

Aerodynamics

APPLICATION NOTE



Disclaimer

Pi Research makes no representation or warranties of any kind whatsoever with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Pi Research shall not be liable for any errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of the software, associated hardware, or this written material.

Pi Research reserves the right to revise this publication from time to time, and to make changes in the content hereof without obligation to notify any person of such revision or changes.

A copy of the Pi Research Terms and Conditions of Sale is available on request, and includes a declaration of the warranty and limitation of liability which apply to all Pi Research products and services.

Contents

- Introduction 5**
- Aerodynamic measurements 6**
- Dynamic pressure 8**
 - Measurement using a pitot-static tube 10
 - Estimation from wheelspeed 14
- Estimation of aero forces 16**
 - Vertical aero force measurement 18
 - Straight-line test procedure 21
- Demo disk 28**
 - GATE_TEST User math function 28
 - AERO1.DAT 29
 - Constant definitions 29
 - Math channels 29
- Pressure measurements 34**
 - Pi aero pressure sensors 34
 - Pi Air flow sensor 36
 - Air speed amplifier 38
 - Surface pressures 39
 - Radiator efficiency 42
 - Air intake efficiency 43
- Using aero maps (2D table lookup) 44**
 - Setting up the worked example 45
 - Out-of-range indices 55
- Sensor installation notes 56**
 - Air speed sensor 56
 - Air speed amplifier 56
 - Single aero sensor 57
 - Hex aero sensor 57
 - Hex aero sensor manifold 58

Sensor specifications	59
Air speed sensor	59
Air speed amplifier	60
Single aero sensor	61
Hex aero sensor	63
Sensor dimensions	65
Air speed sensor	65
Air speed amplifier	66
Single aero sensor	67
Hex aero sensor	68
Pi Pitot-static tube	69
Sensor ordering information	70
Appendix	71
Conversion factors	71
Pitot tubes	73
Contact information	76

Introduction

Aerodynamics is one of the most dominant aspects of the modern racing car, and seemingly small changes in aerodynamic setup can supply crucial changes in lap-time.

This Application Note gives an overview of various data acquisition techniques that can be used to help evaluate the aerodynamic performance of a car.

Among the Pi sensors and software features covered are:

- Single aero pressure sensor
- Hex aero pressure sensor
- Pitot-static tube
- Air flow sensor
- Laser ride height system
- Version 6 2D lookup table
- Pi wind tunnel equipment

To make use of the sensors you should be familiar with the fundamentals of aerodynamics and be able to write Math channels and define Constants using Pi Version 6 Analysis software. For more information on Math channels and Constants see *Version 6 PC Software Guide*.

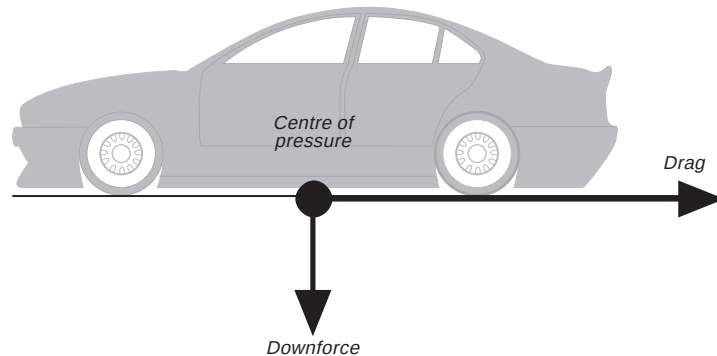
A demo disk is supplied with this Application Note. The demo disk contains files which you will need to copy to your computer hard disk. The content and use of each file is given later. The Application Note also has some worked examples.

Aerodynamic measurements

Downforce and drag

The two basic measures of aerodynamic performance are downforce and drag. An increase in downforce provides the car with extra grip potential, and therefore primarily improves its cornering capability. A decrease in drag lowers the car's resistance to forward movement, and therefore primarily improves its speed through high speed corners and on long straights.

Note that aero forces are usually resolved into the plane of the road, as shown in the following figure.



In general, it is difficult to trim more downforce into the car without increasing drag at the same time. On most circuits, the engineer is usually balancing this trade-off to optimise lap time, or to give the car better performance at an overtaking point.

ChampCars are an interesting example where the different types of circuit demand radically different aerodynamic setups. The car is configured for low drag at oval circuits, where speed through the long banked corners is restricted by the available horsepower rather than grip. In contrast, a road track circuit typically has tight corners and few opportunities to get up to top speed. A configuration with high downforce is therefore used to maximise the car's speed through the corners - a good trade-off for the diminished top speed resulting from the extra drag. The difference between the oval and road-track configurations is visually striking, as wings of vastly different sizes are fitted to the car.

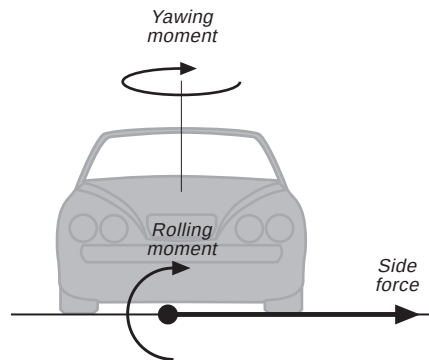
Centre-of-pressure

An additional basic measurement is the longitudinal position of the centre-of-pressure or, put another way, the distribution of downforce between front and rear axles. This aerodynamic balance can be very sensitive to changes in the car's ride height, particularly on a car utilising the 'ground effect'.

Drivers generally want a car that handles smoothly and progressively, so an aerodynamic configuration must be found where the centre-of-pressure doesn't shift dramatically as the car pitches into and out of corners.

Side forces, yaw and rolling moments

The aerodynamics of the car also contribute a side force and yawing moment, which can be considerable at large slip angles and affects the stability of the car in corners. There will also be a rolling moment through corners (left / right redistribution of downforce) due to the car's aerodynamic asymmetry in roll. The analysis of these extra three forces requires a high degree of complexity in test techniques, and can be considered secondary to the study of downforce, drag and centre-of-pressure.



Pressure distributions

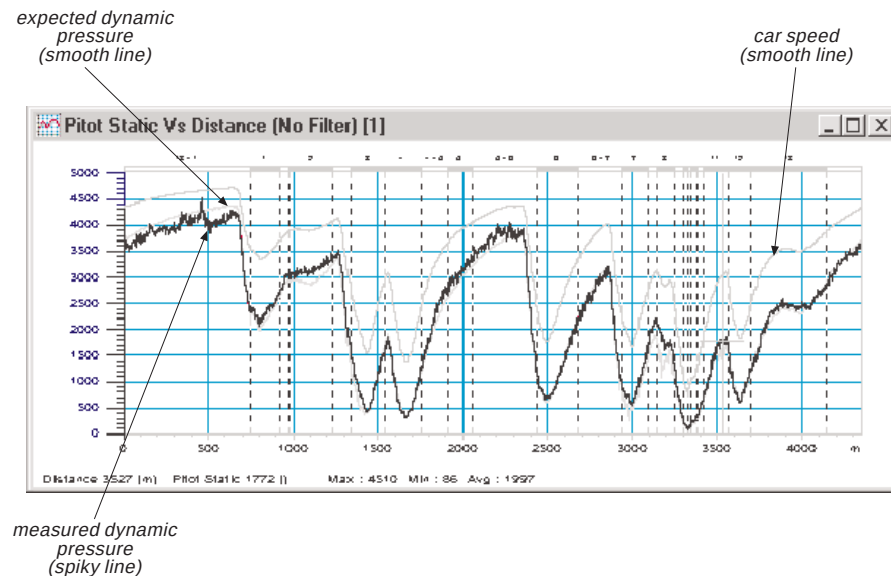
Lastly, there is the study of pressure distributions over the car's surfaces and internal air passages. This can do much to help identify the efficiency of individual components and to help the engineer compare the real-life air flows with those predicted by wind tunnel studies.

Dynamic pressure

Dynamic pressure can be thought of as the frontal pressure seen by the car due to its movement through air. As a car goes into a headwind, the dynamic pressure will be higher than expected, and with a tailwind the dynamic pressure will be lower than expected. The significance is that the aerodynamic forces on the car are proportional to the dynamic pressure, so in a headwind the car may gain extra grip round a corner, but lose top speed down a straight. Measuring dynamic pressure is therefore an important step towards understanding the performance of the car round the track.

Slipstreaming behind another car can have similar effects, but the turbulence makes analysis more difficult.

Here is a graph showing the measured dynamic pressure overlaid against the expected dynamic pressure for a lap of Estoril:



Dynamic pressure here was measured using a differential pressure sensor connected to a pitot-static tube. The expected dynamic pressure was estimated from wheelspeed.

The points of Interest are:

- Tail wind down main straight followed by large tail wind gust just before braking. This may be from the wind swirling round the end of the grandstand.
- Turn 2 benefits from a large headwind hence helping grip.
- Back straight has a headwind and hence will be slower than expected.
- Turns 8 and 11 show a very large headwind which at this slow speed is a big advantage.
- Turn 13 again shows a reasonable headwind during the first part of the corner.
- The last part of the lap shows minimal wind and hence is in the wind shadow of the grandstand.

This data shows a good lap as far as wind is concerned, which helped to give a good lap time.

Measurement using a pitot-static tube

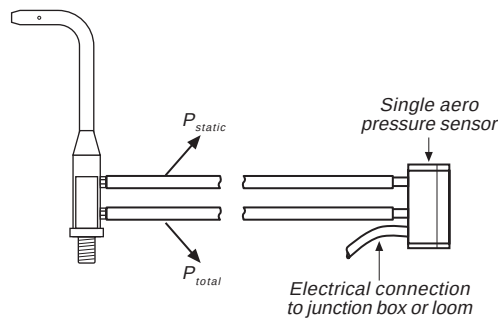
The use of dynamic pressure in the evaluation of aero forces is discussed in a later section.

A pitot-static tube has two ports: P_{total} and P_{static} . The difference between these two pressures is dynamic pressure, which can be measured by connecting a differential pressure transducer across the two ports.

There is a fuller discussion of pitot tubes in the appendix to this application note.

Pneumatic connections

The Pi pitot-static tube is a modified ellipsoidal design, with end fittings to directly match the 3mm OD tubules on the Pi aero pressure sensors. This makes it possible to connect up the pitot-static tube simply using 3mm ID tubing. A single aero pressure sensor is shown in the figure below, but it is possible to use one channel of a Pi Hex aero pressure sensor instead.



Pitot-static tube connections to a single aero pressure sensor

Note: The lower of the pitot-static tubules is P_{total} which should be connected to the pressure sensor +ve, and the upper tubule is P_{static} which should be connected to the pressure sensor -ve.

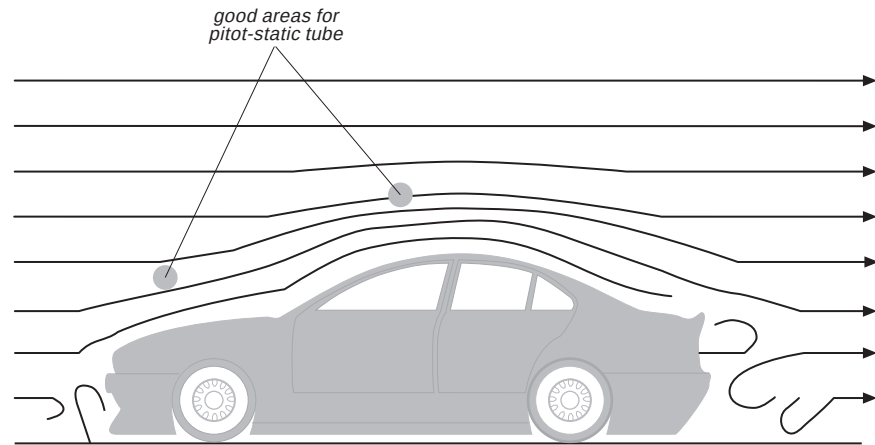
The Pi aero pressure sensors have a range of ± 2 PSI, which is equivalent to a car speed of 603kph (375mph) at standard atmospheric conditions.

Pitot-static position on the car

Ideally the pitot-static tube should be aligned to point into the airflow, and should be well away from the air disturbance caused by the car itself.

Alignment with the airflow is straightforward: the Pi pitot-static is calibrated to remain $\pm 0.5\%$ accurate up to 12° misalignment, which is typically within the combined effects of car pitch, slip angle and crosswind.

Keeping the tube completely clear of the car's air disturbance is usually impossible, so the task is to minimise the effects. The car's movement causes various parts of the air flow to speed up or slow down, which makes the pressure lower or higher than the "free stream" dynamic pressure. The general rule is to keep the tube as far from the car's body and as far forward as possible. If you have access to flow visualisation pictures taken in wind tunnels (smoke flow), then this may help indicate where the flow around the car is least disturbed.



Positioning of a pitot-static tube

Sampling

When analysing dynamic pressure readings, you should bear in mind that wind is nearly always turbulent within 10m of the ground. A car running at constant speed and heading in a 'steady' headwind will therefore see fairly rapid fluctuation on the dynamic pressure reading.

To smooth this out whilst maintaining an accurate moving average, you should consider sampling at 10Hz or greater, and applying a 2Hz filter.

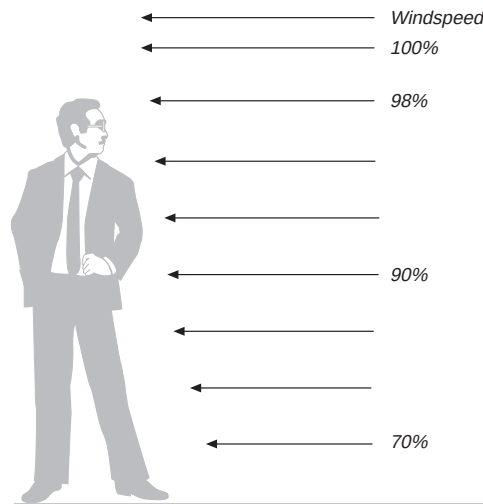
Units

The SI units for pressure are Pascals (Pa), which are equivalent to Newtons per square metre. Alternatives are PSI (pounds per square inch) and mm of water column. Full details of unit conversions can be found in the Appendix.

Wind profile

A frequently asked question is about the magnitude of the headwind close to the ground. It is true that the wind speed diminishes as you get closer to the ground - you can easily test this by lying down! - but the effects are marginal providing that the pitot-static tube is positioned at least 200mm from the ground.

The exact profile of wind speed depends on the type of surface - concrete, water, grass, bushes etc. and is compounded by the leeward "shadow" of surrounding obstructions such as walls and buildings. For road surfaces, a headwind reading at 2m above the ground will lessen to 90% down at 1m, and down to 70% at 0.25m.



Effects of wind at differing heights

Estimation from wheelspeed

Dynamic pressure can be estimated from wheelspeed using the following formula:

$$P_{\text{dynamic}} = \frac{1}{2} \rho V^2 \quad (1)$$

where (the Greek letter 'rho') is the air density measured in kg/m^3 , and V is the speed in m/s .

This is derived from Bernoulli's equation, which relates the pressure distribution within a body of air to the local velocity of the flow.

Now, air density is dependent on the ambient temperature and pressure, according to the following equation:

$$\rho = 1.255 \times \frac{288.15}{T_{\text{ambient}}} \times \frac{P_{\text{ambient}}}{101325} \times V^2 \quad (2)$$

where

T_{ambient} is in kelvin

P_{ambient} is in pascals

Thus for American standard atmospheric conditions:

$T_{\text{ambient}} = 15^\circ\text{C} = 288.15\text{K}$

$P_{\text{ambient}} = 1\text{bar} = 101325\text{Pa}$

air density is 1.225kg/m^3 .

Therefore, combining (1) and (2) gives:

$$P_{\text{dynamic}} = 1.255 \times \frac{288.15}{T_{\text{ambient}}} \times \frac{P_{\text{ambient}}}{101325} \times V^2$$

$$= 0.001741 \times \frac{P_{\text{ambient}}}{T_{\text{ambient}}} \times V^2$$

To estimate dynamic pressure from wheelspeed in Version 6, you need to take the following steps: define two constants (Ambient temperature and Ambient pressure) and create a math channel for the dynamic pressure.

To define the two constants:

1. Choose "Constant Definitions..." from the Define menu. The Enter Constants Definitions dialog appears.
2. Create following two new constants:
Ambient temperature = 278
Ambient pressure = 101300

Since these will vary from outing to outing, it may be best to make them outing-dependent constants.

To define the dynamic pressure math channel:

1. Choose "Math Definitions..." from the Define menu. The Math Definitions dialog appears.
2. Create the following math channel:

Dynamic pressure = 0.001741 * Ambient pressure / Ambient temperature * (Speed ^ 2)

This math channel assumes SI units (K, m/s, Pa, kg/m³). If any other units are being used, then appropriate factors will have to be included. Conversion factor tables can be found in the Appendix at the back of this application note.

For example, to derive dynamic pressure in PSI from speed in mph, ambient temperature in Fahrenheit and ambient pressure in bar, you would have the following math channel definition:

Dynamic pressure = 0.009206 * Ambient pressure / (Ambient temperature + 459.67)
* (Speed ^ 2)

Estimation of aero forces

In theory, the aero forces generated by a car are proportional to the dynamic pressure.

i.e. $\text{force} = A \times C \times \text{dynamic pressure}$

where

A is the frontal area of the car

C is one of a number of non-dimensional force coefficients that characterise the car's aerodynamics (e.g. C_D = Drag coefficient).

So:

$\text{Drag} = A \times C_D \times \text{dynamic pressure}$

$\text{Lift} = A \times C_L \times \text{dynamic pressure}$

etc.

If you already know the force coefficients C_D , C_L etc..., say from wind tunnel tests, then you can estimate the aero forces on the car round the circuit using the dynamic pressure reading. Combined with information from suspension strain gauges and inertial sensors, you can begin to break down how the car's forces are composed of aerodynamic and inertial components.

Conversely, if you don't know the force coefficients, or you want to confirm the force coefficients measured in the wind tunnel, then you need to measure the aero forces on the car, say by looking at damper displacement during a straight-line test. The force coefficients can then be generated by the following:

$$C = \frac{1}{A} \times \frac{\text{force}}{\text{dynamic pressure}}$$

where

A is the frontal area of the car in m^2

force is in N

dynamic pressure is in Pa.

This section covers the following:

- General techniques for evaluating the vertical aero forces on a car
- A specific procedure for measuring aero forces at a straight line test

Of course, the simplistic equations above are complicated by the fact that the force coefficients are usually dependent on the car's ride height:

i.e. $C = A \times C_D(\text{frontride}, \text{rearride}) \times \text{dynamicpressure}$

Measuring ride height accurately is therefore an important component of aerodynamic analysis. A later section details how to use aero maps that store the force coefficients as a function of front and rear ride height.

A note on non-dimensional coefficients

The basic reason for using non-dimensional force coefficients is that their values are independent of speed or ambient conditions. This provides for simple run to run comparisons. Direct comparison of lift and drag force would be of little use if, for example, wind conditions had changed between runs, resulting in a change in the dynamic pressure. It would then be difficult to attribute changes in lift or drag directly to setup changes, since part of it would be due to the change in dynamic pressure.

Vertical aero force measurement

This section covers general techniques for evaluating the vertical aero forces on a car:

- car setup considerations
- instrumentation required on the car
- calculations required to resolve the force measurements
- useful data plots

Car setup

When trying to confirm an aero map measured from a wind tunnel it is essential to be able to accurately determine the front and rear vertical loads for a range of known ride heights. In a wind tunnel it is easy to position the model at a fixed and measurable height above the floor of the wind tunnel. However, on the track it is not possible to maintain such constant conditions because the car is continually moving around due to inertial loads and bouncing due to road irregularities.

If a test is being carried out specifically to evaluate the aero forces, then a special car setup may be considered in order to optimise the measurements.

One solution for car setup is to minimise body motion relative to the ground by removing the springs and fixing solid links in their place. This will certainly reduce suspension motion, but without damping to absorb some of the shock, there will be a dramatic increase in the variation of contact patch load. This is generally not a good technique.

Instead, tests conducted with very soft and optimally damped suspension will yield the minimal variation in dynamic suspension motion. This will also result in a steady state deflection due to the mean aerodynamic load, but this excess suspension travel can be removed by modifying the static ride height to ensure the target ride height is achieved at the target speed.

In summary, the recommended car setup is as follows:

Suspension

- Soft Springs
- Remove any influence of bump rubbers or packers
- Adjust static ride height and pre-load to obtain target ride height.
- Remove any non-linear influence of anti roll bars or Panard rods, preferably run with no anti roll bars

Damping

Set to optimise the ride quality at the target speed and ride height

Instrumentation

Downforce is most simply measured directly by strain-gauged elements in the suspension. Great care should be taken to ensure that the load-sensing element will only react to loads along the axis of the strut and not be affected by bending loads. Saloon cars with McPherson struts will have very large bending loads especially at the end of wheel travel.

An alternative approach is to estimate vertical force from the suspension deflection.

Additional measurements required are:

- Front and rear ride height measured using suspension potentiometers or (more accurately) Pi laser ride height sensors
- Pitot Static tube with Pi Aero Sensor

To resolve out the inertial forces, it is necessary to know the acceleration of the car. On a perfectly flat surface, the weight transfer to the corners of the car can be estimated using the lateral and longitudinal acceleration of the car. These can be measured using, respectively, the Pi LCU's inbuilt accelerometer and the 'in-line accel' channel, which is the derivative of speed.

However, if the test is being performed on a cambered or hilly surface - i.e. 99% of circuits! - then it is essential to know accurately what the forces are due to the vertical acceleration of the car.

- Vertical Acceleration measured using Pi HP Single Axis Accelerometer

Alternatively, all three accelerations can be measured with greater accuracy using a single device:

- Lateral, longitudinal and vertical acceleration measured using Pi HP 3-Axis Accelerometer

Measured Force Correction

With some or all of the above sensor readings, you have sufficient track data to calculate the vertical loads due to aerodynamic effects. Depending on the complexity of your model, you may need to perform the following corrections to the force measurements:

- Calibration of force readings using the vertical force velocity ratio between tyre contact patch and load cells
- Correction for vertical acceleration especially on banked oval type tracks
- Correction for any longitudinal acceleration
- Correction for any anti-squat due to engine strut
- Correction for jacking forces, again especially for data collected on oval banked tracks

Recommended Plots

It is difficult to generalise but the data should be filtered at 5Hz for track data and 2Hz for straight-line (drag strip type) testing.

The following 2D plots in Pi Version 6 will yield useful information about the aerodynamic performance:

- Corrected Total Aero Force vs Dynamic Pressure
 - Overall slope equals $C_L A$
 - Local slope shows $C_L A$ at that speed and ride height
- Front Corrected Aero Force vs Corrected Total Aero Force
- Corrected % Front Aero Force vs Dynamic Pressure
- Corrected Front Aero Force vs Front and Rear Ride Height
- Corrected Rear Aero Force vs Front and Rear Ride Height

Straight-line test procedure

By running up and down a straight, flat section of road, it is not only possible to measure the vertical aero forces in a more controlled way than at a circuit, but it is also possible to estimate drag by looking at the deceleration of the car when coasting in neutral. Such a test is typically performed at an aerodrome or vehicle research station, where a straight of a mile or more may be available.

With the use of Pi beacons and special Pi Version 6 functions, it becomes possible to somewhat automate the generation of force coefficients for both lift and drag.

In this section, we look at the following:

- The run procedure
- Hardware required
- Use of user math function to filter only data taken at the correct test conditions
- Configuration of tabular outing report in Version 6 to show aero forces

Note: You will need Pi Version 6 Standard or Professional versions to run the math functions described here. It is possible, but laborious, to work with Version 6 Lite

The run procedure

It is necessary to make three sets of measurements, each involving runs in opposite directions down the straight, making a total of 6 runs. The reason for running in both directions is to average the effect of any direction-specific factors, such as headwind and gradient.

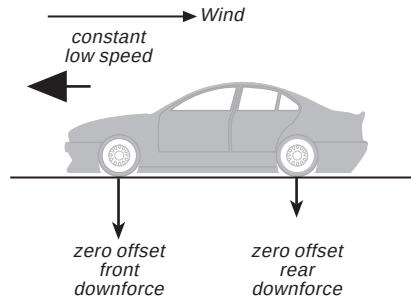
It is important that the driver tries to make the tests as repeatable as possible by sticking to a rigid set of run speeds, and by starting and stopping the procedure at the same part of the track.

The first two runs are used to establish the zero offset for the vertical forces. It is possible to measure these with the car sitting in the garage, but any 'stiction' and hysteresis in the suspension components or the lack of a perfectly flat surface will lead to erroneous readings.

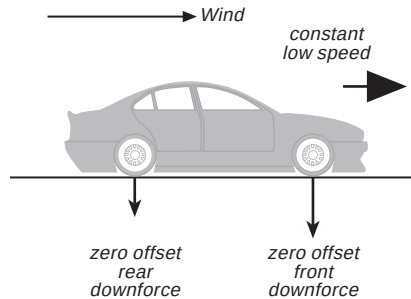
These two runs should be run at sufficiently low speed that the vertical aerodynamic effects can be considered to be negligible; say, 50kph.

It is only necessary to hold this speed for, say 20 seconds, whilst sufficient data is collected. The driver may then speed to the end of the straight and turn ready for the next run.

Runs 1 and 2 establish the zero offset for the vertical forces.

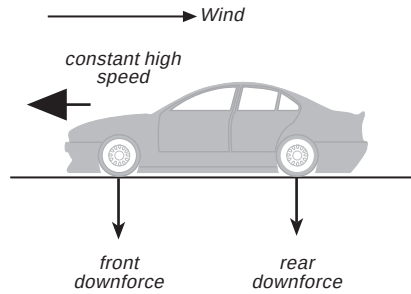


Run 1: Constant low speed in one direction

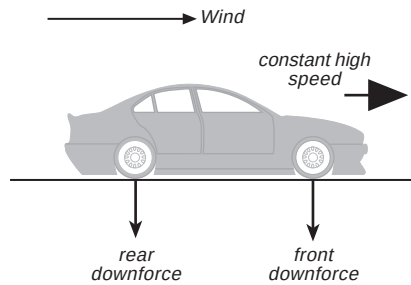


Run 2: Constant low speed in the other direction

Runs 3 and 4 establish the vertical aero forces by running at a constant high speed; say 200kph.

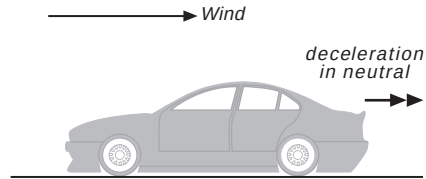


Run 3: Constant high speed in one direction

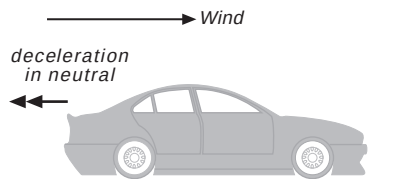


Run 4: Constant high speed in the other direction

Runs 5 and 6 establish the drag force by looking at the coastdown. The car should be taken up to above a particular speed, say 200kph, neutral engaged, and then allowed to coast down to below a particular speed, say 150kph.



Run 5: Coast down in neutral in one direction



Run 6: Coast down in neutral in the other direction

Hardware Required

Sensors

The following measurements should be made on the car:

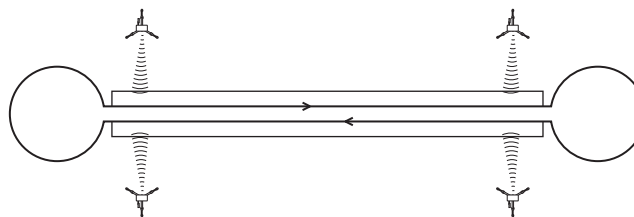
- vertical force
- dynamic pressure measured using a pitot-static tube
- speed

Vertical force may be measured using either strain gauges or potentiometers, as discussed previously. However, since the testing is essentially steady state, it may not be necessary to have the complex math functions that are used on a circuit to eliminate squat, roll effects etc..

Beacons

To implement an automated procedure, it is necessary to split up each run into a 'lap' in the Version 6 data. This can be done manually after each test by inserting beacons into the data. However, it may be simpler to place beacons at either end of the straight.

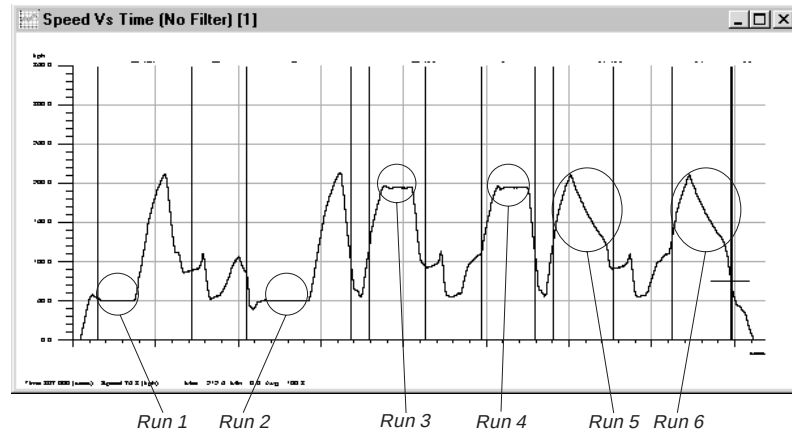
Note: Since the beacon receiver is mounted on only one side of the car, you will need a total of four beacons transmitters.



Beacon positions

Using math functions to filter the data

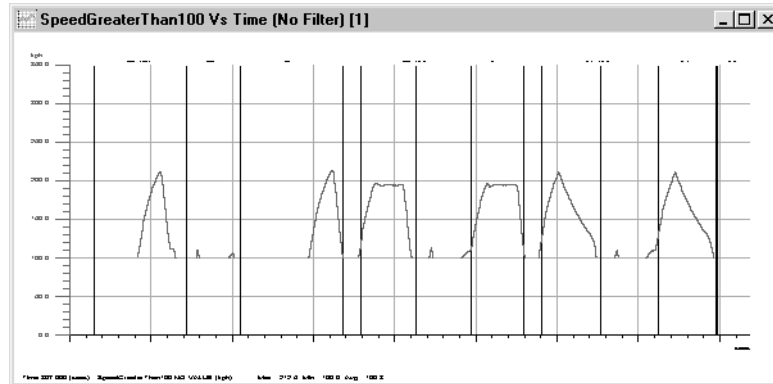
A typical dataset from a test will look as follows:



You can take the force measurements from this run and process them into a tabular outing report. However, you only want to operate on the parts of the data that fall within the correct test conditions. To do this, you make use of the GATE_TEST and COMBINE user math functions.

To illustrate the use of these math functions, consider the following math definition which creates a fragmented version of the speed trace: it takes the value of Speed when Speed is greater than 100kph, but has no value elsewhere:

UserFn (combine , UserFn (gate_test , - (Speed > 100) , - 0) , Speed)



As you can see in the above figure, the trace is chopped into fragments for Speed > 100kph only. A report containing the lap average of such a channel will be taking the average of Speed only for when Speed is greater than 100kph.

You use this technique as follows:

1. Create a channel GateLow, which only takes a value when the Speed is steady at a speed LowS.
2. Create a channel GateHigh, which only takes a value when the Speed is steady at a speed HighS.

Using GateLow and GateHigh in conjunction with the user math function COMBINE operating on measured forces and dynamic pressure will give channels that only take values at the correct conditions. These channels can then be put into tabular output reports. A dataset is supplied on the demo disk as a worked example, together with the supporting math functions.

Demo disk

GATE_TEST User math function

The file **GAT_TEST.DLL** contains the user math function GATE_TEST. This file is installed with your Version 6 application. If it has not been installed, then copy this file from the demo disk to your Version 6 working directory. Then make Version 6 “aware” of the **GAT_TEST.DLL** by editing the file **USERMATH.INI**.

Editing file **USERMATH.INI**:

1. Open the file **USERMATH.INI** (for example, double-clicking on it from Explorer will open the file using the Notepad accessory).
2. Add the line **gat_test.dll** to the file:



3. Save and close the file.

Now each time you start Version 6, it will load up the user math function GATE_TEST.

AERO1.DAT

The demo disk also includes some data in a file called **_AERO1.DAT** containing the following channels:

FrontLift	The downforce (in N) on the front axle
RearLift	The downforce (in N) on the rear axle
Speed	In kph
Dynp	Dynamic pressure (in Pa) measured using a pitot-static tube

To load

1. Copy the file **_AERO1.DAT** into a directory on your hard disk.
2. Open the file in Version 6, using the Outing button in a Data Window. The file should appear as follows:

Session:Outing:Lap	Track	Car	Driver	Comment
998:001:001	Aerodrome	PiCar	Bernoulli	Straightline test

Constant definitions

The demo disk has a file called **AEROCON1.TXT** containing constant definitions for LowS, Err LowS, ErrLowA, HighS, ErrHighS, ErrHighA and AirDensity which are used below. You can open the file in a text editor (for example Notepad) and copy and paste the text into your Version 6 constant definitions.

Math channels

The demo disk has a file called **AEROMTH1.INI** which contains the math definitions described below. To import them into Version 6, follow these steps:

1. Open the Math definitions dialogue from the Define menu.
2. Choose Import, and select the file **AEROMTH1.INI**
3. Move the channels to the right hand box using Add>>
4. Choose OK.

Processing the vertical forces

All the math functions described below are defined in the demo disk file **AEROMTH1.INI**:

```
GateLow      UserFn ( gate_test , - ( UserFn ( fabs , ( Speed - const ( LowS ) ) )
              < const ( ErrLowS ) ) * ( UserFn ( fabs , ( Inline Accel ) )
              < const ( ErrLowA ) ) , - 0 )
GateHigh     UserFn ( gate_test , - ( UserFn ( fabs , ( Speed - const ( HighS ) ) )
              < const ( ErrHighS ) ) * ( UserFn ( fabs , ( Inline Accel ) )
              < const ( ErrLowA ) ) , - 0 )
```

GateLow takes a value when the Speed is $\text{LowS} \pm \text{ErrLowS}$ and the car's inline acceleration is less than ErrLowA. Otherwise it has no value. Note that the absolute value of GateLow when the Speed channel is in the required range has no meaning. GateLow is an intermediate channel which, when used in conjunction with the user math function **combine** allows us to extract values from other channels whilst the low Speed criterion is satisfied.

Similarly, GateHigh takes a value when the Speed is $\text{HighS} \pm \text{ErrHighS}$ and the car's inline acceleration is less than ErrHighA. Otherwise it has no value. Note that the absolute value of GateHigh when the Speed channel is in the required range has no meaning. GateHigh is an intermediate channel which, when used in conjunction with the user math function **combine** allows us to extract values from other channels whilst the high Speed criterion is satisfied.

The values LowS, ErrLowS, ErrLowA, HighA, ErrHighS and ErrHighA are kept as constants so that they may easily be edited without having to access the math functions. They may therefore also be made outing dependant constants.

Given channels that represent the vertical forces FrontLift and RearLift, and also the dynamic pressure DynP, we can define the following additional math channels:

```
■ DynPLow      UserFn ( combine , GateLow , DynP )
■ DynPHigh     UserFn ( combine , GateHigh , DynP )
■ FrontLiftLow  UserFn ( combine , GateLow , FrontLift )
■ FrontLiftHigh UserFn ( combine , GateHigh , FrontLift )
■ RearLiftLow   UserFn ( combine , GateLow , RearLift )
■ RearLiftHigh  UserFn ( combine , GateHigh , RearLift )
```

Viewing these as 'lap averages' in a tabular outing report gives us a value for each of these forces and pressures at the appropriate speed. Runs for which the channel is not applicable (eg. DynPLow on a high speed pass of the straight) appear blank in the table.

Straightline Test [1]							
Lap	Time	DynPLow Lap Avg	DynPHigh Lap Avg	FrontLiftLow Lap Avg	FrontLiftHigh Lap Avg	RearLiftLow Lap Avg	RearLiftHigh Lap Avg
998:001 PiCar Bernoulli							
001	0:15.30	-----	-----	-----	-----	-----	-----
002	0:56.70	117.9	-----	8173.2	-----	8235.9	-----
003	0:32.60	-----	-----	-----	-----	-----	-----
004	1:03.40	119.6	-----	8227.1	-----	8270.0	-----
005	0:11.50	-----	-----	-----	-----	-----	-----
006	0:33.40	-----	1831.0	-----	8780.3	-----	9505.1
007	0:34.10	-----	-----	-----	-----	-----	-----
008	0:32.30	-----	1819.7	-----	8862.0	-----	10056.5
009	0:11.10	-----	-----	-----	-----	-----	-----
010	0:36.30	-----	-----	-----	-----	-----	-----
011	0:35.20	-----	-----	-----	-----	-----	-----
012	0:36.80	-----	-----	-----	-----	-----	-----
013	0:20.39	-----	-----	-----	-----	-----	-----

Tabular outing report

By transferring the values in the outing report to a spreadsheet, you may readily calculate the coefficients of lift as:

$$C_{\text{Frontlift}} = \frac{1}{2A} \times \left[\frac{\text{FrontLiftHigh}_{\text{Run6}} - \text{FrontLiftLow}_{\text{Run2}}}{\text{DynpHigh}_{\text{Run6}} - \text{DynpLow}_{\text{Run2}}} \times \frac{\text{FrontLiftHigh}_{\text{Run8}} - \text{FrontLiftLow}_{\text{Run4}}}{\text{DynpHigh}_{\text{Run8}} - \text{DynpLow}_{\text{Run4}}} \right]$$

$$C_{\text{Rearlift}} = \frac{1}{2A} \times \left[\frac{\text{RearLiftHigh}_{\text{Run6}} - \text{RearLiftLow}_{\text{Run2}}}{\text{DynpHigh}_{\text{Run6}} - \text{DynpLow}_{\text{Run2}}} \times \frac{\text{RearLiftHigh}_{\text{Run8}} - \text{RearLiftLow}_{\text{Run4}}}{\text{DynpHigh}_{\text{Run8}} - \text{DynpLow}_{\text{Run4}}} \right]$$

Note that these equations assume SI units (i.e. Newtons for forces, Pascals for dynamic pressure, ms⁻² for A). If you're using alternative units, see the conversion factors in the Appendix.

Processing the drag force

Now, the forces acting on the car in the coastdown are:

- aerodynamic drag, which we can characterise as $C_D A \times \text{Dynp}$
- rolling resistance, which can be approximated across our speed range as a constant R

[R may be determined experimentally by towing the car inside a windshield and measuring the tension in the towbar.]

If the mass of the car is M (in kg), then from Newton's 2nd Law we have:

$$M \times (\text{In-Line Accel}) = -(C_D A \times \text{Dynp} + R)$$

If the duration of the coastdown is T, during which time the car's speed drops from CoastHigh to CoastLow, then integrating the above equation over the duration of the coastdown gives us:

$$M \times (\text{CoastHigh} - \text{CoastLow}) = \int_{\text{coastdown}} (C_D A \times \text{Dynp} + R)$$

or

$$C_D = \frac{M \times (\text{CoastHigh} - \text{CoastLow}) - R \times T}{A \times \int_{\text{coastdown}} \text{Dynp}}$$

To make this calculation, we need to extract the duration of the coastdown and a trace of dynamic pressure. Create the following math channels in Version 6:

- $\text{CoastTrue} = (\text{Speed} > \text{const}(\text{CoastLow S})) * (\text{Speed} < \text{const}(\text{CoastHighS})) * (\text{Inline Accel} < \text{const}(\text{ErrCoastA}))$
- $\text{CoastTime} = \text{Integ}(\text{CoastTrue} * \text{Elapsed Time})$
- $\text{CoastIntegDynP} = \text{Integ}(\text{CoastTrue} * \text{Dynp})$

Include these three channels as 'lap max' in a tabular outing report.

Coastdown [1]			
Lap	Time	CoastTime Lap Max	CoastIntegDynP Lap Max
998:001 PiCar Bernoulli			
001	0:15.30	0	0
002	0:56.70	127	4857
003	0:32.60	0	0
004	1:03.40	184	4986
005	0:11.50	0	0
006	0:33.40	284	26838
007	0:34.10	0	0
008	0:32.30	330	27642
009	0:11.10	0	0
010	0:36.30	222	19348
011	0:35.20	0	0
012	0:36.80	226	19523
013	0:20.39	0	0

Tabular Lap Report Showing values for CoastTime and CoastIntegDynP

Again, by transferring values in the outing report to a spreadsheet, you may readily calculate the coefficient of drag as:

$$C_D = \frac{1}{2A} \times \left[\frac{M \times (\text{CoastHigh} - \text{CoastLow}) - R \times \text{CoastTime}_{\text{Run10}}}{\text{CoastIntegDynP}_{\text{Run10}}} + \frac{M \times (\text{CoastHigh} - \text{CoastLow}) - R \times \text{CoastTime}_{\text{Run12}}}{\text{CoastIntegDynP}_{\text{Run12}}} \right]$$

Note that in this calculation, the Tabular Outing Report contains irrelevant values for CoastTime and CoastIntegDynP for the constant speed passes. It is important to select only the values for the two coast down tests. In this case laps 10 and 12.

Pressure measurements

The airflow around the car creates pressures on a car's surfaces. This section covers how the measurement of these pressures contributes to the analysis of the car:

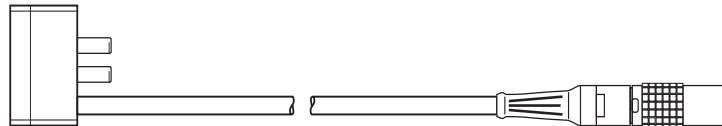
- estimation of wing forces and stall conditions
- air intake efficiency
- radiator efficiency

Pi's aero pressure sensors which have been designed specifically for this task are covered first.

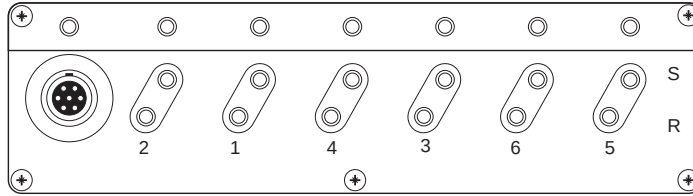
Pi aero pressure sensors

Pi aero pressure sensors are two part differential sensors whose output is therefore proportional to the difference in the air pressures applied to the two ports. This makes them ideally suited to the direct measurement of aero pressures referenced to a free-stream static pressure.

The sensing element used in the sensors provides a differential measurement range of ± 2 PSI. This is offered in two packages: a single aero sensor, and a hex aero sensor.

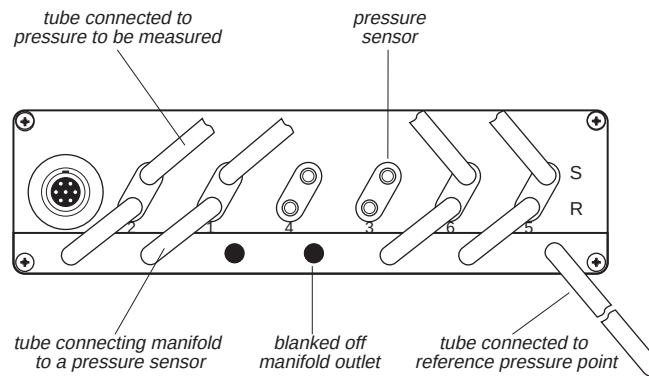


Single aero sensor



Hex aero sensor

A manifold accessory is also available for the hex aero sensor, so that all the reference ports can be commoned together. This is useful when looking at surface pressures, as you need to use static pressure (from the pitot-static tube) as a common reference.

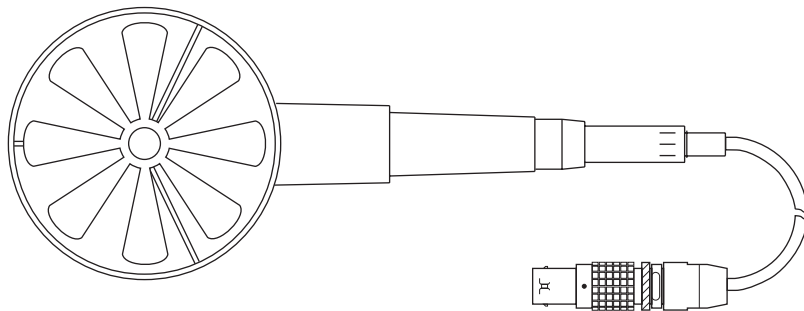


Hex aero sensor with manifold fitted

Pi Air flow sensor

The amount of cooling that a radiator can deliver will be approximately proportional to the speed of the air passing through the radiator. (Strictly speaking, it is a function of the mass flow of the air and the difference in temperature between the air and the water, but for a fixed set of conditions, our approximation holds).

When investigating the aerodynamic efficiency of the radiator inlets, it is therefore preferable to measure the speed of the air through the radiator. Pi have a special air flow sensor specifically for this task. It is a wind vane device that can be placed in the radiator ducts, and can measure between 0.5—20 m/s.



Airspeed sensor 0.5 to 20 m/s

The speed sensing element is a lightweight vane mounted in low friction bearings.

The sensor is supplied with a short stem with a one metre cable fitted with a 4-pin Lemo connector to connect to an Air speed amplifier. The Air speed amplifier provides signal conditioning between the air speed signal and a digital input of a Pi System.

Airspeed sensor calibration information

To make use of the information from the Air speed sensor, you first have to calibrate a digital channel.

To calibrate a digital channel for use with the Air speed sensor:

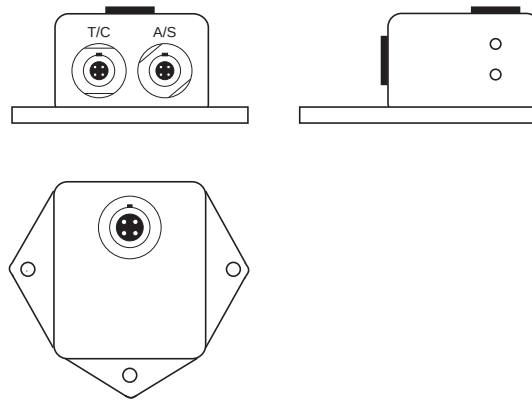
1. Create a channel called 'air speed' in Version 6 Setup.
2. Set the number of pulses for the digital channel to 28 per half revolution.
3. Set a wheel diameter of 198mm.

Graphing this channel will produce a trace of air speed in MPH or KPH depending upon whether you are have metric or imperial software. To show air speed in other units you need to create a maths channel to perform a conversion calculation.

To convert air speed in KPH to air speed in m/s the calculation is: $(\text{air speed})/3.6$.

To convert air speed in MPH to air speed in ft/sec the calculation is: $(\text{air speed}) \times 1.467$.

Air speed amplifier



The Pi Air speed amplifier is needed to connect a Pi air speed sensor to a digital channel of a Pi system. The air speed output is connected to a digital channel using a suitable loom.

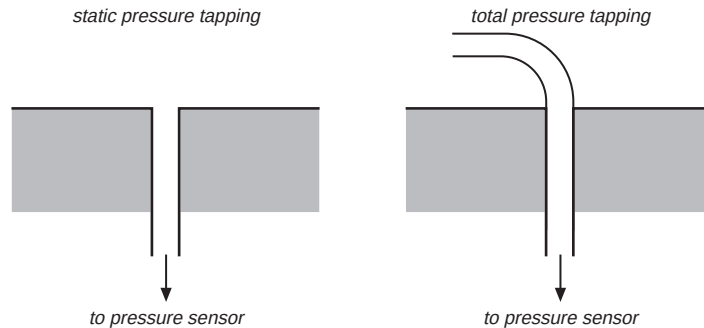
An air speed sensor can be plugged directly into the amplifier or the amplifier may be mounted remotely from the air speed sensor using a sensor extension loom.

Some of the Pi range of air speed sensors have a thermocouple to measure air temperature. The temperature signal passes through the amplifier and must be connected to a thermocouple amplifier or 12-channel junction box using a suitable loom.

Surface pressures

Measurements of surface pressures give a good indication of the performance and efficiency of aerodynamic surfaces such as the wings and engine covers on a formula car. However, such measurements require a great deal of preparation and will only be of real benefit when the data collected can be compared with similar readings from wind tunnel tests.

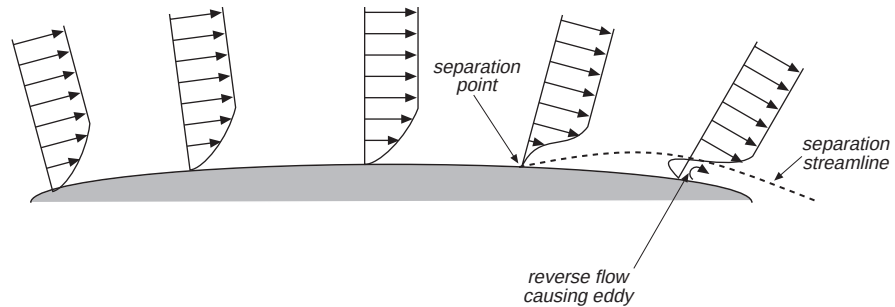
The Pi six channel aero sensor is ideally suited to measuring surface pressures. The six measurement ports are connected to pressure tapings on the aerodynamic surface under test and the reference ports should be connected to a common free stream static pressure from a pitot-static tube. Depending on the configuration of the pressure tapping (see sketch below), this will give either the variation in static pressure from the free stream value, or the variation in dynamic pressure, across the aerodynamic surface in question. From this we can get an idea of the amount of pressure recovery across the surface, in the direction of flow, and a corresponding indication of its efficiency.



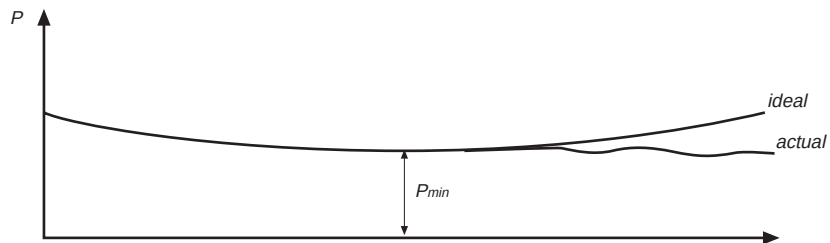
Static pressure tapping and total pressure tapping configurations

Analysis

The static pressure distribution over a surface gives us a good idea of its efficiency. It will indicate whether or not the flow over the surface is attached over the entire surface, or whether part of the flow is separated. Separation occurs when the increase in local static pressure along the direction of the flow becomes such that the flow is actually reversed at the surface and the boundary layer can no longer follow the surface profile. This is indicated in the figure below.



Velocity profile over a surface showing a separation point of the flow



Pressure distribution over surface

When separation occurs, a turbulent wake is formed downstream of the point of separation and this greatly increases the drag caused by the surface. As a consequence lift (or downforce) generated will decrease as there is less distance along the surface for which there is a net pressure downwards.

As the angle of attack of a wing for example is increased, the point of separation will move forward, the turbulent wake will become wider and there is a marked increase in drag and a corresponding decrease in lift. The point at which flow is almost entirely separated from the low pressure side of the aerodynamic surface (the *lower* surface on the wing of a racing car) is known as stall. For obvious reasons stall should be avoided as far as possible!

If an aerodynamic surface is tapped in order to measure the local static pressure distribution across it, this will show an increase in the direction of the flow when the surface is operating efficiently, as pressure recovery occurs, resulting in the generation of net downforce. When the flow over a surface has separated, the static pressure measurements will no longer increase from the point of separation.

Internal flows

A number of on-car systems require air such as engine cooling (including oil cooling), the engine air intake, brake ducts etc. There are therefore a range of complex internal air flows on a typical race car as well as the more noticeable external flows.

Internal flows typically perform a particular task such as engine aspiration and so their efficiency will have a direct impact on the car's performance. However internal air flow results in aerodynamic drag in the same way as external air flow does, and so the efficiency of the internal flows also has a significant effect on the overall aerodynamic efficiency of the car.

Clearly a detailed analysis of the design of a race car's internal ducting is beyond the scope of this Application Note. The discussion has therefore been limited to a brief review of the measurements that can be made to give an idea of the radiator and air intake efficiency.

Radiator efficiency

Radiator efficiency is typically characterised by three parameters:

Radiator pressure drop coefficient

$$K_P = \frac{\Delta P}{\frac{1}{2} \rho V_R^2}$$

where ΔP is the static pressure drop (front to rear) across the radiator
 V_R is the velocity of the air at the radiator face

Velocity ratio

$$K_V = \frac{V_R}{V_\infty}$$

where V_R is the velocity of the air at the radiator face
 V_∞ is the free stream velocity of the air

Drag coefficient of the radiator

$$C_D = \frac{\text{Drag of radiator}}{\frac{1}{2} \rho V_\infty^2 A}$$

where V_∞ is the free stream velocity of the air
 A is the area of the radiator

The most useful analysis of radiator efficiency will be a comparison with the values for these three parameters predicted by wind tunnel studies. However for the purposes of our simple investigation, suffice it to say that a general 'rule of thumb' approach is that the best cooling performance is produced by a large pressure drop coefficient (K_P) and a large velocity ratio (K_V). These parameters can easily be determined experimentally by:

- Using the Pi air speed sensor to determine the air velocity at the radiator face and the Pi pitot-static tube (in conjunction with a Pi aero sensor) to determine the free stream air velocity.
- Using pitot tubes (in conjunction with Pi aero sensors) both in front of and behind the radiator core to determine the pressure drop across it.

Air intake efficiency

The sensors and ideas for measuring pressure distributions and speeds of air through internal ducts, as described above, can easily be adapted to determine the characteristics of the flow of air into an engine airbox.

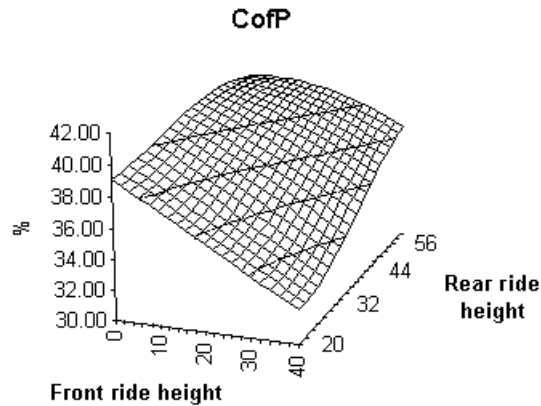
As an example, the airbox on an open wheel racing car such as a Formula One car, will typically be of the 'ram air effect' design. The features of this design should be such that;

- There is sufficient air flow to meet the needs of the engine at low speed, when ram effect is small.
- It is of a suitable shape to convert velocity to pressure (by allowing expansion of the flow) whilst avoiding separation along the internal surfaces and generating a uniform pressure distribution across the inlet trumpets.

Confirmation of the design and on-car performance of the airbox is therefore possible by measuring the pressure distribution across the inlet trumpets using a pitot tube. An efficiently operating airbox will yield an even pressure distribution across the trumpets. Anomalies such as separation will show up as lower pressure areas where the pitot tube is in the region of the wake, downstream of the point of separation.

Using aero maps (2D table lookup)

Aerodynamic characteristics are usually dependent on the ride height of the car. This means that, for example, the centre of pressure is not given by a single number, but by a map covering the complete range of ride heights.



To calculate the centre of pressure at a point on the circuit, it is necessary to look up its value in the map for the instantaneous ride height:

$$\text{CofP} = F(\text{Front ride height}, \text{Rear ride height})$$

Front ride height and Rear ride height are the indices used to pinpoint values in the table CofPTable.

This section describes how to use the Version 6 lookup table facilities, using an aero map as an example. The lookup table method can of course be used for other purposes - for example, an engine dyno map can provide an estimate of torque using the current engine RPM and throttle position as indices:

$$\text{Torque} = \text{Torque}(\text{RPM}, \text{throttle})$$

Note: To use the lookup table facility, you will need a PC that has a copy of Pi Version 6 and a copy of Microsoft Excel™.

Setting up the worked example

The worked example provided on the accompanying demo disk has the following components:

- an aero map in an Excel spreadsheet
- a DLL containing the user math functions for the 2D table lookup
- a Version 6 constant definition, which provides the connection between Version 6 and the aero map
- track data, featuring front and rear ride heights
- a Version 6 math definition, which uses the aero map data

Information on how to set up each of these components is given below.

Aero map

The demo disk has a file **AERO.XLS**. Copy this into a working directory on your hard disk. The spreadsheet has three worksheets, containing respectively aero map tables for Centre of Pressure, Lift and Drag.

A table to be used by Version 6 must have the following structure:

- The top row should contain the values of the row index: X1, X2, X3 etc..
- These must be monotonically increasing: i.e. $X1 < X2 < X3$ etc..
- The left column should contain the values of the column index: Y1, Y2, Y3 etc.
- These must be monotonically increasing: i.e. $Y1 < Y2 < Y3$ etc..
- The “data area” of the table should be completely filled, with no missing data: D11, D21, D31, etc.

	Row index
Column index	Data area

e.g

	X1	X2	X3
Y1	D11	D21	D31
Y2	D12	D22	D32
Y3	D13	D23	D33

Installing D2LUT.DLL

The file **D2LUT.DLL** contains the user math functions for the 2D table lookup. First, copy this file from the demo disk to your Version 6 working directory.

Then make Version 6 “aware” of the DLL by editing the file **USERMATH.INI**.

To edit USERMATH.INI

1. Open the file **USERMATH.INI** (for example, double-clicking on it from Explorer will open the file using the Notepad accessory).
2. Add the line **d2lut.dll** to the file.



3. Save and close the file.

Now each time you start Version 6, it will load up the user math functions in D2LUT.DLL.

Version 6 constant definition of the aero map

Version 6 can only use data from an Excel spreadsheet table if it knows how to find it. This is done by defining the spreadsheet table as a Version 6 Constant. For more details on defining constants see the “Constants” section in *Version 6 PC Software Guide*.

To set a constant definition of the aero map:

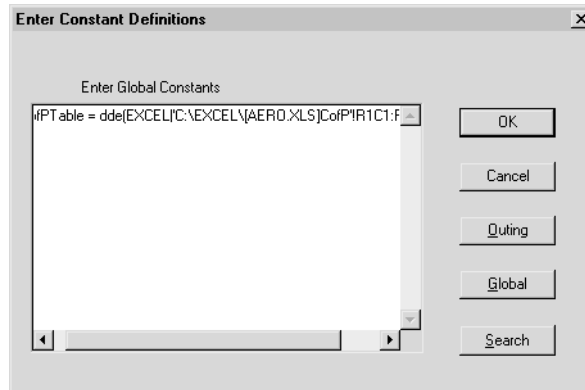
1. Choose “Constant Definitions...” from the Define menu. The Enter Constant Definitions dialog appears.
2. Create a new constant using the following syntax:
`<name> = dde(EXCEL|'<path>[<filename>]<sheetname>'RmCn:RxCy)`

where

<name>	is the name that you want to give the table for use in Version 6
<path>	is the directory containing the Excel spreadsheet
<filename>	is the name of Excel spreadsheet
<sheet>	is the name of the Excel sheet containing the table
m and n	are the row and column numbers of the top left hand corner of the table (including the row and column containing the index values)
x and y	are the row and column numbers of the bottom right hand corner of the table

For example, a constant which links to the CofP table in the demo file **AERO.XLS** would look as follows:

CofPTable = dde(EXCEL|'C:\EXCEL[AERO.XLS]CofP'R1C1:R22C22)

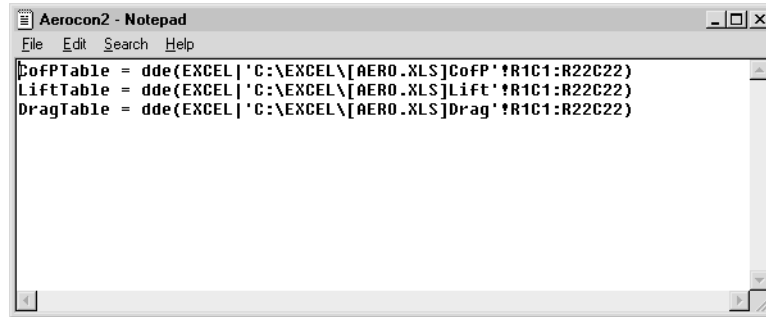


Enter Constant Definitions dialog

The demo disk has a file called **AEROCON2.TXT**. This contains definitions for CofPTable, LiftTable, and DragTable which you can copy and paste into the Version 6 Constant Definitions. The definitions assume that you have copied **AERO.XLS** into a directory called **C:\EXCEL**. Edit this directory path if it is incorrect.

To set a constant definition of the aero map using copy and paste:

1. Choose "Constant Definitions..." from the Define menu. The Enter Constant Definitions dialog appears.
2. Choose the global or outing button (see the Note below).
3. Open the file **AEROCON2.TXT** (for example, double-clicking on it from Explorer will open the file using the Notepad accessory).



4. Select a line of text and chose “Copy” from the Edit menu.

For example:

CofPTable = dde(EXCEL|'C:\EXCEL[AERO\$session.XLS]CofP'R1C1:R22C22) would locate the table in a file called **AER0000.XLS**, **AER0001.XLS** etc_ for sessions 000, 001 etc.

5. Close the **AEROCON2.TXT** file and return to Version 6.
6. Place the cursor in the freeform text box of Enter Constant Definitions dialog and paste the text using **[Ctrl] + [V]** keys.

Note: You can choose to make a definition 'global' or 'outing dependent', depending on whether the data should change from outing to outing. In the above example, the Excel spreadsheet table has been defined as a global constant, which would be appropriate if the car's aerodynamics were fixed. However, if you are switching between different aerodynamic configurations, and have an aero table to support each configuration, then you may want to define the table as an outing dependent constant.

Track data

You will need to define a math channel that makes use of the Excel data. However, before doing this you need to identify the channels which will be used as the indices. The demo disk contains a file which has Version 6 data with appropriate channels.

The file is called **_AER02.DAT**, and contains data from a fictitious car running around Silverstone GP circuit. The data was actually generated using a simulation, so it is unrealistically smooth. However, the important thing for the sake of the example is that it contains the following channels:

F Ride Height	The front ride height of the car, measured in mm
R Ride Height	The rear ride height of the car, measured in mm
Speed	In kph

On a real car, you would ideally use Pi's Laser Ride Height System to create the ride height channels.

The front and rear ride heights will be used as indices for looking up data in the Excel spreadsheet. Speed will be used for generating an estimate of dynamic pressure.

Copy the file **_AER02.DAT** into a directory on your hard disk.

Open the file in Version 6, using the Outing button on a Data Window. The file should appear as follows:

Session:Outing:Lap	Track	Car	Driver	Comment
999:001:001	Silverstone	PiCar	Bernoulli	Aero demo

Version 6 math channel definitions of "CofP"

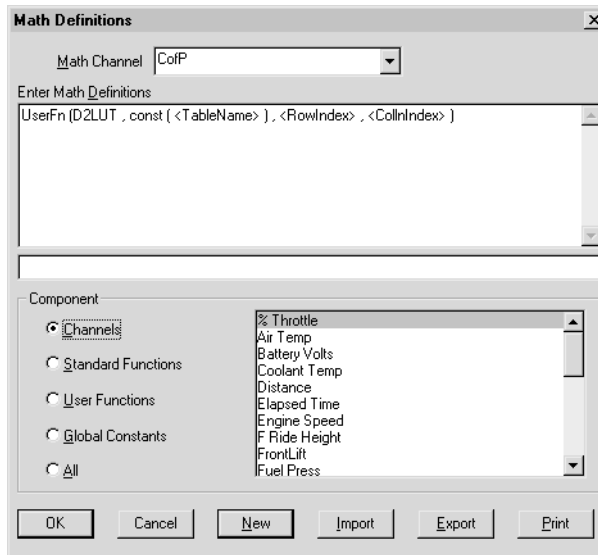
You are now in a position to define the channel which uses the data in Excel™. This is done using a Version 6 math channel.

You can either type the math channel definitions or you can import definitions contained in a file called **AEROMTH2.INI** on the demo disk. This file contains math definitions for DynP, CofP, Lift, and Drag which you can import directly into Version 6.

Both methods are described below.

To define the CofP math channel:

1. Choose “Math Definitions...” from the Define menu. The Maths Definitions dialog appears.



Math Definitions dialog

2. Create a new math channel using the following syntax:
UserFn (D2LUT , const (<TableName>) , <RowIndex> , <ColIndex>)
where:

D2LUT is the User Function which performs the table lookup

<TableName> is the Global constant defining your Excel table area

<RowIndex> is the Channel to be used as the row index

<ColIndex> is the Channel to be used as the column index

For example, define CofP as follows:

UserFn (D2LUT , const (CofPTable) , F Ride Height , R Ride Height)

To import the math channel definitions:

1. Choose "Math Definitions..." from the Define menu. The Math Definitions dialog appears.
2. Choose "Import", and select the file **AEROMTH2.INI**.
3. Move the channels DynP, CofP, Lift, and Drag to the right hand box using the Add>> button.
4. Choose OK.

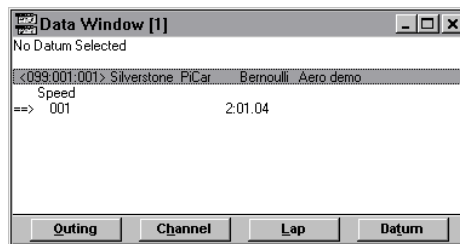
*Note: If you have previously tried the first worked example in this Application Note, you will already have a channel named **DynP**. You will need to delete this channel using the 'Scalings' option from the Define menu before V6 Analysis will let you import the math channel named **DynP**. Refer to the **Modifying graphical views** section in the Version 6 PC Software Guide for details on how to delete a channel.*

Putting it all together

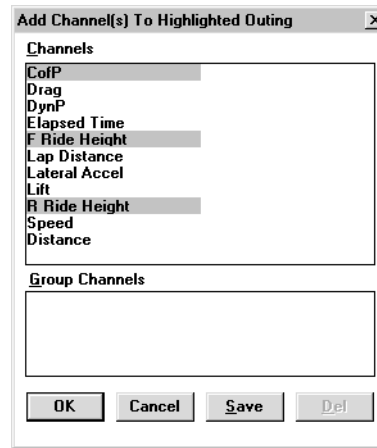
You have now prepared everything required to display a lookup channel.

To display the channel "CofP":

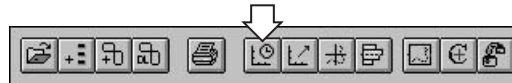
1. Make sure that Excel is running and that the file **AERO.XLS** is open.
2. From Version 6, open the demo data file in a Data window.



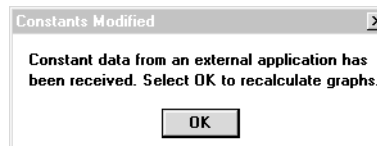
3. Select the Channels CofP, F Ride Height and R Ride Height.



4. Select the Time Plot button on the toolbar (or select "Time Plot" from the View menu).

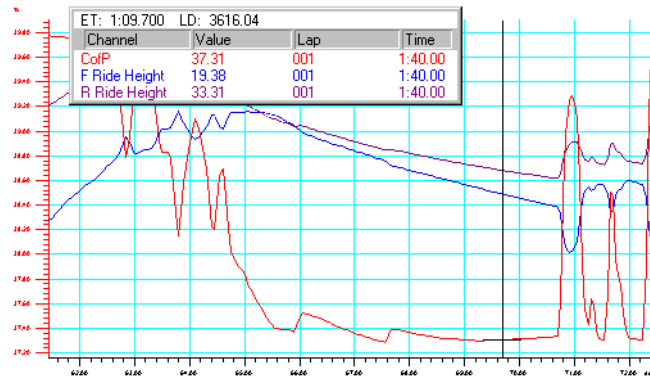


The Constants Modified dialog should appear, showing that CofPTable has successfully linked to the Excel spreadsheet.

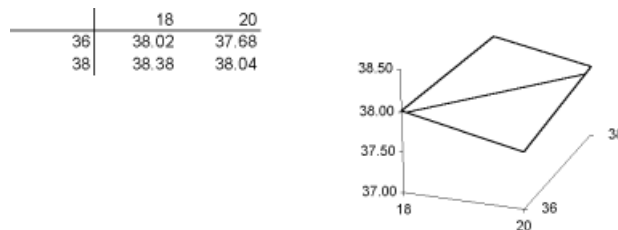


5. Select OK.

The graph will show the two index channels and the resultant CofP channel.



If you note the value of CoP (in the above figure, 37.31), then this will relate to the table location provided by the two indices (F Ride Height = 19.38, R Ride Height = 33.31).



Linear interpolation between adjacent points to give the resultant value of CoP

In this example the math channel CoP is simply reading the values from the CoP table based on the lookup indices of front ride height and rear ride height. This gives us a good qualitative indication of the variation of the position of the centre of pressure around the track and lets us pick out any abnormality in the aerodynamic balance of the car.

For a more quantitative analysis we might want to calculate the lift and drag forces around the circuit. This information is provided by the other math channels found in **AEROMTH2.INI**; DynP, Lift and Drag. First it is necessary to define several new constants

To define new constants:

1. Choose "Constant Definitions..." from the Define menu. The Enter Constant Definitions dialog appears.
2. Place the cursor in the freeform text box of the Enter Constant Definitions dialog and enter the following constants;

CarRefArea = 1.2

AirDensity = 1.225

KPHperMPS = 3.6

3. Close the Enter Constant Definitions dialog box.

These additional constants allow us to estimate dynamic pressure from car speed (using the math channel DynP). From this the math channels Lift and Drag estimate the lift and drag forces around the circuit extracting the values of C_D and C_L from the aero maps provided in **AERO.XLS** using front ride height and rear ride height as indices.

Once these additional constants have been defined the channels DynP, Lift and Drag will appear in the channel list for the example dataset and graphs of these channels can be viewed as normal.

Out-of-range indices

Ensure that the range of the indices in your spreadsheet map covers the range to be found in your logged data. If the data tries to look up an index that is off the edge of the map, then it will return the value zero.

An alternative is to 'gate' the index channels to the range of the map. The GATE_TEST user math function is described in the section on straightline testing.

For example, to confine the data to a front ride height range of 0-40mm and a rear ride height range of 0-60mm, you could use the following function:

```
FRHGated = F Ride Height * UserFn ( combine , UserFn ( gate_test ,  
    - ( ( F Ride Height > 0 ) * ( F Ride Height < 40 ) , -0 ) , 1 )  
RRHGated = R Ride Height * UserFn ( combine , UserFn ( gate_test ,  
    - ( ( R Ride Height > 0 ) * ( R Ride Height < 60 ) ) , -0 ) , 1 )
```

You would then look up in the table as follows:

```
UserFn ( D2LUT , const ( CofPTable ) , FRHGated , RRHGated )
```

Sensor installation notes

Air speed sensor

When installing the Air speed sensor:

- protect the sensor from vibration;
- support the sensor head to prevent damage;
- make sure that the air flow to the vane is in the direction of the arrow on top of the sensor and is not obstructed by body panels and parts of the chassis;
- select a position where the sensor will not be in constant contact with water, fuel or oil;
- make sure that the sensor will not be affected by heat soak;
- try not to place the sensor near sources of interference i.e. ignition coils, plug leads, ECMs, alternators and telemetry antenna.

Air speed amplifier

When installing the Air speed amplifier:

- select a position where the amplifier will not be in constant contact with water, fuel or oil;
- make sure that the amplifier will not be affected by heat soak;
- make sure that air can flow over the amplifier to keep it below 60°C;
- try not to place the amplifier near sources of interference i.e. ignition coils, plug leads, ECMs, alternators and telemetry antenna.

Single aero sensor

When installing a Single aero sensor:

- select a position where the sensor will not be in constant contact with water, fuel or oil;
- protect the sensor from vibration;
- make sure that the sensor will not be affected by heat soak;
- make sure that air can flow over the sensor to keep it below 60°C;
- try not to place the sensor near sources of interference i.e. ignition coils, plug leads, ECMs, alternators and telemetry antenna.

Hex aero sensor

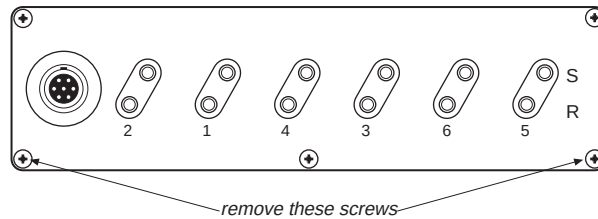
When installing the Hex aero sensor:

- select a position where the sensor will not be in constant contact with water, fuel or oil;
- protect the sensor from vibration;
- make sure that the sensor will not be affected by heat soak;
- make sure that air can flow over the sensor to keep it below 60°C;
- try not to place the sensor near sources of interference i.e. ignition coils, plug leads, ECMs, alternators and telemetry antenna.

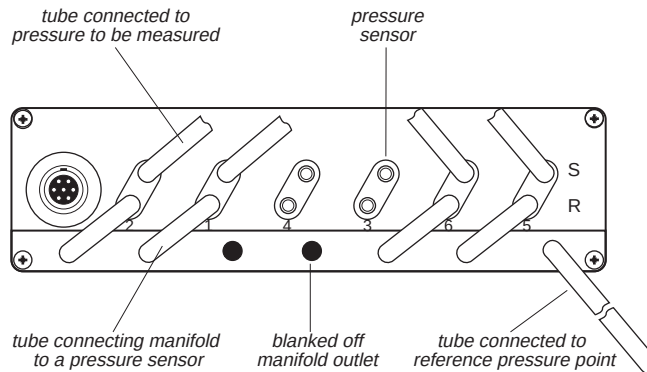
Hex aero sensor manifold

To install the hex aero sensor manifold:

1. Remove the two screws at the bottom edge of the hex aero sensor.



2. Fasten the manifold onto the sensor using the two screws supplied with the manifold kit.
3. Connect the tube from the common pressure point to one of the ports on the manifold.
4. Connect the other ports on the manifold as required to the individual pressure sensors using sections cut from the plastic tubing supplied.
5. Any ports not used on the manifold must be blanked off.



Sensor specifications

Air speed sensor

Connector details

Pin	Function	Pin	Function
1	A/S +ve signal	2	no connection
3	no connection	4	A/S –ve signal



Sensor connector	Mating connector
FGG 1B 304	EEG 1B 304

Specification details

Description	Value
Air speed (m/s)	0.25–20
Output voltage	AC
Environmental	IP65

Air speed amplifier

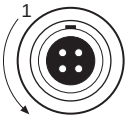
Air speed amplifier connector details

Air speed amplifier sensor input connector

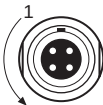
Pin	Function	Pin	Function
1	Air speed +ve	3	Temperature –ve
2	Temperature +ve	4	Air speed –ve

Air speed amplifier T/C and A/S connectors

Pin	T/C	Pin	A/S
1	Temperature +ve	1	no connection
2	Temperature –ve	2	Ground
3	no connection	3	+5V
4	no connection	4	Air speed signal



Sensor input



T/C and A/S

Connector	Mating connector
EEG 1B 304	FGG 1B 304
EEG 0B 304	FGG 0B 304

Air speed amplifier specification details

Description	Value
Supply voltage	+5V DC
Supply current	10mA
Measurement ranges	0.25–60 m/s
Operating temperature range	+10°C to +60°C
Weight	65 grams
Environmental	IP65

Single aero sensor

Single aero sensor connector details

Pin	Function	Pin	Function
1	12V at 15mA	3	no connection
2	0V	4	sensor output



Sensor connector	Mating connector
PHG 0B 304	FGG 0B 304

Single aero sensor calibration information

Pressure	Output (10-bit ADC)	Output (12-bit ADC)
–2 PSI	0.05V (10 counts)	0.05V (41 counts)
0 PSI	2.50V (511 counts)	2.50V (2047 counts)
+2 PSI	4.98V (1019 counts)	4.98V (4079 counts)

Single aero sensor specification details

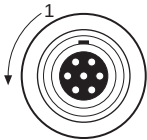
Description	Value
Supply voltage	10–16V DC
Supply current	15mA
Output voltage range	0–5V
Measurement range	±2 PSI
Overload (x rated pressure)	3
Accuracy	±3% at 25°C
Zero error at 25°C	±0.5% F.S.
Zero error +10°C to +60°C	±0.5% F.S.
Gain error at 25°C	±0.5%
Gain drift +10°C to +60°C	±0.5%
Calibration	1.25V/PSI
Operating temperature range	+10°C to +60°C
Weight	30 grams
Environmental	IP65

Specifications are typical values.

Hex aero sensor

Hex aero sensor connector details

Pin	Function	Pin	Function
1	Ground sense	5	Analogue 6
2	0V	6	Analogue 7
3	Multiplex signal	7	12V at 40mA
4	Analogue 5		



Sensor connector	Mating connector
EHG 1B 307	FGG 1B 307

Hex aero sensor calibration information

Pressure	Output (10-bit ADC)	Output (12-bit ADC)
-2 PSI	0.05V (10 counts)	0.05V (41 counts)
0 PSI	2.50V (511 counts)	2.50V (2047 counts)
+2 PSI	4.98V (1019 counts)	4.98V (4079 counts)

Hex aero sensor channel names

The Aero sensor can be connected in place of a Pi Expansion junction box and the channels are labelled by Pi software.

Sensor	Hardware index
1	E1
2	E2
3	E3
4	E4
5	E5
6	E6

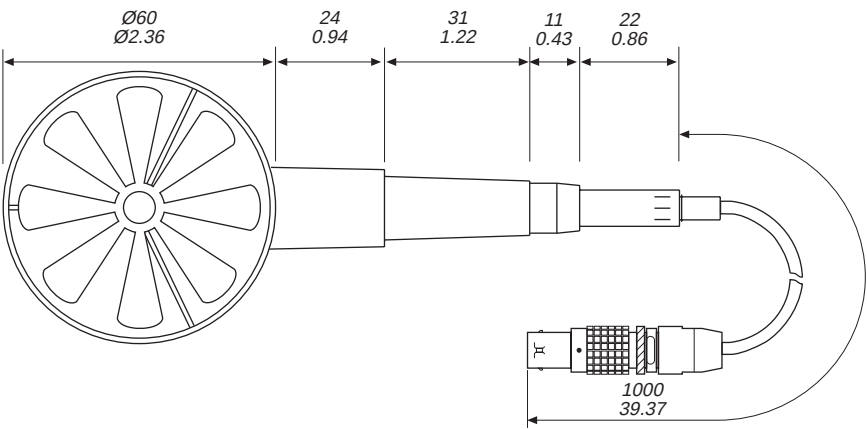
Hex aero sensor specification details

Description	Value
Supply voltage	10–16V DC
Supply current	40mA
Output voltage range	0–5V
Measurement range	±2 PSI
Overload (x rated pressure)	3
Accuracy	±3% at 25°C
Zero error at 25°C	±0.5% F.S.
Zero error +10°C to +60°C	±0.5% F.S.
Gain error at 25°C	±0.5%
Gain drift +10°C to +60°C	±0.5%
Calibration	1.25V/PSI
Operating temperature range	+10°C to +60°C
Weight	70 grams
Environmental	IP65

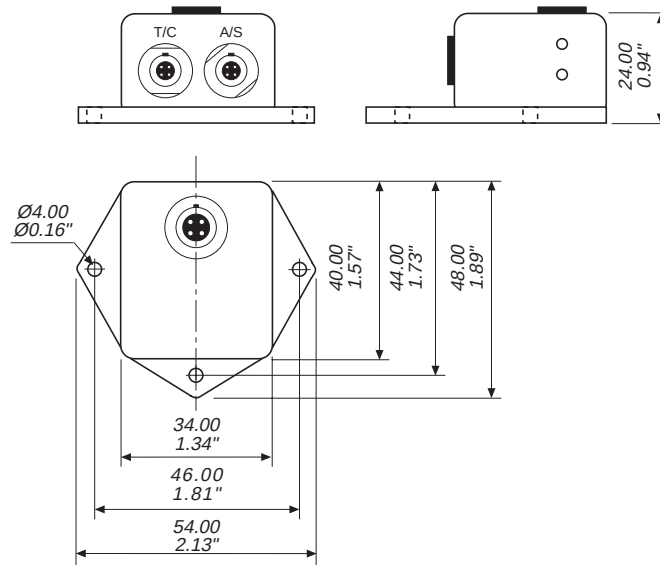
Specifications are typical values.

Sensor dimensions

Air speed sensor

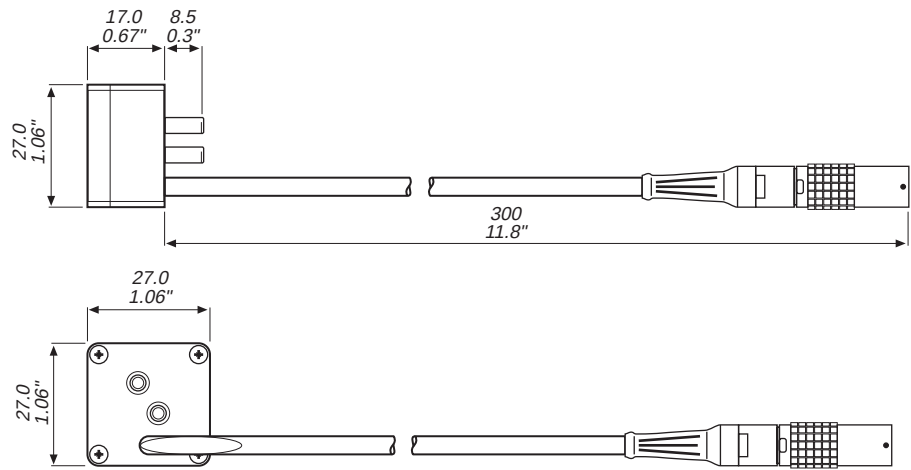


Dimensions in millimetres and inches



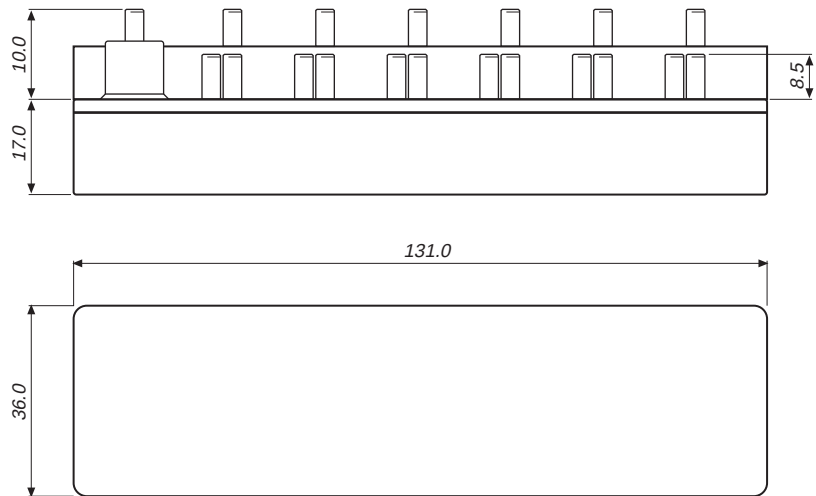
Dimensions in millimetres and inches

Single aero sensor



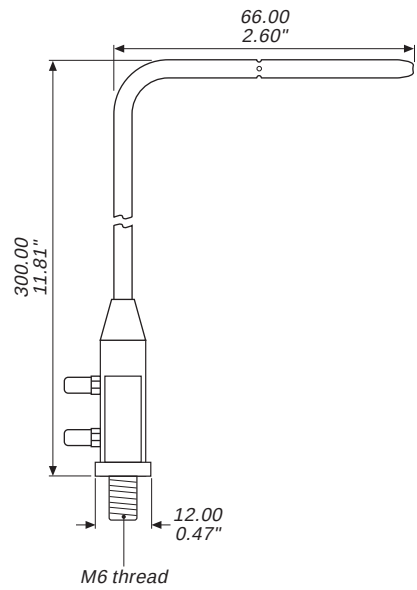
Dimensions in millimetres and inches

Hex aero sensor



Dimensions in millimetres. (Shown with hex manifold fitted)

Pi Pitot-static tube



Dimensions in millimetres and inches

Sensor ordering information

Product	Part number
Air speed sensor (with 60mm flying lead)	21A-0070
Air speed amplifier	01A-050278
Air speed sensor extension loom	03I-1236
Single aero sensor	01B-050215
Single aero sensor connection loom	03E-051
Hex aero sensor	01B-044112
Hex aero sensor connection loom	03E-007
Hex aero manifold kit	30B-044202
Pitot-static tube	01B-050245

Appendix

Conversion factors

Pressure

Value	Pa	mmH ₂ O	inH ₂ O	inHg	PSI	ATM
1 Pa	1	0.10207	0.004019	0.0002953	0.0001450	0.00000987
1 mmH ₂ O	9.7971	1	0.03937	0.002893	0.001421	0.00009669
1 inH ₂ O	248.82	25.4	1	0.07348	0.03609	0.002456
1 inHg	3386.39	345.66	13.61	1	0.4912	0.03342
1 PSI	6896.55	703.77	27.71	2.036	1	0.06805
1 ATM	101317.1	10342	407.17	29.922	14.695	1

Force

Value		newtons	lbf	kgf
1 newton (N) ≡		1	0.22481	0.10197
1 lbf ≡		4.4483	1	0.45360
1 kgf ≡		9.80665	2.2046	1

Speed

Value		m/s	kph	mph
1 m/s ≡		1	3.6	2.237
1 kph ≡		0.2778	1	0.6214
1 mph ≡		0.4469	1.609	1

Temperature

Value		K	°C	°F
Kelvin (K)	≡	K	°C + 273.15	5/9 (°F + 459.67)
Celsius (°C)	≡	K – 273.15	°C	5/9 (°F – 32)
Fahrenheit (°F)	≡	9/5 K – 459.67	9/5 °C + 32	°F

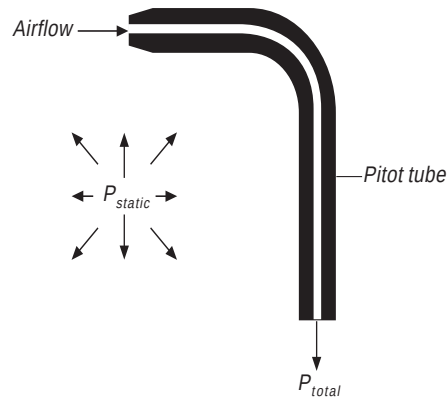
Area

Value		m2	sq. feet	sq. inches	sq. mm
1 sq. metre (m2)	≡	1	10.764	1550.0	1000000
1 sq. foot	≡	0.09290	1	144	92903
1 sq. inch	≡	0.0006452	0.006944	1	645.16
1 sq. mm	≡	0.000001	0.00001076	0.001550	1

Pitot tubes

A pitot tube is simply an open ended tube facing into the airflow, such that it sees the pressure due to the flow. However, an absolute pressure sensor attached to a pitot tube will not measure dynamic pressure P_{dynamic} directly, as it is superimposed on the static atmospheric pressure P_{static} to give the total pressure, P_{total} .

$$P_{\text{total}} = P_{\text{dynamic}} + P_{\text{static}}$$

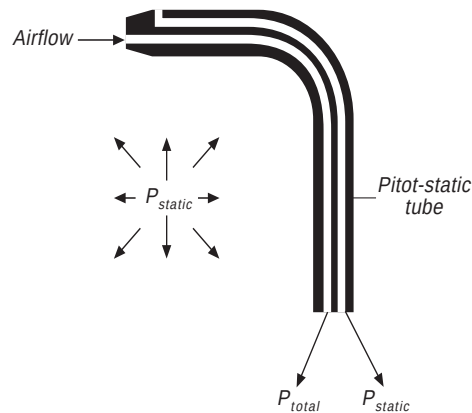


Pitot tube

Although it is possible to estimate P_{dynamic} by this method, it affords poor resolution, as P_{static} is much greater than P_{dynamic} (e.g. dynamic pressure at 150kph will approximately equal 0.02 atmospheres).

What is required is a pitot-static tube, which has a second set of holes around its circumference which effectively sees the static pressure. A differential pressure transducer connected across the tube's two ports is therefore measuring dynamic pressure P_{dynamic} directly as:

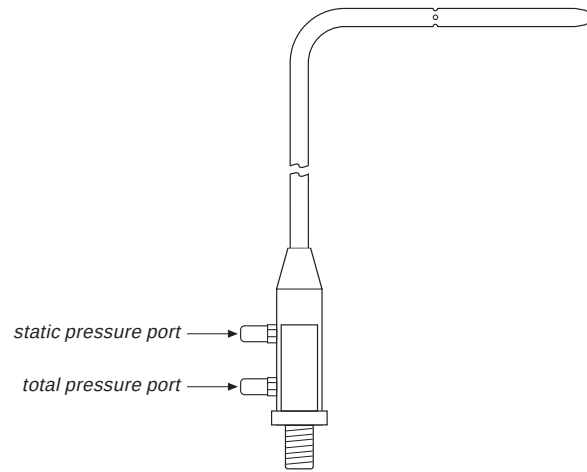
$$P_{\text{dynamic}} = P_{\text{total}} - P_{\text{static}}$$



Pitot-static tube

Pitot-static tubes are carefully designed to give accurate measurements, even when the tube is not precisely aligned with the flow. There are many variants, each with different configurations of nozzle shape and hole positions, and some can tolerate up to 10° flow misalignment.

Note: A common mistake is to assume that "still air" within the cockpit or driver's compartment is at static pressure. This is not the case, as it will vary with the car's speed, and will typically be below static pressure. Never leave the -ve ports of the differential pressure transducers disconnected from a properly derived static pressure, otherwise you will get misleading results. The best source of a static reference is from a pitot-static tube



Pi Pitot-static tube

Contact information

For more information about Pi products and details of worldwide authorized agents, please contact:

Pi Research

Brookfield Motorsports Centre
Twentypence Road
Cottenham
CAMBRIDGE
UK
CB4 8PS

Customer Support Tel +44 (0) 1954 253600
Fax +44 (0) 1954 253601

Pi Research, Inc.

8250 Haverstick
Suite #275
Indianapolis
IN 46240
USA

Tel +1 (317) 259-8900
Fax +1 (317) 259-0137

Part Number: 29B-071141-1E

Issue 1.0 June 1998. Adobe Acrobat© Format

© Pi Research 1998

Pi and the Pi logo are trademarks of Pi Group Limited

www.piresearch.com

