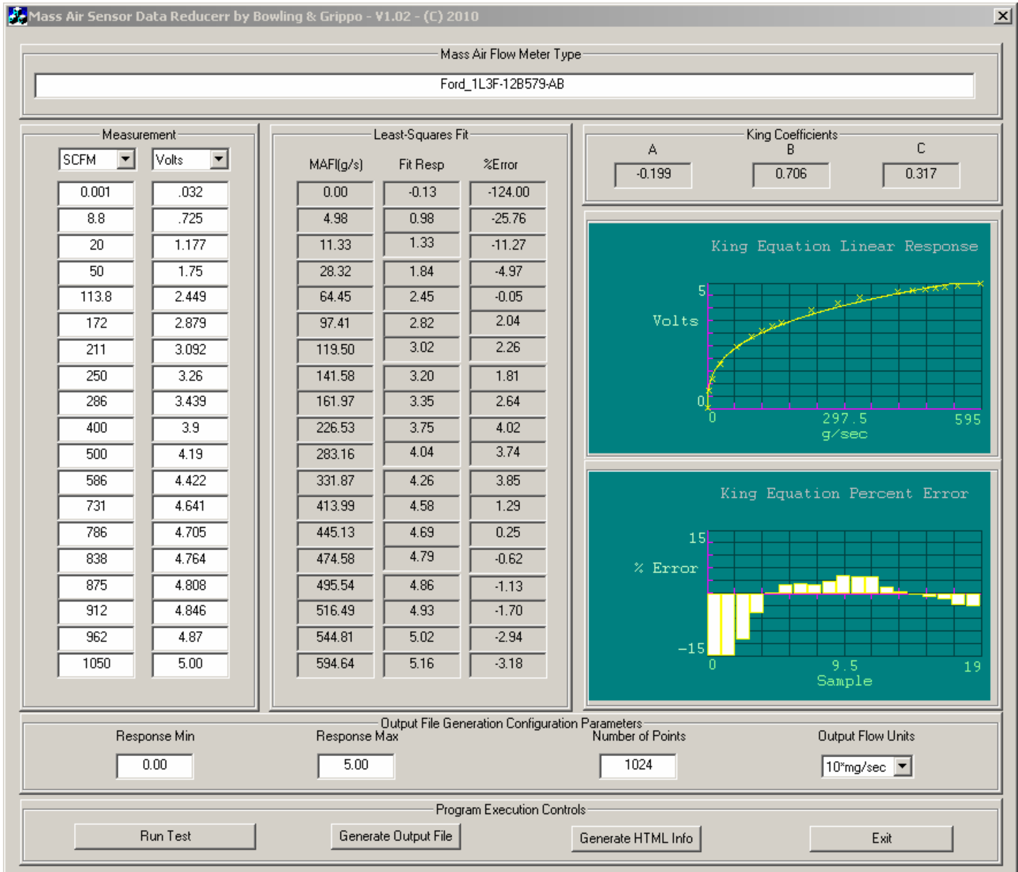
** TOTAL NUMBER OF CALIBRATION POINTS = 20 **

Entering in this data into the **MAF Analyzer** program and executing yields the following results:



First thing to note is that the program converts the SCFM to grams per second in the least-squares fit calculation, you can see the converted values in the MAF(g/s) box column. There are 20 points here, and you can only enter 19 in the **MAF Analyzer**, so the second to the last point was omitted, an arbitrary decision based on the fact that at high flows there is very little sensor output movement compared to changes in MAF flow. And this MAF sensor puppy maxes out close to 600 g/sec, which is pretty good. But, remember that it is important to size your MAF sensor to the engine application… use the 600 g/s MAF only if your engine sucks in 600 g/s of air at max RPM and load; otherwise you will be wasting sensor bandwidth.

Looking at the fit, the low end (again) has a point where there is a lot of error, and there looks to be a slight systematic error in the higher flow points. We know that its hard to accurately measure low flow rates, and once the sensor is installed on the vehicle chances are significant that the low flow calibration may change due to ducting, etc. This data set included a zero-flow value which we entered. This constrained the fit to linear least-squares. If we want to include the ***a*** coefficient in the fit we can simply make the zero flow value non-zero, but still close to zero… here is the result with the zero flow set to 0.001 SCFM (which really is still 'practically zero' compared to the many orders of magnitude of the other SCFM values):

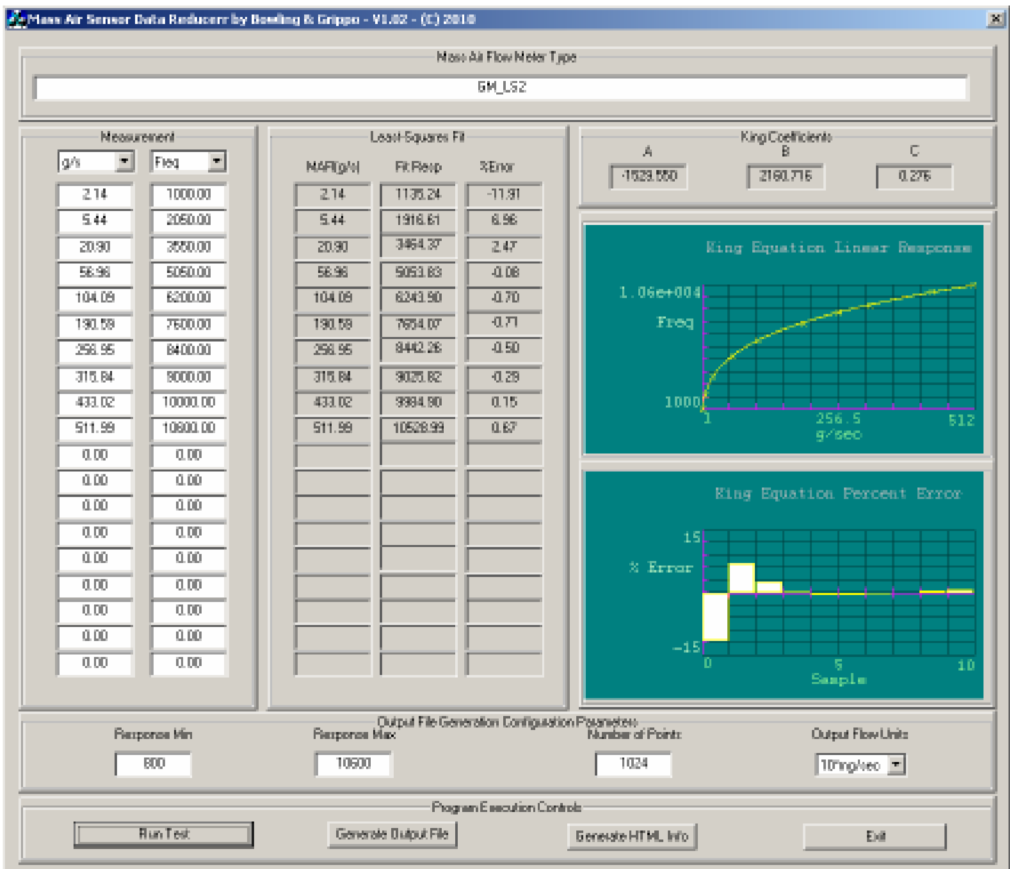


This time we get more low-end error but higher flow errors are reduced. Also notice that the value of the King ***a*** coefficient is -0.199 volts. This represents the zero-flow MAF output value. It's kinda hard to get a negative 0.19 volts out of a MAF sensor that runs from 0 to 5 volts, all positive voltages. Note that if you were to generate an output file for an ECU, negative response values (if they happen to occur) are truncated to zero, so this negative offset is not a big deal.

We have said this several times already – it is hard to measure low flow rates accurately, and the MAF sensor calibration will change on the vehicle because of installation plumbing, intake reversion effects, ECU numerical integration method, etc. So don't get all suicidal on the low end values, just try to get the higher-flow values close and be prepared to tweak in the MAF sensor on the vehicle. Most vehicles will idle with a MAF flow rate of 5 – 10 g/s so this is also a point watch in the calculated data – but again this is in the tweak zone that you have to contend with later on.

Example Run #3: GM LS2 Frequency MAF – General Motors LT1 and LSx engines use frequency-based MAF sensors. The MAF Analyzer can handle frequency MAF units equally well, but you need to make sure you set the measurement type to Freq and (very important) to set the Response Min and Response Max boxes correctly. For the GM LS2 sensor, the minimum useful response is around 800 Hz and the maximum response at max flow is roughly 10,600Hz.

There are numerous sources for the LS2 sensor flow data on the Internet and publications. There is usually gobs and gobs of data points, so we just took 10 values within the range of MAF sensor bandwidth and ran these – take a look at this (after the run):

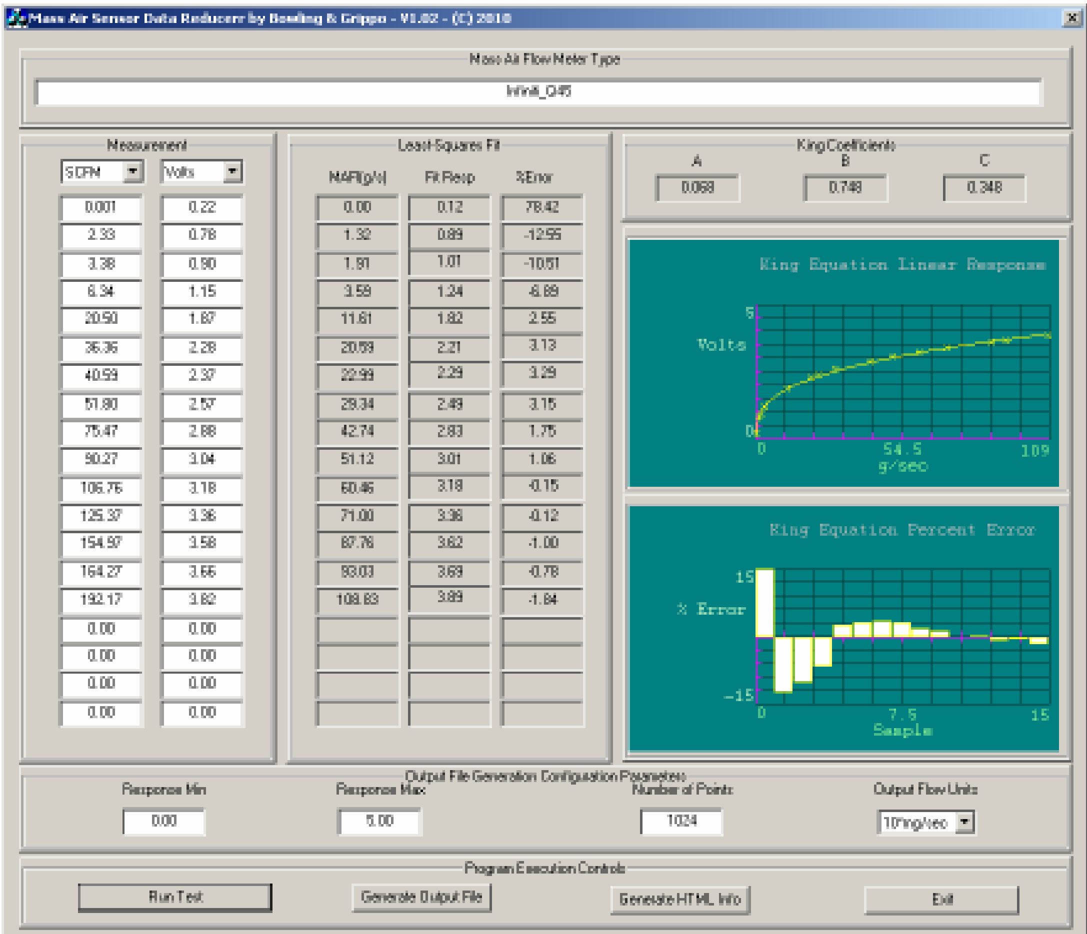


What a super-nice fit! Sure, there are the error values at extreme low flow rates, but the magnitude is low. And the higher flow data values are spot on – to have the % error in the flows from 50 g/s to 500 g/s less than a percent is excellent!

There are several advantages to frequency-based MAF sensors, and with a data fit like this they are worth considering!

Example Run #4: Infiniti Q45 Voltage MAF – The Infiniti Q45 MAF sensor is a popular sensor for 6-cylinder swaps because of the reduced bandwidth (300 g/s compared to the 500-600 g/s Ford). This makes sense because it is desirable to match the maximum flow of the MAF sensor to the maximum flow of the engine.

Here is a fit to SCFM data for the Q45 MAF:



This is a really nice fit overall, excluding the first beginning values. From 20 g/s on up the error is less than 3% which will result in very little on-car tweaking.

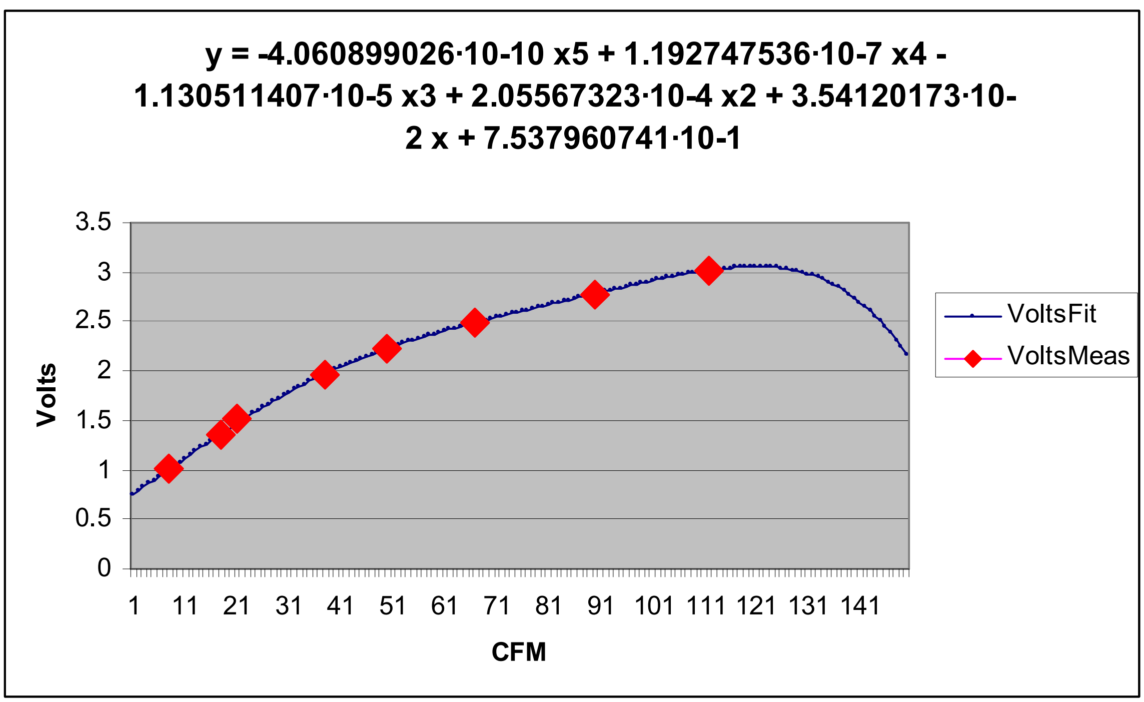
Final Thoughts

Finally, a word on why the choice of using the King function form to represent the numerical data. After all, there are many other functional forms that could be used for the data representation. For instance, a high-order polynomial could be used to fit the data. In fact, a polynomial may actually fit a given data set better than King representation. What gives on the choice?

Realize this – the data set that you have most likely has inherent error associated with each measurement point (this is known as variance). Except for special cases where the measurement was carried out on calibrated machines that were repeated with the error sources known and quantified, each of your data values has some sort of error associated with it. And the magnitude of the error may depend on flow rate, upstream and downstream air duct plumbing, measurement detection device, etc. For instance, volumetric flow measurement devices (yielding SCFM for example) rely on the accuracy of determining the flow environment (barometric pressure, humidity, temperature) to yield density, and any error introduced will introduce an error that grows with increasing flow rates. Another example is the use of Pitot manometers – small flow rates are harder to nail down numerically and will result in increased error.

So the data you throw into the **MAF Analyzer** has all sorts of error of unknown magnitude, which can be both systematic and random. Point here is you want the fit curve to go exactly thru each of your datapoints then you must think that your data is special and of the utmost accuracy. Sorry to pop your bubble… your data has measurement error that you probably do not know. This is one of the reasons why one will fit the data to the physical model of the device being measured.

Another reason for not using high-order polynomials is that they do not behave very well outside of the fit region. Often you may only have a small data set range of measurement values, but you need to be able to accurately go outside of this range (i.e. extrapolate). A MAF sensor has a pretty high bandwidth range of operation and the measurement system may not be able to go to the high flow extremes – but your engine will go there and the MAF sensor flow value needs to be reasonable. If you fit this to a high-order polynomial, the data range where you have values will fit very well, but outside of this range the fit may go goofy and all whacked-out, folding back on itself in all sorts of insane twists. The higher the order the more crazy the fit may become. For example, below is a fit of a small range of MAF data to a 5th order polynomial. The curve matches extremely well within the data range, but outside of this the curves folds back down for high flow rates – you can bet your sensor will not do this…



With the use of the King function, the fit will extrapolate to the right with increasing numerical value since it is a power function curve.

When you represent your measured data to the correct physical model you may actually increase the accuracy in a global sense. If you assume each data point has random measurement error (which is a good practical assumption) then the data fit to the functional model will tend to average the error and reduce the magnitude in some cases. And, again, with a functional model fit you can confidently extrapolate beyond the data set range.

In actual use, when you load in the data and perform the functional fit, there will be variability in how well each data point fits to the curve. Do not get all strung out on the errors or lose sleep over the fit. In practice there may be a 5% or more variance in each data point, or even much higher. If you have a really whacked-out data point, remember it will screw with the overall fit. In this case it may make sense to not include the value in the fit. As a general rule, if you start to see errors in the +/-20% range or more you may want to not use this value. Also, again – you are not shooting for 0.1% error in all of the pairs, the actual data may have 3 – 5% error. Your engine will happily run all day long with 3% error – and once you bolt on the MAF sensor the errors will increase due to ducting, bends, etc. With this said, a point of reference – commercial hotwire anemometers will often correlate to the King equation with an error of less than 0.25% over full operating range. But, these are precision probes without housings and other air-modifier devices to affect the flow over the wire.

When testing, for the data points that show a lot of deviation – please don't go and *monkey around* with the number in an attempt to make the error "*go down*". This is called cheating and all that it does is help your ego – not your data. Trust your measured values and throw out those that are suspect, but don't go mucking with the actual value. If the error is terrible everywhere then it will be time to regroup and think about using another source of measurements.

As you use the program you may note that there is more error for small flow rates and that higher flow numbers start matching the King model better. Recall that it is difficult to numerically measure low flow rates without special equipment and this can lead to increased variability. Also realize that the MAF sensor (for many of them) is a "point-source" measurement device. Many MAF sensors have a small opening that allows a sample of the air flow stream. This is usually situated within the center of the MAF sensor, axially-centered within the air flow stream. This is because in general cases the flow is greater in the center of the pipe and it has a better chance of being laminar (straight). Of course, when you go and install the MAF sensor, in order to get it to fit you had to put in several bellows, 90-degree bends, and an intake filter element. Guess what to your nicely-fitted curve? Laminar flow is all screwed up and low flow rates are in error. And I am sure you yanked off the flow-straightening screen on the MAF because some magazine article said to do this – and if its in a magazine then it must be right.

The point here is that you may have to tweak up the MAF flow curve once it's on the vehicle all hooked and plumbed-up. Normally, the lower flow rates are most affected, so this is where most of the tweaking will occur. Pretty much all modern ECU setups have a fine-tuning table for MAF flow rate transfer function that you can clean up the curve. Using a 5-gas analyzer displaying lambda or a narrow-band O2 sensor targeted to stoich you can determine the amount of correction needed and then fold this into the tweak table. You don't have to use a ND for normal engine tuning, but for MAF calibration it is much preferred over a wideband UEGO meter. If you must use a wideband, then calibrate at the stoich point of the fuel; errors on a WB are the minimal at this region. The beauty of MAF is that you can do this calibration only as a function of RPM with no load on the engine and cover the low end air flow region which is the most prone to error from your ducting.

Summary

A few points of summary:

- The **MAF Analyzer** is used as a data reducer for correlating Mass Air Sensor test flow data to a numerical model, for use in engine control applications.
- MAF sensor flow data can be measured, or obtained from magazines and Internet sources.
- **MAF Analyzer** performs least-squares fitting of the test data to a continuous model known as the King equation
- **MAF Analyzer** can be used to generate transfer function output files for ECU use and HTTP-ready web reporting.
- Do a final calibration on the car using the MAF tuning table and a narrowband sensor or 5-gas analyzer.
- Good measurement data = good correlation.
- Be honest with your data and keep in mind the required accuracy for the end application.
- Use the application to help match the MAF sensor to your engine's maximum RPM and load mass air flow rate.