TECH/OPS

SEVCON

Gen4 Application Note

Title	Setting up PMAC software
Filename	App Note - PMAC.docx
Date of Creation	17/09/2009 12:28:00 by Howard Slater
Last Updated	21/03/2011 08:45:00 by Paul Shipley
Revision	14

Contents

Sevcon Ltd
TVTE
Gateshead
Tyne & Wear
England NE11 0QA

Tel +44 (0)191 4979000
Fax +44 (0)191 4824223
www.sevcon.com

1. Introduction

This document is intended to guide an applications engineer through the process of setting up a Gen4 controller using the Permanent Magnet Software.

In order to keep the document simple, the document focuses on setting up the controller to drive the motor not the overall drivability of the vehicle. For this any other application note that deals with drivability is applicable.

Examples of DVT command sequences are provided in several places throughout this document. These commands are intended to provide examples of typical setup values. Although the values provided are typical for most vehicles, the reader should not make the assumption that they are going to work for all vehicle types. It is up to the user to investigate what function is being performed by values being written to the unit's object dictionary. In most cases, looking up the object in the "Master Object Dictionary.xls" spreadsheet, the purpose will become clear.

1.1. Prerequisites

The document assumes some prior knowledge and experience:

- The reader should be familiar with the concept of the object dictionary, and be familiar with reading and writing values to and from the controller. Refer to the CANopen specifications and software design documentation for more information.
- The reader should be familiar with the overall system architecture, particularly the distinction between vehicle application sub-systems and motor control sub-systems. Refer to the software architecture design documentation for more information.
- Knowledge of PDO functionality is also an advantage, particularly when debugging live systems and graphing vehicle behaviour. Refer to the application note "PDO Fundamentals.doc" for more information.

2. What is a PMAC machine and how is it different to an induction machine.

The differences between an Asynchronous Induction Machine and Synchronous Permanent Magnet AC machine are significant despite the similarity of the control strategies and hardware. The principle difference is in the inclusion of magnets on the rotor of the machine to provide the flux in the machine rather inducing the flux via transformer action.

The “synchronous” part of the name that is sometimes associated with PMAC motors comes from the fact that the output frequency of the drive and the mechanical speed of the rotor are always the same. This differs from asynchronous induction motors, where the drive’s output frequency and mechanical motor speed are usually different. This difference, known as slip, will always be zero in PMAC motors.

For induction machines, slip is required to transfer energy to the rotor. The motor does not contain any brushes like a DC motor. Instead, the slip frequency causes energy to be transferred to the rotor through induction, similar to a transformer. A constant slip frequency is required to allow this energy transfer to take place.

In the case on a PMAC motor, the rotor magnets are permanently magnetised, and thus rotor flux is always present. Slip is not required, and the AC output frequency can remain synchronised to the mechanical position of the rotor.

A further consequence of the flux existing regardless of the controller being switched on or not is that the control system no longer has any control over the angle of the flux (an induction machine induces its rotor flux and therefore controls where it sits). Consequently, to induce torque into the machine, the controller needs to know the angle of the flux so that its output vector can be aligned to it. It is for this reason that PMAC motors are normally controlled with either a UVW or sin-cos encoder. These types of encoder are designed to deliver position information rather than speed. However it is important that the encoder signals are aligned with the mechanical position of the rotor. Misalignment in this respect will reduce the torque per amp available.

3. UVW Encoder Installation

3.1. UVW or Absolute Position Encoders and Principle of Operation

UVW type encoders are commonplace on many existing permanent magnet motor systems. Their purpose is to indicate to the controller what the electrical angle of the rotor is so that the correct phases can be energised by the controller in order to make the motor rotate. Several early PMAC controllers used a simple trapezoidal scheme where the required phases were switched on at the same time as the encoder edges were received. Today, more sophisticated control schemes allow the electrical angle to be interpolated between these edges, resulting in a smoother sinusoidal output.

The output signal from a UVW encoder resembles that of a quadrature AB encoder in that it generates a series of digital signals that switch as the motor rotates. The main difference between these encoder types is that the UVW encoder is designed to provide angle information, rather than speed.

The basic setup process for a UVW encoder is as follows:

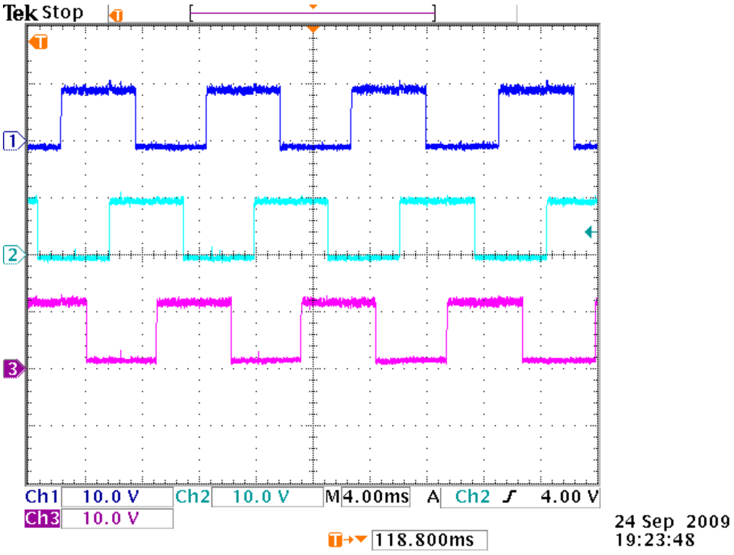
1. Connect the encoder to the controller, verify the ordering of the signals and ensure the controller is able to measure motor speed.
2. Determine which of the motor cables is to be connected to the M1, M2 and M3 terminals on the controller.

3.2. Initial Encoder Verification

Before attempting to drive the motor with the controller, it is advised to check the operation of the encoder using a scope. The UVW encoder generates three signals. The pulses generated on each signal are 120 electrical degrees out of phase. It is important that when the motor is rotating forwards, the pulses are generated on the U input, followed by the V input, followed by the W input, in that order.

The easiest way to test this is to connect the encoder to the controller, but leave the controller in pre-operational. This will allow you to rotate the motor by hand and monitor the signals being received by the controller.

Set up a scope to monitor each of the U, V and W encoder signals, referenced to B-. Rotate the motor in the forwards direction. If the signals are pulsing in the correct order, you should be able to see a trace that resembles the following:



In the above example, channel 1 is connected to U, channel 2 to V, and channel 3 to W. Note that the pulses overlap – this is normal. However, the important thing to note in this trace is that the pulses are occurring in the correct sequence i.e. U first, then V, then W, then back to U again.

If you have a trace where the pulse sequence appears to be reversed, this can be corrected by exchanging any of the two encoder input signals.

Please remember that this test is only valid if rotating the motor in what is considered to be the forwards direction. If the motor is turning backwards, the pulse sequence will be different.

As a final check, you can verify that the encoder is connected correctly by monitoring the controller's speed measurement. Motor speed can be read from object 0x606C in the object dictionary. You should receive a positive value when the motor is rotating forwards, and a negative value when the motor is rotating backwards.

3.3. Determining the Motor Cable Connections

With the encoder connected and working, the next step is to determine which motor cable should be connected to which motor terminal on the controller. PMAC motors require that the phase connections be matched with the encoder connections. This section explains how to match the phases to the encoder signals by measuring the back EMF generated by the motor when it is turned by hand.

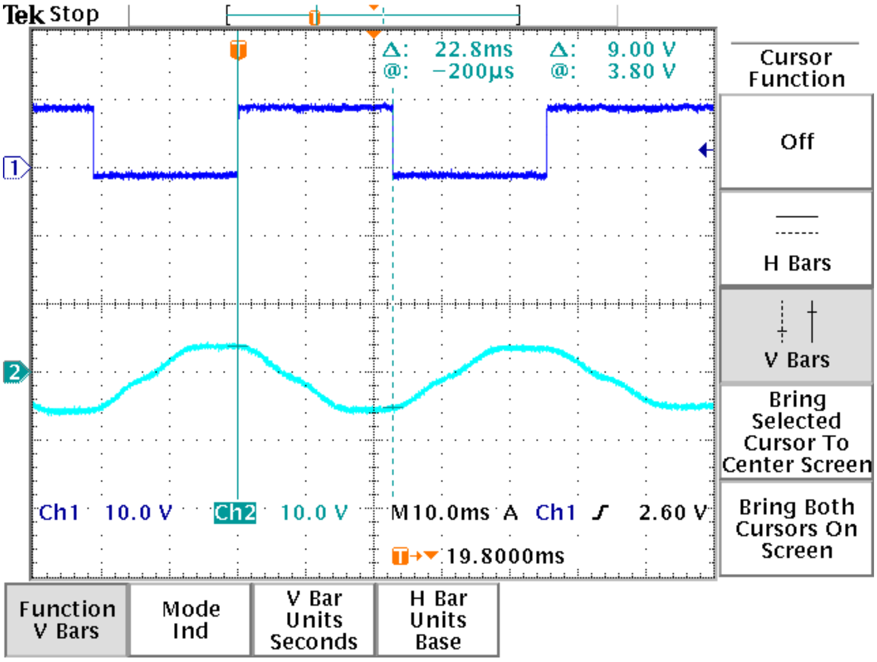
The process requires the following:

- The motor cables should not be connected to anything – especially the controller. Take care that the cables are not shorted to each other, or are touching a conductive part of the vehicle chassis.
- The encoder should be connected to the controller, and the controller is able to receive the signals
- The controller should be powered up in order for the encoder to run, but should be left in pre-operational.

With the above set up, connect one channel on the scope the measure the signal going in to the controller as encoder signal U, with respect to B-. With the other channel, monitor the voltage on one of the phases with respect to any other phase.

Rotate the wheel by hand in the forwards direction. The scope trace you receive will match one of the traces in the following sub-sections. These will describe what action should be taken.

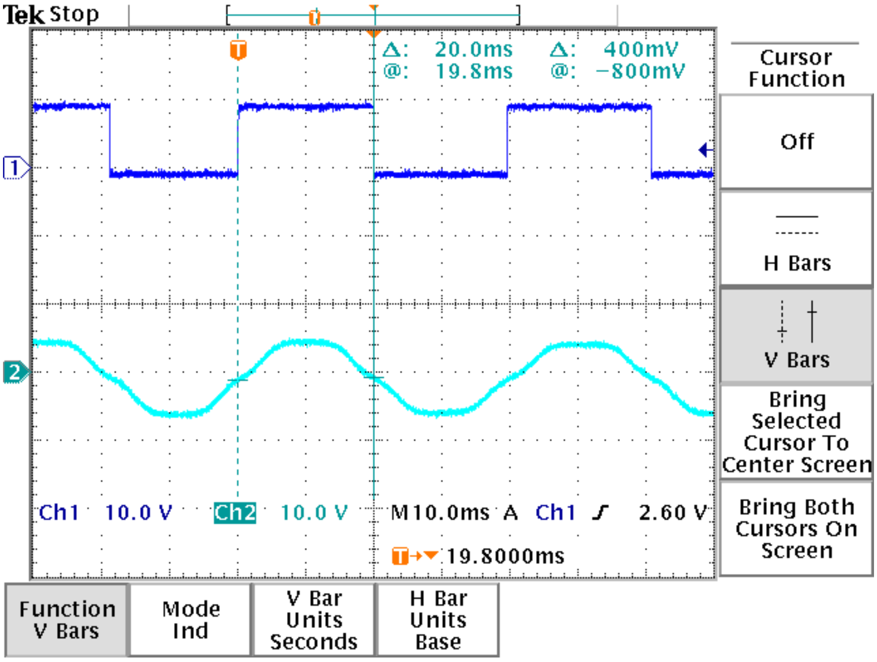
3.3.1. Phase Not Aligned



When aligning the phase to an encoder pulse, the objective is to the back EMF cross the zero point when the encoder edge occurs. In the above trace, we can see that the edges do not align with the back EMF crossover.

To correct this situation, a different phase should be tested. For the scope probe that is in the motor cables, move the probe tip to a different phase. Also, move the probe ground clip to the motor cable that you have just removed the probe tip from. Repeat the test.

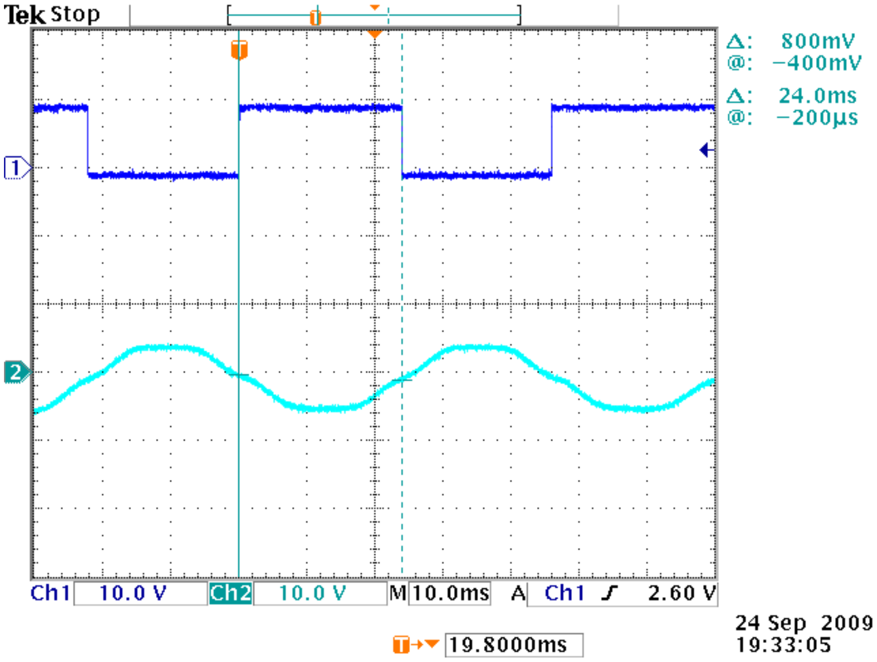
3.3.2. Phase Aligned but Inverted



In this trace, we can see that this time the encoder edges do coincide with the point at which the back EMF crosses zero. However, the phase is reversed. For correct alignment, it is required that the back EMF is low when the encoder signal is high.

To correct this, the connection of the ground clip and probe tip for the probe connected to the motor cable should be reversed.

3.3.3. Phase Matched



The above trace shows the expected result. In this case, the encoder signal 'U' is aligned with the motor phase connected to the scope. Note that the back EMF is low when the encoder signal is high. This is correct when the motor is being rotated forwards.

If you have this graph, then the motor connections are as follows:

- The motor cable connected to the probe tip should be connected to the controller's M1 terminal.
- The motor cable connected to the probe's ground clip should be connected to the controller's M2 terminal.
- The remaining cable should be connected to the controller's M3 terminal.

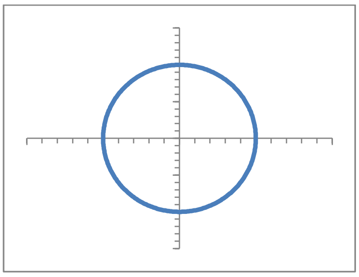
Power down the controller and make the relevant motor terminal connections.

4. Sin Cos Encoder Installation

4.1. Introduction

An increasingly popular method to achieve absolute position information to a vastly higher precision than the equivalent UVW (see below) system is known as a sin cos encoder. This is a hall effect device responding to a shaft mounted magnet. The device produces two outputs a sinusoid and co-sinusoid which when represented as an x-y polar co-ordinates give instantaneous and precise information of the rotor position. The encoder output will cycle once for each mechanical revolution of the rotor.

A typical sine-cosine encoder will have a voltage range of +/-1.5 V which gives an angle resolution of (0.17 x Npp) degrees per bit. This improves low speed torque ripple and simplifies the angle algorithms when compared to UVW type encoders.



Scope plot of sine input versus cosine input for a rotating motor

4.2. Controller Hardware Requirements

Sin-cos encoders typically generate a signal with a magnitude of a few volts. However, for accurate position measurement, the scaling of the inputs must be changed.

The original circuit design for Gen4 allowed digital and analogue input voltages to be measured up to the controller's peak voltage rating. While this allows for flexible vehicle signal wiring, it does not provide enough resolution for the sin-cos encoder input. Instead, the resistor values for voltage measurement were changed in the controller, allowing a smaller voltage range to be measured but with more precision.

These modifications were made to analogue input 4 and digital input 8, which have now become the sin and cosine inputs.

Pin	Function
15	Encoder 0V
26	Encoder supply (5V or 10V, depending on configuration)
21	Sine input (was digital input 8)
35	Cosine input (was analogue input 4)

Please be aware that due to the change in the input circuit, the controller's protection is now limited. While excessive voltages on the input pins should not cause permanent damage, they could affect the measured values of other inputs. Do not apply more than 5V to these inputs.

Not all Gen4 controllers have had the hardware modification for sin-cos inputs. To determine if the controller is suitable for use with sin-cos encoders, the hardware version should be checked. Sin-cos support is encoded into bit 24 of the hardware version, which can be read from object 0x1009,0 of the controller's object dictionary using the following DVT command:

```
sdo_rns 1 0x1009 0
```

Example hardware versions:

0x01050004	- 80V 550A hardware version 4 with sin cos hardware modification.
0x00050004	- 80V 550A hardware version 4 with standard hardware.

4.3. Initial Encoder Verification

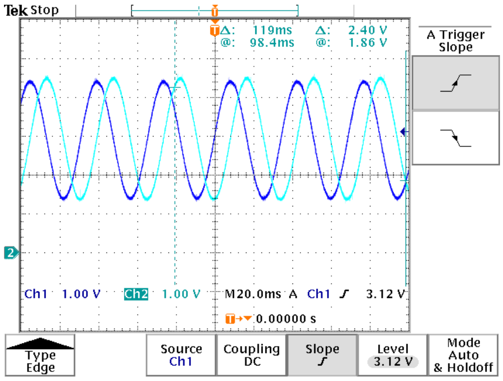
Before using the sin-cos encoder, it is advised to check the function of the encoder using a scope. This allows you to check the encoder is working correctly, and determine which signal is sin and which is cos.

First, the following object dictionary entries should be set:

- Object 0x4630,3 should be set to a value of 5 specifying the sin-cos encoder type.
- Object 0x4630,1 configured the internal pull up. This is not likely to be required. Setting this object to a value of 1 will disable the pull up.
- Object 0x4630,2 sets the supply voltage for the encoder. This can be 5V or 10V. Check the encoder's datasheet for the appropriate voltage.

Next, connect the encoder supply cable (normally coloured red) to pin 26 of the Gen4, and the encoder ground (normally coloured black) to pin 15 of the Gen4. Power up the controller.

Using the scope, measure the voltage between the encoder ground and each of the signals. Rotate the motor in the forwards direction. You should be able to see the sine and cosine signals from the encoder.



Tek Stop

Δ: 119ms Δ: 2.40 V
@: 98.4ms @: 1.86 V

Ch1 1.00 V Ch2 1.00 V M20.0ms A Ch1 ƒ 3.12 V
0.00000 s

Type Edge
Source Ch1
Coupling DC
Slope ƒ
Level 3.12 V
Mode Auto & Holdoff

If the motor is rotating forwards, then you will see the cosine signal leading the sine signal. So, for the above example, the channel 1 (dark blue) represents cosine, and the signal cable corresponding to this should be connected to pin 35 on the Gen4. Similarly, the channel 2 (light blue) represents sine, and should be connected to pin 21 on the Gen4. In other words, the peak of the cosine signal will occur just before the peak of the sine signal.

Please note this is only applicable when the motor is rotating forwards. If the motor is rotating backwards, then the signals will be reversed. When setting up the motor wiring there must be a clear convention on which motor direction constitutes as rotating in the forwards direction. While this is straightforward for most vehicles, other vehicle types e.g. sideloaders or palette trucks the distinction may not be as clear. Please be sure the forward direction is defined and the motor is rotating in this direction during set up.

4.4. Commissioning the Sin Cos Encoder

Use of a sin-cos encoder requires that the minimum and maximum voltages for each signal are programmed into the controllers object dictionary. In the above example, the magnitiude of the signals is approximately 1.5V to 4.5V. However, the values loaded into the object dictionary will need to have extra precision. Furthermore, on closer inspection of the above trace, we see the minimum and maximum voltages for each signal are slightly different. In this case, the sine signal is a few millivolts greater than the cosine signal. While this difference may seem small, it is enough to cause problems in the angle calculation algorithm.

Fortunately, the controller provides a measurement of the minimum and maximum voltages it measures in the object dictionary. These are located in the encoder configuration object.

Object	Function
0x4630, 5	Sine input minimum voltage configuration
0x4630, 6	Sine input maximum voltage configuration
0x4630, 7	Cosine input minimum voltage configuration
0x4630, 8	Cosine input maximum voltage configuration
0x4630, 9	Minimum measured voltage on sine signal
0x4630, 10	Maximum measured voltage on sine input
0x4630, 11	Minimum measured voltage on cosine input
0x4630, 12	Maximum measured voltage on cosine input

In object 0x4630, used to configure the encoder, minimum and maximum voltages for the inputs should be written to sun-indices 5, 6, 7, and 8. The minimum and maximum voltages measured by the controller can be read from sub-indices 9, 10, 11 and 12.

The values in these objects are measured in volts, scaled such that 0x0FFF equates to 6.0V on the input. So, to convert these values to voltages, multiply by 6 and then divide by 4095.

The following procedure can be used to commission an encoder:

3. Connect the sine and cosine signals from the encoder to their relevant inputs.
4. Put the controller in to pre-operational and cycle power.
5. Rotate the motor shaft manually by hand, such that the controller can see the signal for several complete revolutions. While this happens, the controller will monitor the minimum and maximum voltages measured on the inputs. These values can be read from sub-indices 9, 10, 11 and 12.
6. To commission the encoder, simply copy the values in sub-indices 9, 10, 11 and 12 to their respective entries at sub-indices 5, 6, 7 and 8.

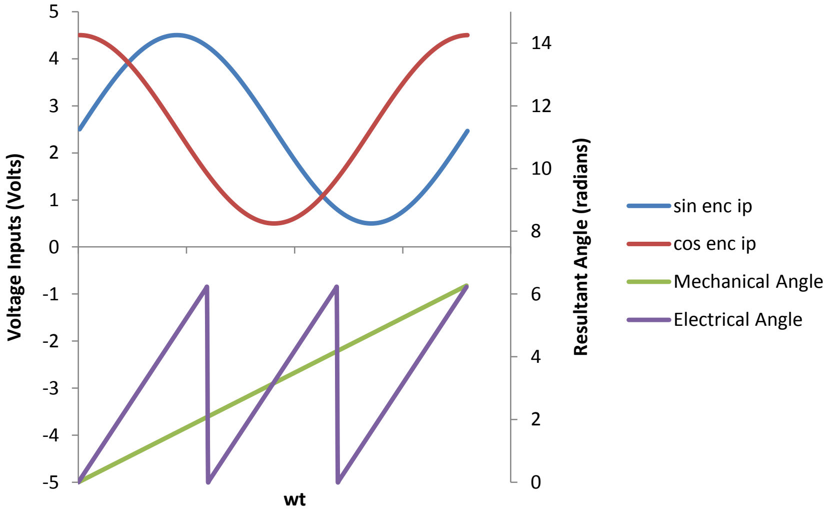
The following piece of DVT code can be used to perform the final step in the procedure, copying the minimum and maximum voltages in as appropriate. It will also output the actual voltages it measured, so these can be confirmed against what was measured by the scope.

```
proc commit {{node_id 1}} {
    puts "[sdo_wnx $node_id 0x4630 5 [set r [sdo_rnx $node_id 0x4630 9]]] sin min set
       to [expr $r] adc counts ([expr (6*$r/4095.0)]V)"
    puts "[sdo_wnx $node_id 0x4630 6 [set r [sdo_rnx $node_id 0x4630 10]]] sin max set
       to [expr $r] adc counts ([expr (6*$r/4095.0)]V)"
    puts "[sdo_wnx $node_id 0x4630 7 [set r [sdo_rnx $node_id 0x4630 11]]] cos min set
       to [expr $r] adc counts ([expr (6*$r/4095.0)]V)"
    puts "[sdo_wnx $node_id 0x4630 8 [set r [sdo_rnx $node_id 0x4630 12]]] cos max set
       to [expr $r] adc counts ([expr (6*$r/4095.0)]V)"
}
```

When setting the encoder voltages, it is also advised to check the magnitude of the signal. Normally there should be at least 2V between the minimum and maximum voltage signals. Any less than this could result in problems occurring due to poor resolution or interference from external noise sources.

Given a valid encoder configuration, you should be able to measure a motor speed, in rpm, from object 0x606C. The speed measurement should be stable, give a positive number when the motor rotates forward and give a negative number when the motor rotates backwards.

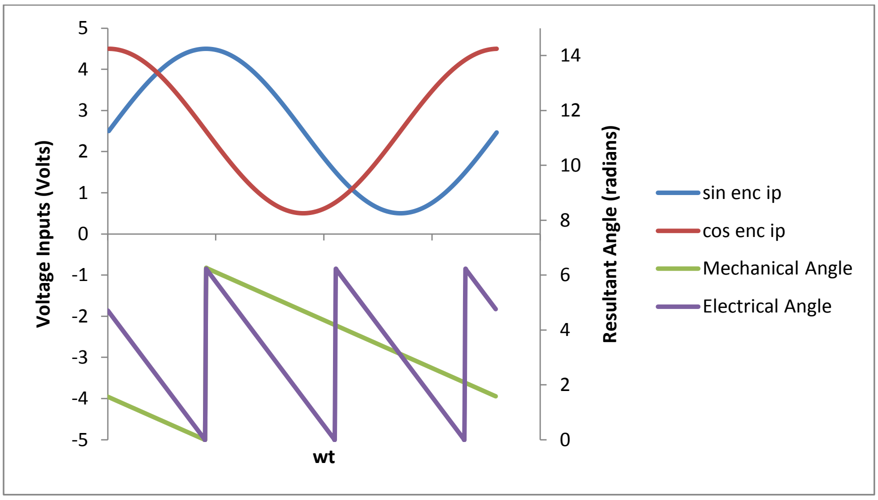
4.5. Sin Cos Interpretation by Gen4



Correct Sin Cos Conversion to Angle

The calculated angle indicates to the controller where in space the rotor flux is pointing as this is the direction which the torque producing current must be applied. Therefore the mechanical mounting of the encoder must be such that the calculated angle of the encoder corresponds to the angle of the rotor flux.

The two waveforms are taken and normalised to PU values this results in both the sin cos and cos inputs taking a -1 to 1 value, this is then converted to an angle via a high precision arctan. In the graph above is can be seen that clockwise rotation from the shaft end results in a positive increment in the angle. The derivative of this (ie the gradient = dangle/dtime) is the speed and it can be seen as a positive speed(gradient of angle) above. This however results in a mechanical angle which requires further modification to convert it to an electrical angle as shown in the graph above. This then becomes the angle used by the current control algorithms. As this is a direct path to the current controllers care should be taken to shield this signal from noise and should be routed away from the power cables.



Incorrect Sin Cos Conversion to Angle

By inverting the sin cos inputs to the algorithm a 90 degree phase shift in the mechanical angle and reverse in direction is seen. This is wrong. This is same effect as wiring the sin and cos inputs to the controller the wrong way. If there is an even number of poles there will be one other motor phase combination which corrects this mistake but operation cannot be guaranteed.

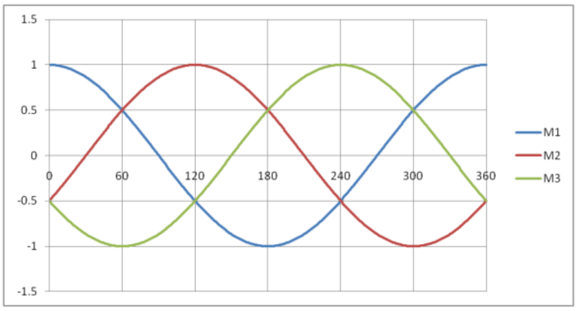
4.6. Determining the Motor Cable Arrangement

Once the sine and cosine inputs have been connected and the controller is able to measure motor speed correctly, the next step in the set up sequence is to connect the motor cables. Normally, a PMAC motor is supplied with cabled marked “U,V,W”, or “1,2,3”, but these do not necessarily correspond directly to the “M1,M2,M3” connections on the controller. The objective of this next section is to determine which motor cable should be connected to which terminal on the controller.

Before proceeding, the following needs to be completed:

7. A convention has been decided for which motor direction represents “forward”.
8. The sine and cosine encoder inputs have been set up and are connected to the controller.

It is also worth knowing in advance how the motor voltages are expected to interact with the motor.



The voltages on each phase rise and fall, 120 degrees out of phase. What is important to note here is that phase M1 rises first, aligned to zero degrees. M2 rises next in sequence, 120 degrees later. A further 120 degrees again, and M3 rises. The rotational sequence is M1, M2, M3, M1, M2, M3, etc… . The objective here is to align the back EMF of the motor to this sequence of rising edges.

The remainder of this section of the document will assume the motor cables are labelled “U”, “V” and “W”. While this may not be the case with the motor you are dealing with now, the set up procedure is still valid. The actual names of the cables are not important – providing the naming scheme remains consistent throughout the set up process.

4.6.1. Measurement of back EMF angle offset

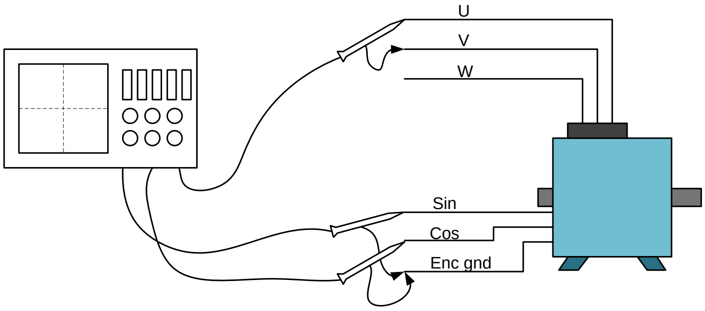
The first part of the process is to use a scope to measure the back EMF from each motor phase, and determine its offset from the zero angle. The back EMF of the phase can be measured by connecting the positive tip of the scope probe to the channel to measure, and connecting the ground clip to the next phase in turn. Thus:

- To measure the back EMF on phase U, connect the probe tip to the U phase motor cable and connect the probes ground clip to the V phase motor cable.
- To measure the back EMF on phase V, connect the probe tip to the V phase motor cable and connect the probes ground clip to the W phase motor cable.
- To measure the back EMF on phase W, connect the probe tip to the W phase motor cable and connect the probes ground clip to the U phase motor cable.

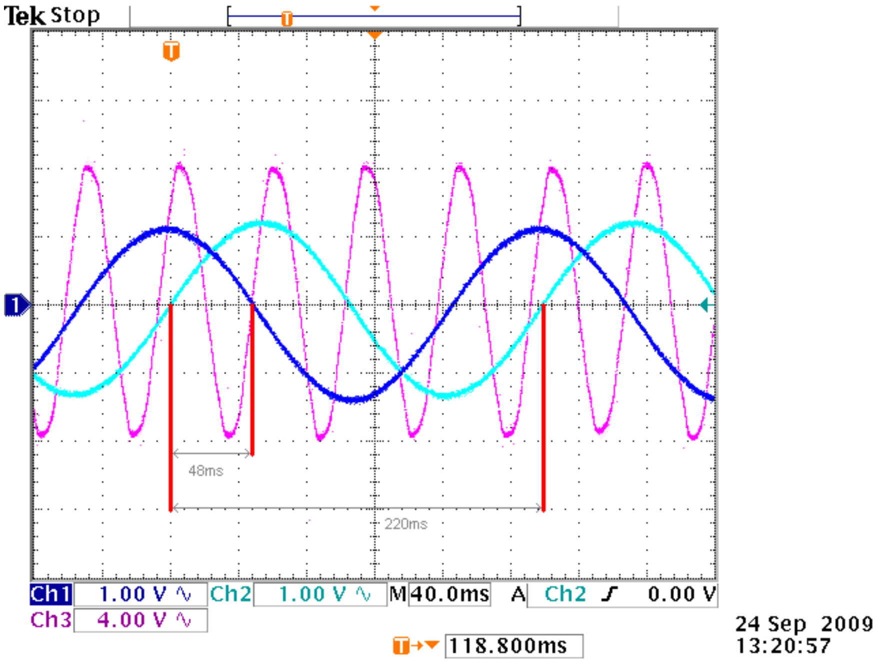
It is important that this test is completed with the motor cables disconnected from the controller. Any rotation of the motor will cause a voltage to be generated. If the motor is connected to the controller at this point, the controller’s internal circuitry will affect the results.

It is also required to configure the scope inputs as AC coupled. This will allow the centre points the sin-cos signal and back EMF to be identified easily.

Start by measuring the U phase. Connect the positive tip to U, and the ground clip to V. Remember, only the encoder is connected to the controller - the motor phase cables should be left disconnected at this stage.



When the motor is rotated by hand, you should get a scope trace showing the back EMF along with the sine and cosine encoder signal:



The blue lines show the encoder sine and cosine signals. The pink line is the measured motor back EMF. The number of electrical back EMF cycles per mechanical sin-cos cycles will equal the number of pole pairs in the motor. In this example, the motor has 4 pole pairs. Therefore, there are four electrical rotations per mechanical rotation of the motor.

From this trace we need to measure two things:

- The length of time for one complete mechanical revolution
- The length of time between the encoder showing zero degrees and the start of the phase's electrical cycle.

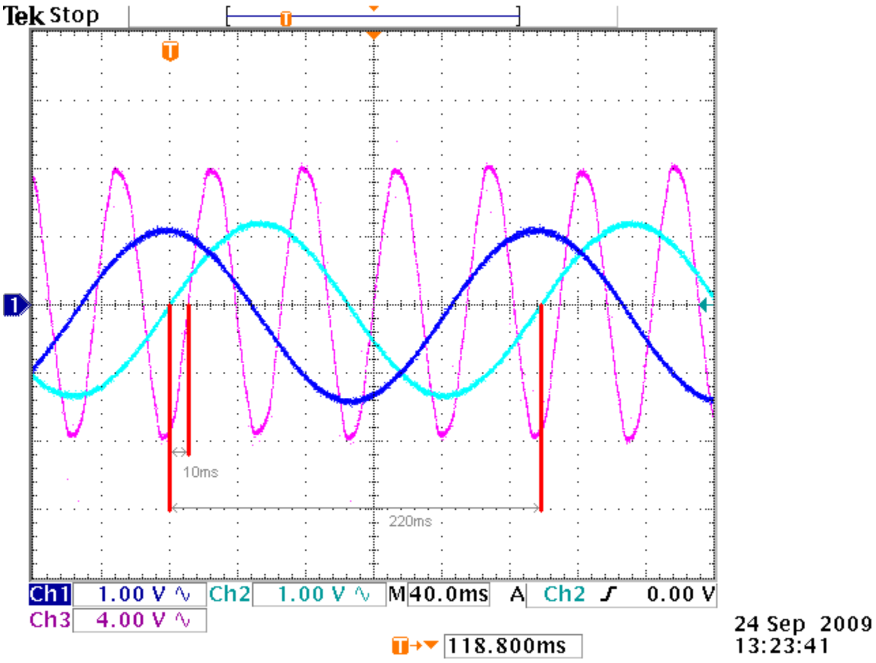
The length of time for a complete mechanical revolution can be calculated by measuring how long it takes between rising points on the sine signal. in the above trace, a trigger has been set to activate on the sine signal crossing from a negative value to a positive value. The time between this event and the next is 220ms. This has been marked on the trace.

The angle offset for the phase under test can be found by measuring the time between the same point on the sine wave, and the next point after it at which the back EMF goes positive. In the above example, this has also been marked on the trace, and is measured as 48ms.

The angle offset for this phase can be calculated using the following formula:

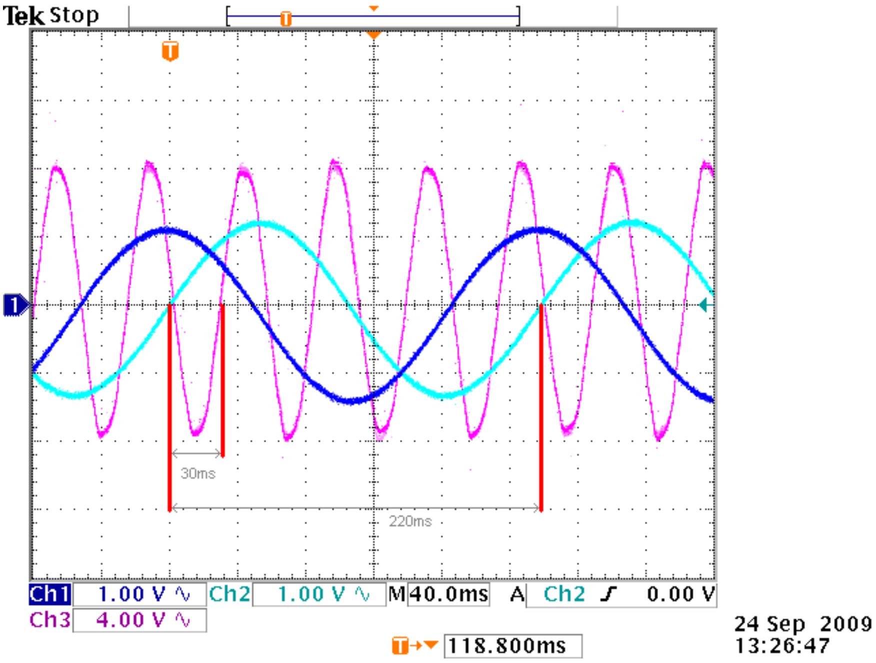
So, when the time to the next phase is 48ms, a full rotation is 220ms and we have 4 pole pairs, the angle offset is 314°.

Next, the process is repeated with the probe tip connected to V and the ground clip connected to W. The following graph is obtained:



In this trace, the time is shorter, approx 10ms. For this test, the time for a single mechanical rotation is still 220ms, as the motor was rotating at the same speed as the previous test. Using the above equation, we get an angle offset of 65°.

The final test is performed with the probe tip connected to W and the ground clip connected to U. The following graph is obtained:



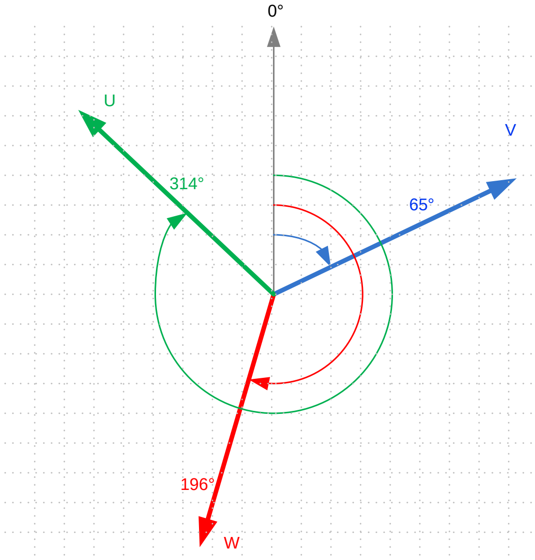
In this case, the time is 30ms. The angle offset is calculated as 196 degrees.

4.6.2. Determining Phase Order

From these tests, a table can be produced showing phase and angle offset:

Phase to which probe tip was connected:	Angle offset
U	314°
V	65°
W	196°

If these angles are plotted as vectors then we would have:



As a final check, you should see that the vectors are approximately 120° apart. Although the measurement scheme is not exact, there should be at least some correlation. If the vectors are not evenly spaced, there is likely to be a measurement error and the process should be repeated.

The final step is to actually attach the cables to the motor terminals on the controller. The phase that is closest to zero degrees is connected to M1, in this case it is U. The next phase in clockwise rotation is connected to M2, in this case V. The next phase in clockwise rotation again is connected to M3, in this case W.

4.7. Setting the Encoder Offset

Providing you have the sin and cosine inputs set correctly, and you have M1, M2 and M3 set up as per the previous section, then the system should be aligned well enough to see some smooth motor rotation when drive is requested. This can be checked by requesting drive torque in the usual ways; either via TracApp or one of the other vehicle application sub-systems, or by using the DVT to control motor demands directly. Drive in both directions should be possible, and measured speed at object 0x606C should be accurate.

Whilst the motor might be turning at this point, best motor performance is achieved when the encoder alignment parameter is set. This parameter will correct for any mechanical deviation from the encoder’s zero point, giving the controller a better measurement of the angle. Better angle measurement means the controller’s output vector is better aligned, improving efficiency.

Techniques for calculating precisely what the angle offset is do exist, but are difficult to perform when specialist equipment such as power analysers and dynamometers are not available. If the motor is spinning, then there are experimental techniques that can be used instead to maximise efficiency.

This section outlines some simple methods that can be used to maximise output efficiency. Either can be used, but the basic principle remains the same – trying to maximise the motor’s torque output per ampere input. Depending on the application, it may be possible to modify or adapt these schemes to improve accuracy.

4.7.1. Writing an Offset to the Object Dictionary

The encoder offset is stored in the object dictionary at object 0x4630 sub-index 4. The offset can be negative as well as positive, and is usually in the range +/-60°. The value is stored in the object dictionary as a signed 8.8 format integer. Values can be updated while the motor is rotating. This allows small adjustments to be made on-line, to see how they affect motor control.

The following DVT command can be used to set the angle offset:

```
sdo_wnx 1 0x4630 4 [ af <angle> S8.8 ]
```

The “S8.8” at the end of the command indicates this is a signed number in 8.8 format. Replace <angle> with the desired offset angle. The following examples are all valid:

```
sdo_wnx 1 0x4630 4 [ af 7 S8.8 ]
sdo_wnx 1 0x4630 4 [ af 12.3 S8.8 ]
sdo_wnx 1 0x4630 4 [ af -5.8 S8.8 ]
```

Before starting the alignment procedure for the first time, the offset value should be reset back to zero:

```
sdo_wnx 1 0x4630 4 [ af 0 S8.8 ]
```

4.7.2. Using the d-axis Voltage

This method uses the d-axis voltage as a check to see how much q-axis current is actually going into the d-axis. When the encoder is aligned, this effect should be minimised. Whilst this method is not completely accurate (this is due to a cross coupling effect) it can give a broad indication as to what the actual offset value is.

This method also has the advantage in that it is relatively quick to perform, and can be executed on a motor with little load applied, such as a vehicle wheel raised off the ground.

To start, with the vehicle in pre-operational, map the Ud, Id and Id_ref objects to TPDOs, allowing them to be monitored with the vehicle interface viewer:

```
map_in_tpdo 1 0x4600 5
map_in_tpdo 1 0x4600 7
map_in_tpdo 1 0x4600 9
```

Now with the controller in operational, request a small amount of drive demand against an unladen motor. Try to find a level of drive torque that gives the following:

- Torque demand should be absolutely constant. You can use a pot to set the torque demand, or set demands directly from the DVT. If you only have an accelerator pedal, don’t try to hold a steady demand using your foot – it won’t be accurate enough. Instead, set 100% accelerator push to be a low torque demand (e.g. 5%) and hold it there.
- The motor should be spinning steadily, ideally somewhere between 1000 and 2000 rpm.
- The motor should not be in field weakening, i.e. Id_ref should remain constantly at zero.

This may take only 5% of drive torque but providing the motor is spinning steadily, the test can be performed.

The idea here is to run the motor with the target d-axis current set to zero. Controlling zero amps on the d-axis should take zero volts, but if there is an error in the angle then we start to see current from the q-axis couple on to the d-axis. The end result is that while zero amps is being controlled, it takes a small amount of d-axis voltage to maintain it.

With the motor running, look at Ud. If it appears to be operating at around zero volts, then this is a sign that the encoder is closely aligned. If Ud is too low or too high then the encoder alignment may need to be adjusted.

Comment [hjs1]: What about corss coupling of the voltage, there will be some Ud voltage due to iq depending upon electrical frequency and motor inductances.

It is best to experiment with this to see how adjusting the offset angle affects Ud. While the motor is running, increase the angle by one degree at a time and monitor what happens to Ud. Typically, if Ud is lower than zero then the angle offset should be increased, and if Ud is higher than zero then the angle offset should be decreased.

4.7.3. Maximising the Output Efficiency

Making small changes to the offset angle of only one degree at a time is a good way to see how the alignment of the encoder affects the efficiency of the motor. As discussed earlier, the above method does not provide an exact encoder offset value. This is due to a cross coupling effect where it is normal for some q-axis current to be seen on the d-axis and vice versa, even for a perfectly aligned motor. However, by keeping a constant torque demand and monitoring the motor speed instead, you can get a better idea of what adjustments should be made to the offset.

Remember, the main objective here is to get maximum motor output for a given input. Keep the torque demand constant and perform more adjustments to the offset angle. You should find that for different offsets, the motor speed will stabilise at different values. It may take a while for the speed to level out, but with some experimentation you should be able to find a more accurate offset value.

5. Object Dictionary Encoder Setting

The encoder type needs to be set in the object dictionary:

Object 0x4630 sub index 3:

Sub 2	Encoder Type	Uses	Comment
0x0000	AB only	Induction Motor	Not for use with PMAC motors
0x0001	AB and UVW	PMAC with both encoders	AB encoder not supported – use UVW only instead.
0x0002	UVW only	PMAC with single UVW encoder	
0x0003	AB and inverted UVW	PMAC with both encoders	AB encoder not supported – use inverted UVW instead.
0x0004	Inverted UVW only	PMAC with single UVW encoder	
0x0005	Sin-cos encoder	PMAC with sin-cos encoder	
0x0006	Single channel A	Induction motor intended for rotation in one direction only (e.g. pump)	Not for use with PMAC motors

6. Motor Parameters

In comparison to the flux vector induction machine algorithm the permanent magnet software is far less dependent upon motor parameters.

Like the Flux Vector code the PMAC code uses two current references, ‘D’ and ‘Q’. The D axis current controls the flux and the Q axis current controls the torque. Because the motor has its field provided by the magnets the D axis current is usually controlled to 0A.

In contrast to Induction Machine control the parameters required for PMAC control are considerably simpler.

1) Current control gains
2) Field weakening control gains
3) Maximum modulation index
4) Minimum field weakening current
5) Maximum stator current.
6) Back EMF constant Ke
7) Power limit table.

6.1. Current Control Gains

These are the fundamental inner current control loop gains. It is of the upmost paramount importance that some time is spent ensuring that the current control is adequate.

6.2. Field Weakening Control Gains and Maximum Modulation Index

To enable the current controllers to be able to control current there must be available voltage, as the motor spins faster the rotor magnets generate more voltage. When this voltage exceeds the maximum modulation index the control algorithm uses a PI controller to govern the set point of the flux producing current Id. Unlike in the induction machine this current is negative and is used to effectively oppose and reduce the magnet flux. This reduces the amount of voltage generated and therefore allows the current controllers to continue to operate.

6.3. Is Max

This is defined as a maximum short term current rating of the motor. Care must be taken with thermal protection and duty so that motor is not overheated in long term operation.

6.4. Back EMF constant Ke

The most important parameter to set up is the back EMF voltage constant Ke. This constant dictates how much current is demanded for a given torque demand. There is a direct relationship between torque and the ‘Q’ axis current. It is also governs the size of the feed forward terms for the q-axis controller, this is important for starting into a spinning motor.

So for example for a 4 pole pair motor with a Ke=0.022

Tdemand = 10Nm , Current demand = 10/(4x1.732x0.022) = 65A

Tdemand = 20Nm, Current demand = 131A

Etc.

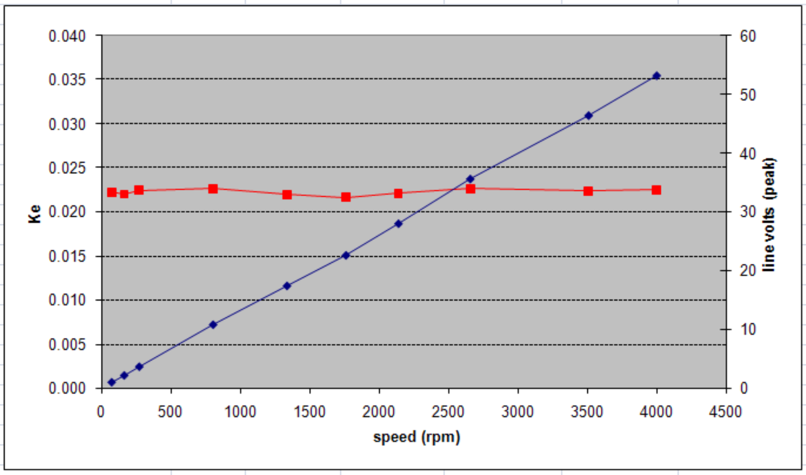
This direct relationship between current and torque can be used when setting maximum torque settings.

The back emf constant is defined as:

$$\boxed{K_e = \frac{V_{line}}{\omega_e}}$$

Where Vline is the rms line to line voltage generated and we is the electrical frequency. Clearly this can be easily measured with the motor not connected to the controller. If the motor is rotated externally, and the line voltage is captured on an oscilloscope.

Speed (rpm)	pole pairs	Freq (Hz)	we (rads/sec)	Volts Line Peak	Volts Line RMS	Ke
76	4	5.1	31.83	1	0.707	0.022
165.9		11.1	69.49	2.16	1.527	0.022
274		18.3	114.77	3.64	2.574	0.022
806		53.7	337.62	10.8	7.637	0.023
1338		89.2	560.46	17.4	12.304	0.022
1763		117.5	738.48	22.6	15.981	0.022
2141		142.7	896.82	28	19.799	0.022
2658		177.2	1113.38	35.6	25.173	0.023
3510		234.0	1470.27	46.4	32.810	0.022
4000		266.7	1675.52	53.2	37.618	0.022



Ke is entered via 0x4641 sub index 18 entered in 1.15 format.

6.5. Minimum Magnetising current

Under normal operation the fluxing current Id is controlled to be 0A. When motor speed increases with applied torque the back emf also increases up to the point where it equals the DC link, current can no longer be driven into the motor. This is referred to as the base speed of the motor and obviously varies with battery voltage. In order to allow operation at higher speeds the motor, like an induction machine, must be field weakened. This is done by controlling negative current to oppose the field of the magnets.

There is a PI control loop which attempts to limit the applied voltage to the maximum motoring mod index (0x4641 sub 30) usually set to 90%. This PI loop reduces Id demand below zero down to a minimum of the value set as the minimum magnetising current 0x4641 sub 3.

This number must be entered as a negative number, the value is range checked to ensure that only numbers 0 and less are set in the object.

The number is an 1 amp per bit format, so for example if -50A was required as the minimum value then a value of 0xffce has to be entered into 0x4641 sub 3. The following table shows a selection of minimum magnetising currents and the correct 16.0 format number:

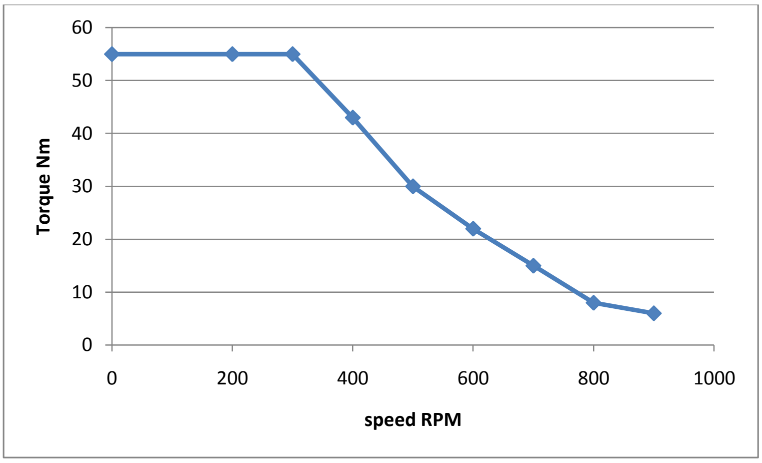
Minimum Magnetising current	0x4641 sub 3, 16.0 format
-5A	0xFFFB
-25A	0xFFE7
-50A	0xFFCE
-75A	0xFFBD
-100A	0xFF9C

The gains of the PI controller which reduces Id reference are the usual field weakening controller gains freq_Kp and freq_Ki which are set via 0x4641 sub 25 and sub 26 respectively.

In to prevent unexpected behaviour of the motor the current controllers must always be in control this is managed by keeping the mod index away from 1. In order to do this field weakening must be allowed to control the Id reference to the value which allows the mod index to be held at 90% up to the value of Is max.

6.6. Power Limit table

The power limit table 0x4611 is used in exactly the same way as in the flux vector software. The peak torque in this table should match the motor peak torque value set in 0x6076.



It was found on early applications that the power limit table if not known could be derived on the application. This was done be setting the power limit values flat, i.e. all torques set to the maximum value. The vehicle was then driven and a DVT trace taken of the Iq current. The actual current would reduce naturally as the speed increases even though the reference was constant, the current controllers would be saturated. The Ke constant could then be used in the relationship between torque and Iq in order to calculate what torque would result in the actual Iq currents measured. By doing this the reference current would then be very close to the actual achievable.

In order to perform this procedure the current control fault will need to be disabled by setting Temporary DSP Configuration (0x4653,0) bit 0. This setting is not persistent and will be lost after a key recycle.

7. On Road eBike Applications

For on road e-bike applications of PMAC the following features must be configured in the specified way in order to prevent unforeseen and potentially dangerous situations occurring on the vehicle.

- *Roll off (0x2930,1) Disabled*
- *Hill hold delay (0x2901,0) set to 0s*
- *Speed limit proportional to throttle disabled (0x2910,1 bit 2) set.*
- *Belly switch not configured (0x2139 does not appear in 0x3300)*
- *Inch switches not configured (0x2129, 0x212A do not appear in 0x3300)*
- *Electro brake not configured (0x2420 does not appear in 0x3200)*
- *Torque mode (0x2900) set to 0, AND (0x6060) set to 4*

The following parameters are highly recommended suggestions which will make the vehicle initially perform in an acceptable way.

- *Neutral braking set to 0%*
- *Direction braking set to 100%*
- *Direction braking proportional to throttle (0x2910,1 bit 0) set*
- *Torque ramp up and down rates set appropriately fast*

8. Summary

Steps to take in setting up a PMAC Gen4

With the motor not connected to the Gen4, but the UVW wires connected to Gen4, rotate the motor by hand and check the back emf voltage against the UVW encoder signals. Identify UVW lines and whether or not encoder is inverted. See section 3.
Still with the motor not connected to the Gen4 controller, spin the motor as fast as possible, measure back emf generated. Determine motor constant Ke as per section 4.1 enter this into 0x4641 sub 18.
Set maximum torque values, in 0x6076 and power limit table 0x4611. You can check maximum current which will be requested by using the Ke value and the equation in section 4.1
Set minimum magnetising current 0x4641 sub 3, ensure some gains are set in 0x4641 25 and 26

9. Extras

A NOTE indicates important information that helps you make better use of your computer system.

A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

A WARNING indicates the potential for bodily harm and tells you how to avoid the problem.