Application Note

CML Microcircuits

COMMUNICATION SEMICONDUCTORS

CMX998 Cartesian Feedback Loop DC Calibration

Publication: AN/2WR/998/DCCal/3 December 14

1 Introduction

This Application Note is intended as an aid to customers developing linear transmitters using the CMX998 Cartesian Feedback Loop IC[1]. This IC allows the implementation of an analogue Cartesian feedback loop (CFBL) around a power amplifier (PA). Transmitters using this topology require calibration and control to ensure correct performance. One such control element is the correction of DC offsets in the loop, a well-known issue with Cartesian feedback because the loop is DC coupled. The DC offset in the signal results in unwanted carrier being present in the output signal, so called 'carrier leakage'. This Application Note focuses on the correction of these DC offsets.

This document describes a test environment using the CMX998 and the CMX981 Advanced Digital Radio Baseband Processor IC[2]. Results of controlling the devices using a Texas Instruments C5510 DSK[3] are included along with suggestions on how a DC calibration loop can be implemented using the Auxiliary ADC/DACs and Sigma-Delta DACs (Tx DACs) on the CMX981.

In this Application Note, TETRA standard[4] Pi/4-DQPSK modulation is used as an example to demonstrate the performance of the techniques discussed. TETRA is chosen because it is anticipated that readers will be familiar with its real system performance relative to its specification. It should be noted however that the techniques and results noted herein are equally applicable to many other modulation schemes and international standards.

The latest versions of the CMX998 Data Sheet and the EV9980[5] and EV9810[6] Evaluation Kit User Manuals should be downloaded from the CML website, www.cmlmicro.com.

2 Contents

3 Hardware Configuration

The test setup includes the EV9980 Evaluation Kit, the EV9810 Evaluation Kit and a Texas Instruments TMS320VC5510 DSK. Figure 1 shows the connections made between the boards.

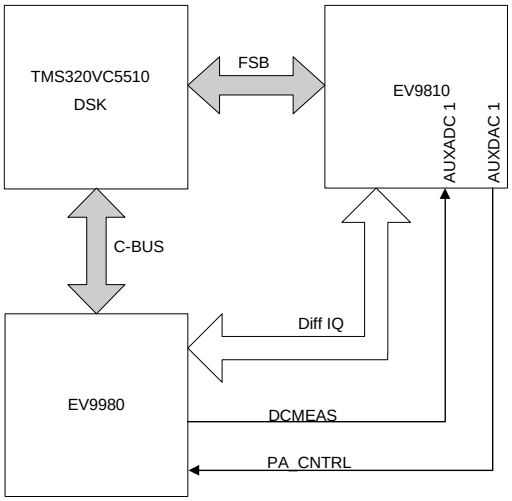


Figure 1. Test Setup for DC Calibration Demonstration

3.1 Interconnections

The DSK acts as a controller for the EV9980 and EV9810 cards and provides three Multichannel Buffered Serial Ports (McBSP) for serial digital control. One of these is configured to a C-BUS compatible mode and interfaced to the C-BUS port of the EV9980. The other two are configured as fast serial ports to provide a fast serial bus, FSB, to the EV9810.

The EV9810 and the EV9980 should be connected as shown in the EV9980 User Manual. The DCMEAS output[1] and the PA_CNTRL output should be connected as shown in Table 1.

EV9810 Signal name	EV9810 Connector:pin no.	EV9980 Connector:pin no.	EV9980 Signal name
AUXADC1	J5:3	J12:3	DCMEAS
AUXDAC1	J5:15	J12:15	PA_CNTRL
AGND	J5: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20	J12: 2, 4, 6, 8,10, 12, 14, 16, 18, 20	AGND

Table 1. EV9980 and EV9810 interconnection details

3.2 CMX998 Input Amplifiers

The CMX998 Input Amplifiers should be re-configured as shown in Figure 2 to allow the EV9980 to interface directly to the EV9810. 100k resistors should be fitted at unused locations R44, R14, R51 and R9. R100 and R97 must be removed, R53 and R7 must be replaced with 620k resistors and R45 and R13 must be replaced with 10k resistors.

[1] DCMEAS is a pin on the CMX998 that provides a number of measurement signals.

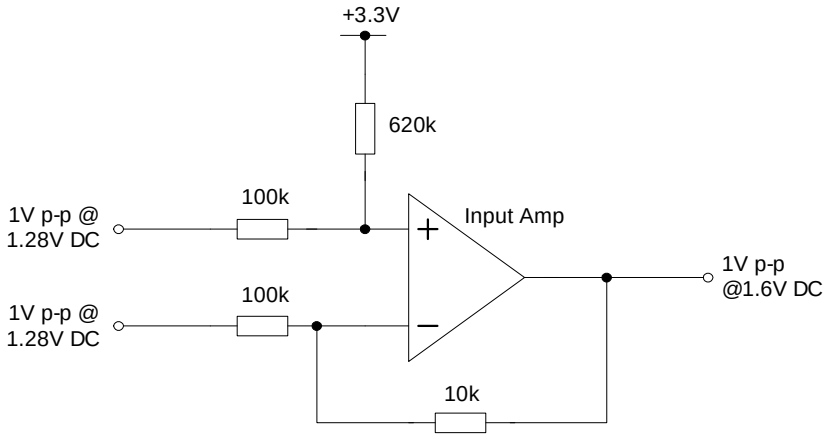


Figure 2. CMX998 Input Amplifier Configuration

3.3 CMX998 Error Amplifiers

The standard EV9980 error amplifier configuration is used in the tests and is shown in Figure 3. The 1st Pole is at ~16kHz, the 2nd Pole is at ~32kHz and the Zero is at ~320kHz. These can be changed for other user requirements.

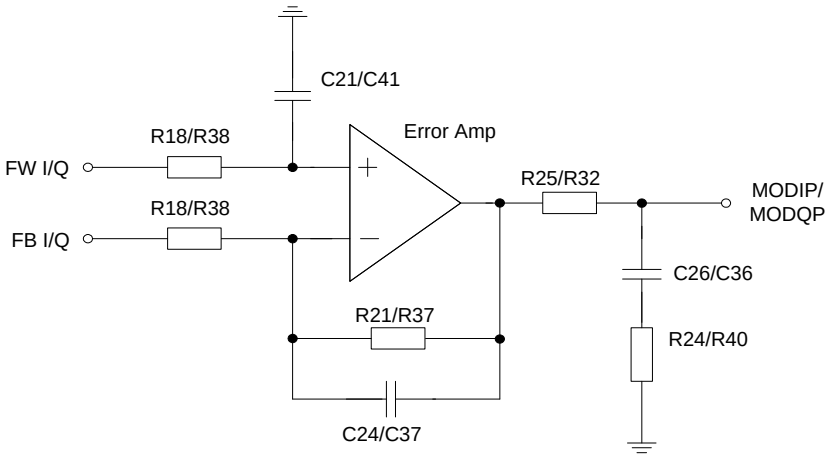


Figure 3. CMX998 Error Amplifier Configuration

3.4 Other EV9980 Hardware Configuration

The PA control level shifting op-amp, U4, must be altered by changing R72 to 5.1k ohms, allowing full-scale operation when driven from the CMX981 AUXDAC1.

Typical settings for the loop using the RF2175 400MHz PA are:

Forward path attenuation = 0dB
Feedback path attenuator = 13dB
Phase = 22.5°

At these settings the output power is ~+30dBm with a 1st. ACP of ~61-62dBc.

4 DC Calibration Overview

4.1 Cartesian Loop Overview

For reference, a block diagram of a Cartesian loop circuit is shown in Figure 3. An input signal is fed through some conditioning amplifiers and then passed to the error amplifiers. The error amplifiers drive up-converter mixers, which in turn, provide an RF signal to drive the PA stages. The output signal is sampled by a coupler and fed to the down-converter mixers via the down-converter attenuator. The resulting baseband signal is suitably amplified and fed back to the error amplifiers. Note that there are two channels, one for in-phase and one for quadrature signals, to the left of the up-converter and down-converter mixers. Also note that there is a bandwidth limiting function H(s) associated with each error amplifier.

An RF local oscillator signal is supplied to the two mixers. The LO is usually fed, with optimum signal-to-noise, to the up-converter mixer and fed to the down-converter mixer via a variable phase modulator so that the feedback phase of the loop can be adjusted.

The Cartesian loop is a DC coupled system so any DC offset generated by the various amplifiers and mixers will have the effect of causing a DC error in the modulated baseband signal at the output of the error amplifiers. This causes an increased carrier component in the RF signal, carrier leak, which, if excessive, causes the intended modulation to be distorted and therefore a method of minimising the DC error is required.

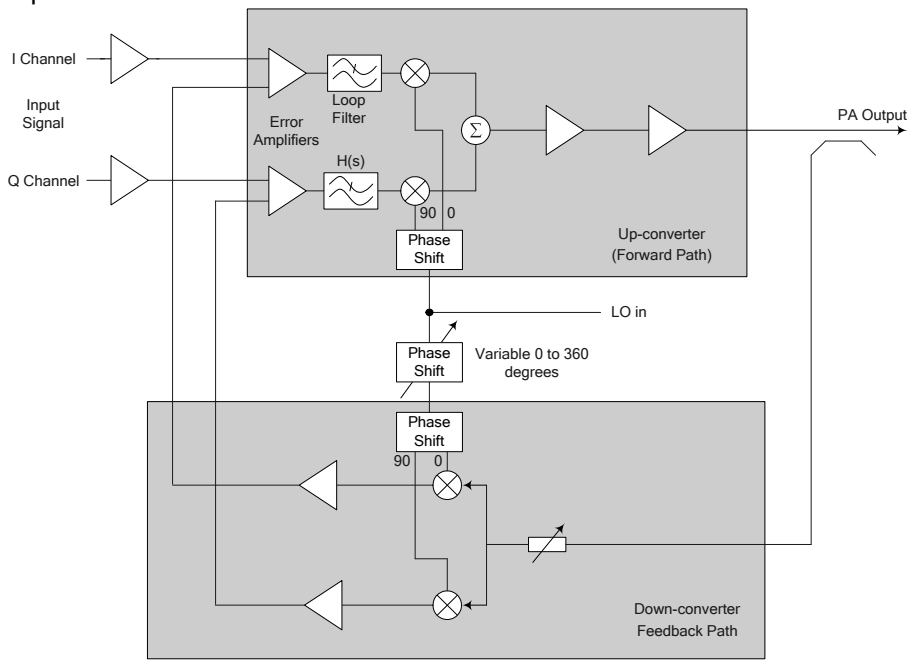
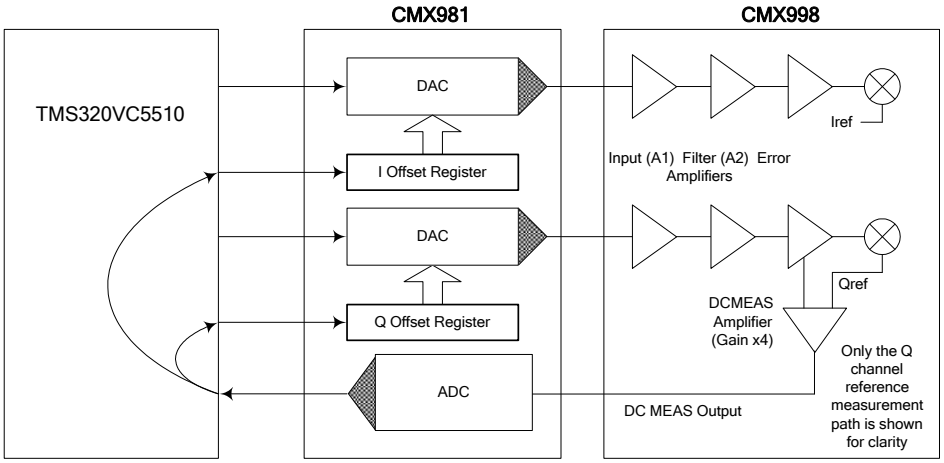


Figure 3. Block diagram of a Cartesian Loop

4.2 DC Calibration Loop



Note that only the Q Channel connections are shown for simplicity. The I Channel can also be connected to the DCMEAS Amplifier.

Figure 4. DC calibration loop

DC offset on the modulating I and Q inputs can be determined using the DCMEAS output on the CMX998. This offset can then be compensated by adjusting the modulating I and Q input signals. The DCMEAS output has four different measurement conditions that can be selected by writing commands to the AUX Control Register ($07). The following conditions can be measured; Iref, Qref, Idc and Qdc. Iref and Qref should be very similar or identical. The error in the I channel, for example, can be easily calculated by measuring the difference between Iref and Idc, then the DC at the I input can be adjusted to compensate for the errors found to attempt to make Iref = Idc. The same process can be applied to the Q-channel.

The error signal from the DCMEAS output is sampled by the Auxiliary ADC in the CMX981. A correction is computed that is then loaded into the registers for offset correction, TxIOffset and TxQOffset, of the Sigma-Delta DACs in the CMX981. The correction is applied to the I and Q inputs of the CMX998, completing the loop.

The CMX998 Data Sheet gives a detailed description of the DCMEAS circuitry and operation.

4.2.1 Open and Closed Loop

The use of the terms open loop and closed loop can be a little confusing in the context of a CFBL using the CMX998. Open loop is the case when the feedback loop is not operational, i.e. some element in the loop is disabled preventing the feedback action occurring. Closed loop occurs when all elements in the loop are active and the system is operating as a feedback loop.

Open loop mode is normally facilitated in two ways:

- Turning the PA ‘off’
- Opening the ‘loop switch’

The loop switch can be used to break the loop for various functions including self-test or loop phase calibration. The switch is configured so that when in ‘open loop’ mode the gain of the error amplifier is reduced. Thus ‘open loop’ mode and ‘closed loop’ mode are often taken as referring to the setting of the ‘loop switch’. In this way the loop can be in ‘closed loop mode’ but actually ‘open loop’ i.e. the loop switch is closed but the PA is ‘off’. This state is quite a useful state as it allows DC calibration without the emission of high levels of signal. Thus calibration may be performed prior to the start of transmission without the need for taking air-interface time to perform the function.

4.2.2 Gain reduction in the error amp

The high error amplifier gain in normal loop operation means correction steps are course if the system is in 'closed loop mode' with the PA 'off'; thus it is difficult to determine the correct correction rapidly (see section 6.1). As the gain reduction and breaking of the loop functions cannot be separated using the loop switch function this gain reduction method can not be used in closed loop mode. As an alternative the CMX998 includes the option to reduce the error-amp gain by switching in an internal 1kΩ feedback resistor across the error amplifier. Where the 'normal' gain of the error amplifier is 100 (40dB) and with the internal switch closed this reduces close to unity.

4.2.3 DC Null Steps

The following sequence explains the DC nulling process for closed loop operation. This assumes that loop feedback phase has already been established and correctly set.

Step	Action	Result
1	Set the CMX981 Sigma-Delta DACs to an initial state, either last known DC null value or nominal 'zero'. PA is disabled but all other parts of the Cartesian loop should be enabled, loop switches closed and Error Amplifier gain Reduction mode selected.	Set up initial conditions. Note: Ensure all circuits have had time to power 'on' and settle before starting DC calibration. This includes the local oscillator, which should be settled on the correct frequency prior to calibration (changes in frequency can have an effect on the level of DC offsets).
2a	Measure the value of modulator I and Q channel references (MODIN and MODQN – see Error Amplifier configuration in EV9980).	
2b	Perform DC nulling on I and Q channels involving sampling and correction of errors to equalize values (MODIN=MODIP & MODQN=MODQP). Note: the correction voltage may be different in the I and Q paths.	Detail of how to do this can be found in Sections 5 and 6.
3	Switch the Error Amplifier gain Reduction back to normal. (Register $02, b0 = 0).	There will be a small error between the Error Amplifier input and the feedback path signal due to quantisation errors. As the Error Amplifier typically has a high DC gain (e.g. 40dB), this can result in the Error Amplifier output rapidly drifting to a voltage rail.
4	Apply feedback by ramping up the PA in a controlled manner.	A controlled application of feedback will allow the Error Amplifier output to reach its closed loop operating value in a controlled way (assuming it is previously far from that condition). This will allow the loop to turn on without excessive transients.

Table 2. Calibration and Control Sequence

5 Software Operation

Software running on the DSK uses the FSB to control the EV9810 and C-BUS to control the EV9980 as detailed in Section 2.1. The software controls the CMX981 Sigma-Delta DACs for modulation and DC correction, the AUXADC1 to sample the DCMEAS output and the AUXDAC1 to control the power up of the PA. On the CMX998, the following registers are used to configure the loop as described in Section 3.3:

1. The General Control register, to power up relevant parts of the IC and control the Error Amplifier gain Reduction.
2. The Phase Control Register.
3. The Gain Control Register.
4. The Aux Control Register to set up the DCMEAS output.

Some typical settings for the CMX998 based on the loop configuration described in Section 3.3 are:

1. *General Control Register* ($02) = 0xFC. Enables the forward path, feedback path, Vbias, filter amps (Amplifier 2), Input Amplifiers and Error Amplifiers. Error Amplifier gain Reduction is normal and loop mode is closed. This condition is normal when DC calibration is not taking place. Writing 0xFD will reduce the Error Amplifier gain to allow DC calibration to take place.
2. Phase Control Register ($03) = 0x10. This indicates a phase shift setting of 22.5 degrees.
3. *Gain Control Register* ($04) = 0x0020. This indicates a forward path attenuator setting of 0dB and a feedback attenuator setting of 13dB.
4. *AUX Control Register* ($07): b7 switches the DCMEAS circuits on and off, b0 and b1 select the measurement mode.

5.1 Software Description

The control software for DC calibration, followed by data transmission, is scheduled at the rate at which the DSP receives CMX981 I and Q sample pairs. The fact that this is receive data is not relevant, it simply allows the software to be scheduled at a convenient rate and one that is synchronous to the CMX981 internal clocks. I and Q sample pairs are received at 144kHz (6.94µs apart).

After the system is initialised, basic calibration operation involves the following steps:

1. Select the CMX998 measurement to make (Iref, Idc, Qref or Qdc).
2. Start the CMX981 AUXADC conversion.
3. Wait for the conversion to complete and then retrieve the result.
4. Compute a correction and apply it.
5. Wait for the correction to take effect and the DCMEAS signal to be stable.
6. Loop from step 2 as required to achieve greater accuracy.

The software measures Iref and Qref once only and assumes that they remain constant during several repeat calibrations.

Two parts of the sequence described above take significant time:

1. AUXADC Read.
2. Settling of the corrected IQ output signals.

The AUXADC is run in single shot mode. This allows direct control of the time that conversion actually takes place. It solves the problem that would exist in continuous mode where reading the two halves of the AuxAdcData registers could produce an inconsistent value if a conversion were to complete between reading one half of the result and reading the other half of the result. The disadvantage of the single shot mode is that the conversion takes longer than when running in continuous mode. Each conversion takes 88 MCLK periods to complete but in single shot mode, two

further 88 MCLK periods are required to activate the AUXADC. The total time for a conversion is therefore 28.65µs at MCLK = 9.216MHz. This will take just over 4 x 6.94µs periods to complete.

From having implemented a DC correction, it takes time for the adjusted DC measurement from the CMX998 to settle at its corrected DC value. This is due to the accumulated group delay of the various filters in the signal path. The length of the delay is described in Section 6.1. By alternating between calibrating Idc and Qdc, it is possible to structure the software so that the AUXADC samples Qdc while the previous Idc correction is settling. This reduces calibration time and removes the need to run the AUXADC inputs faster.

The resulting software (called at 6.94µS rate as described above) is outlined as follows:

START
If count = 0
  Write to the CMX998 to select DC measurement of Idc.
  Write to the CMX981 to request an AUXADC conversion.
If count = 1 to 5
  Do nothing (waiting for the conversion to complete).
If count = 6 or 7
  Read the result of the conversion back from the CMX981, two words need to be read.
If count = 8
  Compare previous AUXADC reading for Iref to Idc and compute a correction.
  Write to the CMX981 TxIOffset register to implement the correction.
If count = 9
  Write to the CMX998 to select DC measurement of Qdc.
  Write to the CMX981 to request an AUXADC conversion.
If count = 10 to 14
  Do nothing (waiting for the conversion to complete).
If count = 15 or 16
  Read the result of the conversion back from the CMX981. Two words need to be read.
If count = 17
  Compare previous AUXADC reading for Qref to Qdc and compute a correction.
  Write to the CMX981 TxQOffset register to implement the correction.
Increment count
END

The calibration may be carried out in closed loop mode with the Error Amplifier gain Reduction enabled (low gain mode) or set to normal (high gain mode). Initial calibration is best done in low gain mode because in high gain mode the error may be large enough to cause the measured DC offset to be saturated. In the majority of cases, calibration in low gain mode is accurate enough. Section 6 provides more detail on the effects of the selected gain mode.

Computing the correction is fairly straightforward. The measured Iref minus Idc is proportional to the correction required, assuming that the measurement is not saturated. The constant of proportionality is dictated by the external components but should remain reasonably constant.

When the calibration was done in low gain mode, it was observed that the ideal correction differs slightly from the ideal correction in high gain mode. It was concluded that the DC offset present changes slightly between the two gain modes but appears in a predictable manner. This is also thought to be a product of the board design and external component values.

Therefore a constant must be introduced to the calculation, which becomes:

Required adjustment to Ioffset = ERROR_GAIN*(Iref - Idc + ERROR_OFFSET)

Where;

ERROR_GAIN is the constant of proportionality and ERROR_OFFSET is the difference between the ideal correction in low and high gain modes.

The values for ERROR_GAIN and ERROR_OFFSET are unlikely to be 100% accurate. Repeating the calibration process on the resultant output will reduce the error further. The resulting behaviour models a slightly overdamped feedback loop.

It can be seen that one calibration pass takes 18 x 6.94us = 125µs, so each additional pass will add another 125µs.

A listing of the code is given in Section 9.

5.2 Making the Code Faster

The execution of the loop described in the previous section has not been optimised for speed and at least two and probably three counts can be saved. The execution of the AUXDAC1 read (counts = 6 or 7 and counts = 14 or 15) can be halved to a single 6.94µs period with a modified read routine. This reduces the cycle time to 16 counts or 111µs. If the read of the Q-channel is initiated before the calculation of the I channel correction, then a further cycle is saved. This saves a single count in a single stage process or 3 counts in a two-stage process.

The net result is a reduction to 104µS for a single cycle or 194µS for a two cycle process.

It should be noted that these are processing times and analogue delays should be added (see Section 6.4).

6 Observation and Results

6.1 Initial Observations

During the DC calibration cycle the nominal state of the CFBL is: closed loop[2], PA off[3] and the Error Amplifier gain Reduction set to normal (high gain mode). In this state, a saturated condition at the Error Amplifier output is likely because of the high gain. An iterative process is therefore required to step the result towards the correct value but this can make it difficult to perform a DC calibration quickly. A correction is applied using a 12-bit DC offset added to the I or Q outputs[4] and it has been found that approximately 10 steps of this offset can cause the DCMEAS signal from the CMX998 to step from positive to negative saturation. In this loop configuration it has been found that, provided the error is within a small range[5], it can be corrected within a few 100µS, but care must be taken to avoid oscillation due to the high gain.

A DC calibration with the Error Amplifier gain reduced[6] is much more stable. The value sampled on the Auxiliary ADC is a 10-bit result and approximately 1 on the Auxiliary ADC equals 1 on the offset to correct it. This is due to the gain of 4 in the DCMEAS circuitry countering the 1/4 resolution of the converter. Therefore, given a measurement of Iref, Qref and Idc, Qdc it is simple to adjust I and Q inputs to compensate for the error between Iref and Idc or Qref and Qdc.

In high gain mode it has been found that after making a step change to the DC level on I or Q, it takes ~70µs to result in a new, stable level that can be measured on DCMEAS. This is a combination of the time it takes to drive the CMX981 output (RC filter at the output) and the CMX998 input and measurement circuitry. The same test in low gain mode takes less than 10µs. Switching the

[2] The CMX998 mode is 'closed loop', i.e. the feedback path is connected to the Error Amplifier. This is essential so that when normal operation commences the dc errors of the feedback path are included in the calibration.

[3] This is achieved by adjusting the PA control voltage, controlled by CMX981 DAC1, to a low state (i.e. 'off').

[4] Using the CMX981 Transmit I and Q Channel Offset Registers.

[5] $2.5 \times 10 / 2^{12} \approx 6\text{mV}$ where 2.5 is the connection full-scale voltage.

[6] This can be done using the Error Amplifier Gain Reduction bit in the CMX998 (General Control register $02, bit 0).

DCMEAS output between measurements i.e. Idc and Qdc requires ~7µs of settling time. Other timings involved with obtaining the DC error measurements are:

1. A CMX981 Auxiliary ADC single shot conversion takes 3 x 88 / 9.216MHz = 28.65µs, (the first 2 are just the wake up period).
2. Serial comms to the CMX981 or CMX998 can be done at ~1.8µs per transfer.

6.2 Results

With the default CMX981 power-up state, the resultant carrier level is ~+6dBm because no correction is applied (note full output power of the CFBL is ~+30dBm mean power with Pi/4-DQPSK modulation).

Typical test scenarios have been evaluated to see the effect on the carrier level after a calibration sequence. Note in all cases the loop is closed and 3 iterations of sampling DCMEAS and calculating the result are completed. After the calibration procedure, the PA is enabled to evaluate the effectiveness of the calibration. Results are as follows:

Low gain mode selected. No ERROR_OFFSET factor included in S/W calculation:
I offset = -50
Q offset = -107
Resultant Carrier Level ~ -8dBm (-38dBc)
Low gain mode selected. ERROR_OFFSET factor[7] of 7 for I & Q:
I offset = -60
Q offset = -108
Resultant Carrier Level ~ -25dBm (-55dBc)

High gain mode selected:
I offset = -58
Q offset = -108
Resultant Carrier Level ~ -28dBm (-58dBc)

Initially, low gain mode is selected (No ERROR_OFFSET factor) and a calibration cycle is performed, then high gain mode is selected and a further calibration cycle performed:
I offset = -57
Q offset = -107
Resultant Carrier Level ~-22dBm (-52dBc)

From these results and the comments made about timings in Section 6.1, it can be seen that the optimal calibration method requires the low gain mode (Error Amplifier gain Reduction enabled) and a correction factor to be included in the calculation. The following sections investigate the consequence of altering a number of factors within the software:

1. Number of measurement and offset calculation iterations (sample and correct).
2. ERROR_GAIN factor. See Section 5.1.
3. ERROR_OFFSET factor. See Section 5.1.
4. Injecting a forced DC error on the I and Q inputs.

6.2.1 Changing the Number of Measurement Iterations

In this test scenario, the difference between 2 and 3 passes through the sample and correct is evaluated. The common test conditions are:

1. Low gain mode
2. Closed loop mode
3. ERROR_GAIN factor = 0.85

[7] See section 5.1

4. ERROR_OFFSET factor = 7
5. A forced error of 200 on I and –200 on Q (equivalent to ~90mV of DC offset).

6.2.1.1 Two Iterations of Sample and Correct

The resultant carrier level after calibration is ~-9.2dBm (full output power is ~+30dBm mean power with Pi/4-DQPSK modulation) and the software calculated offset values are I offset = -102 and Q offset = -57. Figure 5, 6 and 7 are plots of the DCMEAS output (top trace) from the CMX998 and the I or Q input (bottom trace). DCMEAS is sampled by the CMX981 approximately half way through the relevant measurement slot to allow DCMEAS to settle properly before a measurement is taken. This is roughly indicated in the plots by the left hand vertical marker bar, which indicates the position of the first measurement to be taken for I and Q. The right hand marker indicates when the second measurement pass has been completed. It should be noted that the correction is applied while the sample on the next channel is in progress.

Note: In all of the following timing plots, the trigger point is based on an arbitrary timing signal generated to allow consistent measurement of the calibration waveform. It does not represent the actual start of the calibration process.

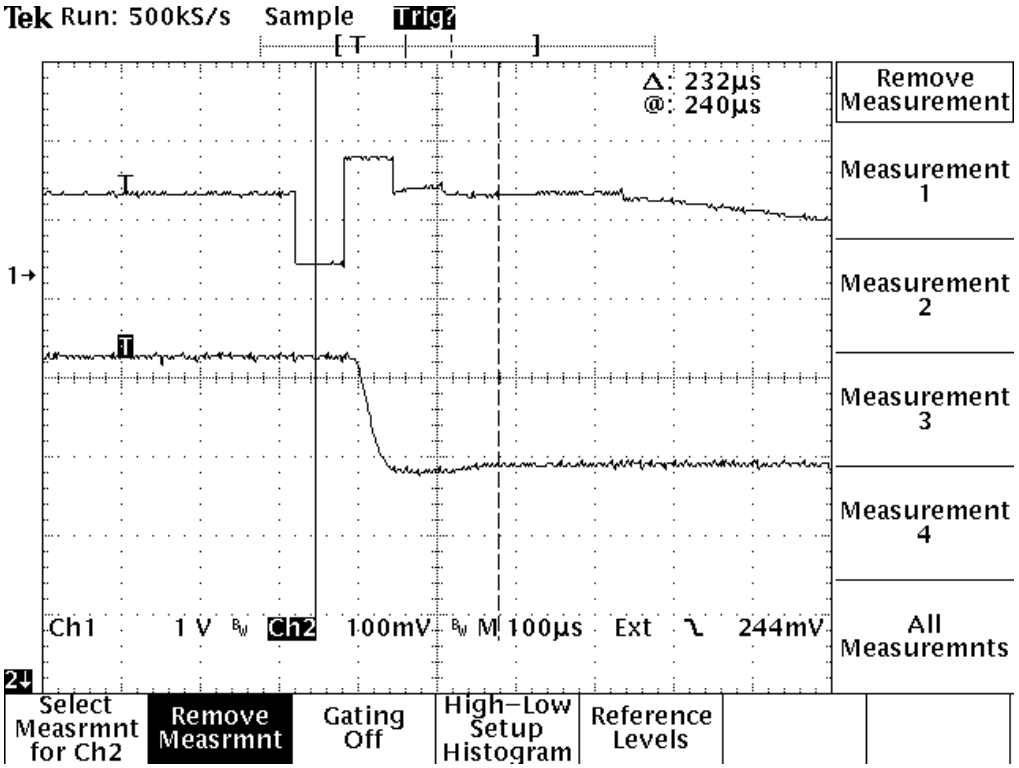


Figure 5. Plot of DCMEAS and I+ Input

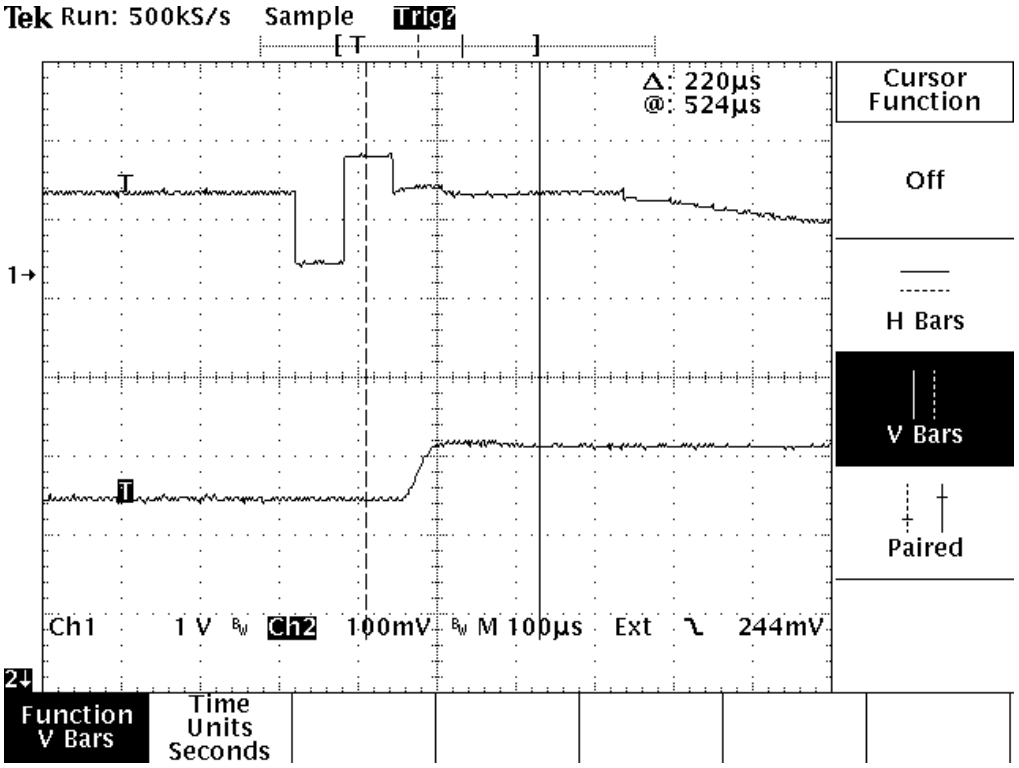


Figure 6. Plot of DCMEAS and Q+ Input

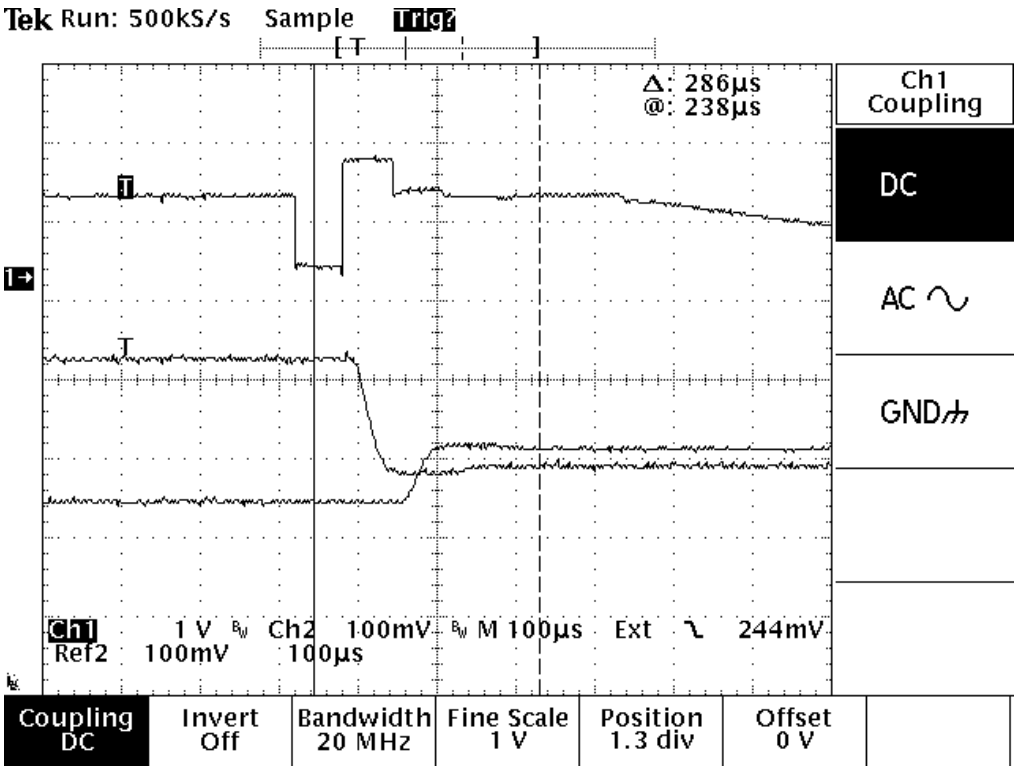


Figure 7. Plot of DCMEAS and I+ & Q+ Inputs after 2 iterations of sample and correct

From Figures 5 and 6 it can be seen that it takes ~230µS to run two iterations of the measurement on I or Q, and Figure 7 shows that it takes only ~290µS to complete correction of both I and Q as the two tasks are overlapped. (See also Section 6.4).

Figures 8, 9, 10, 11 and 12 are plots of the resultant carrier level (easily seen while transmitting tones), the TETRA ACP performance, eye diagram, constellation and symbol/error table respectively. The plots show that reasonable DC calibration has been completed and that the TETRA performance is good with this level of calibration.

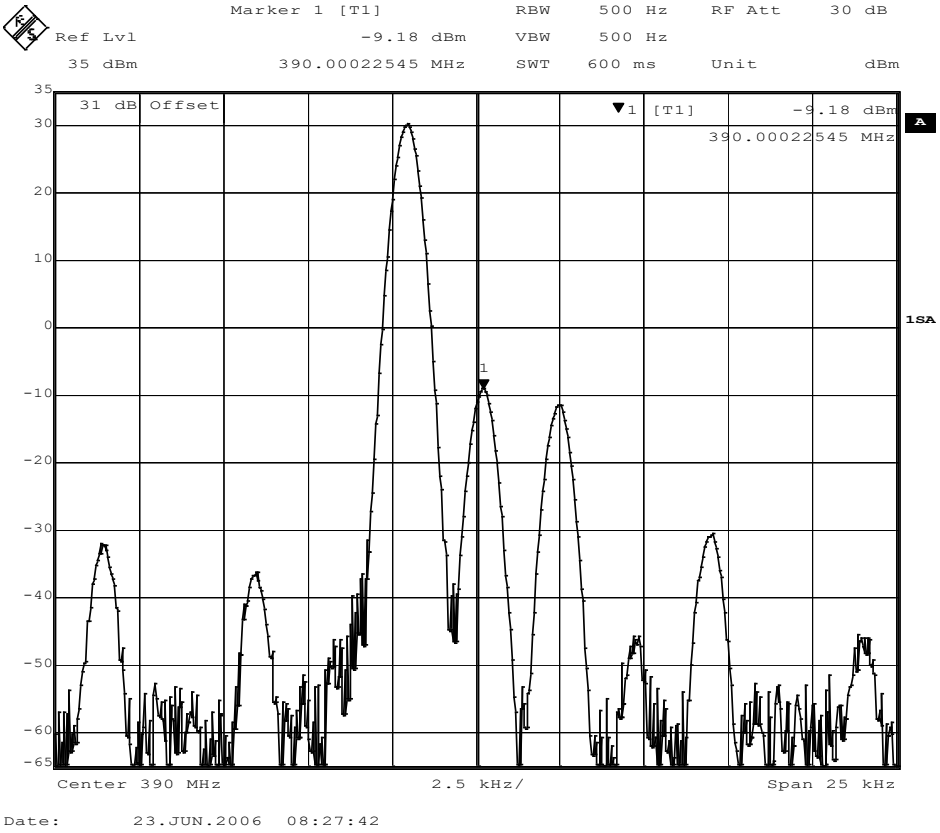


Figure 8. Resultant Carrier level after 2 iterations of sample and correct

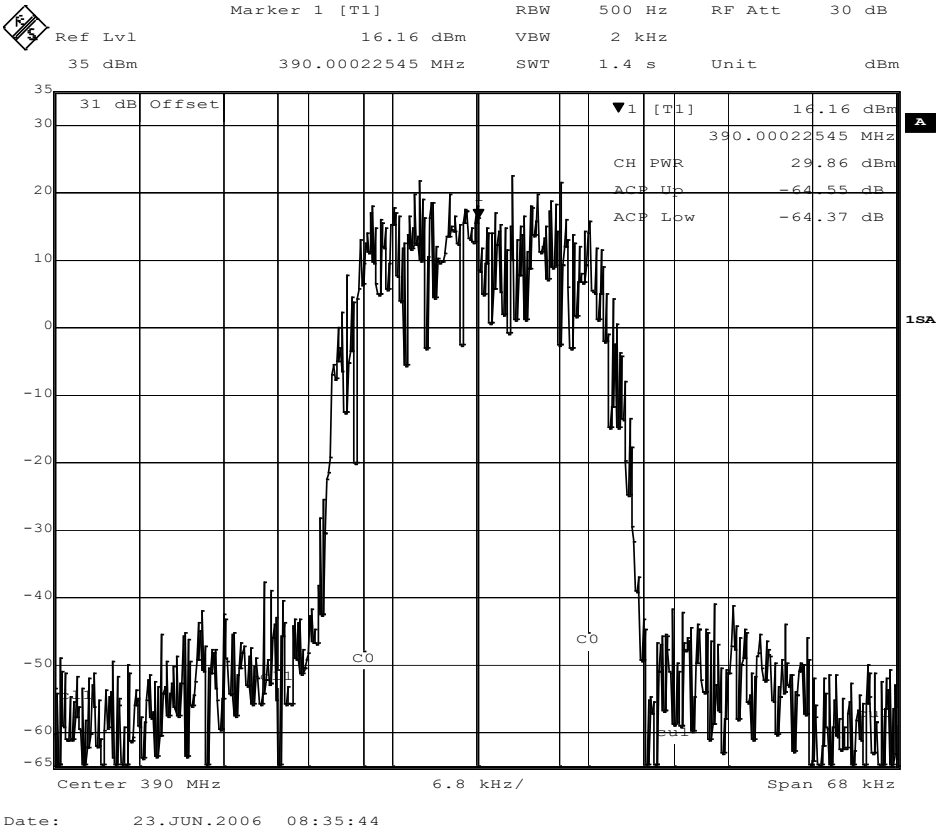


Figure 9. TETRA ACP Performance after 2 iterations of sample and correct

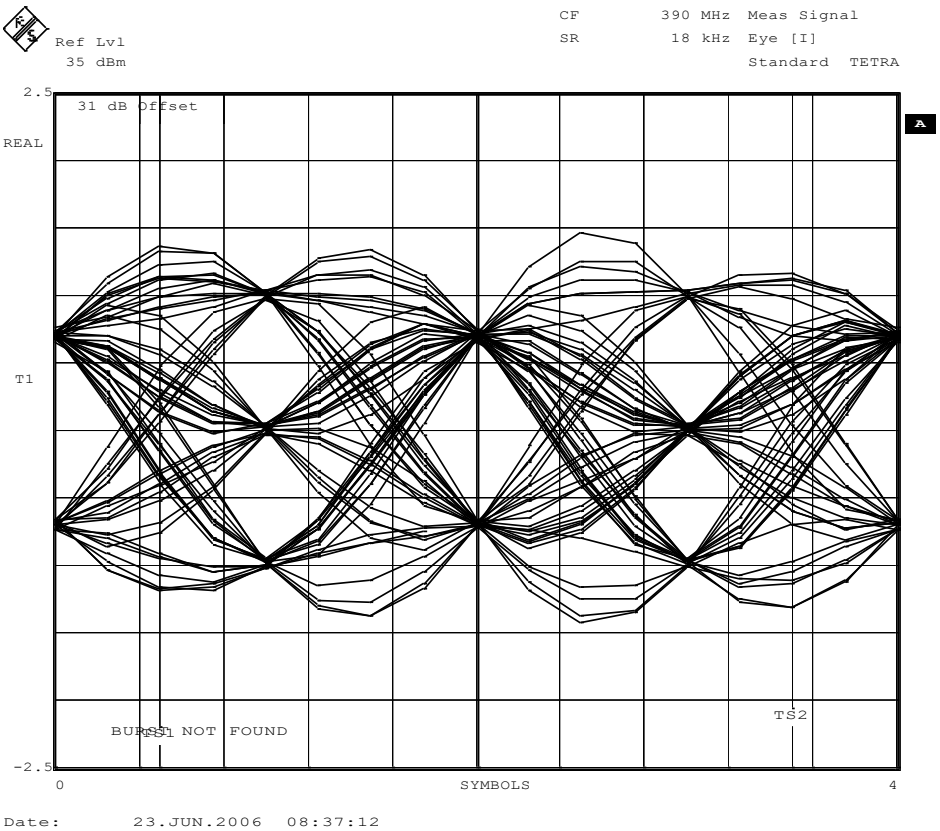


Figure 10. Eye Diagram after 2 iterations of sample and correct

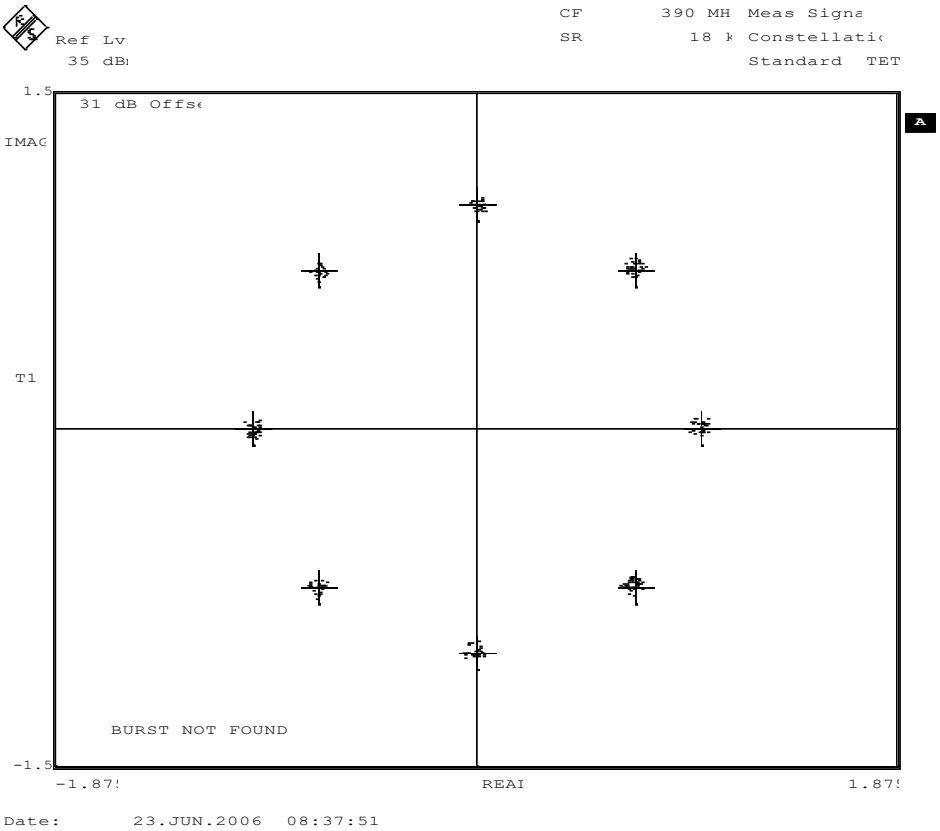


Figure 11. Constellation Diagram after 2 iterations of sample and correct

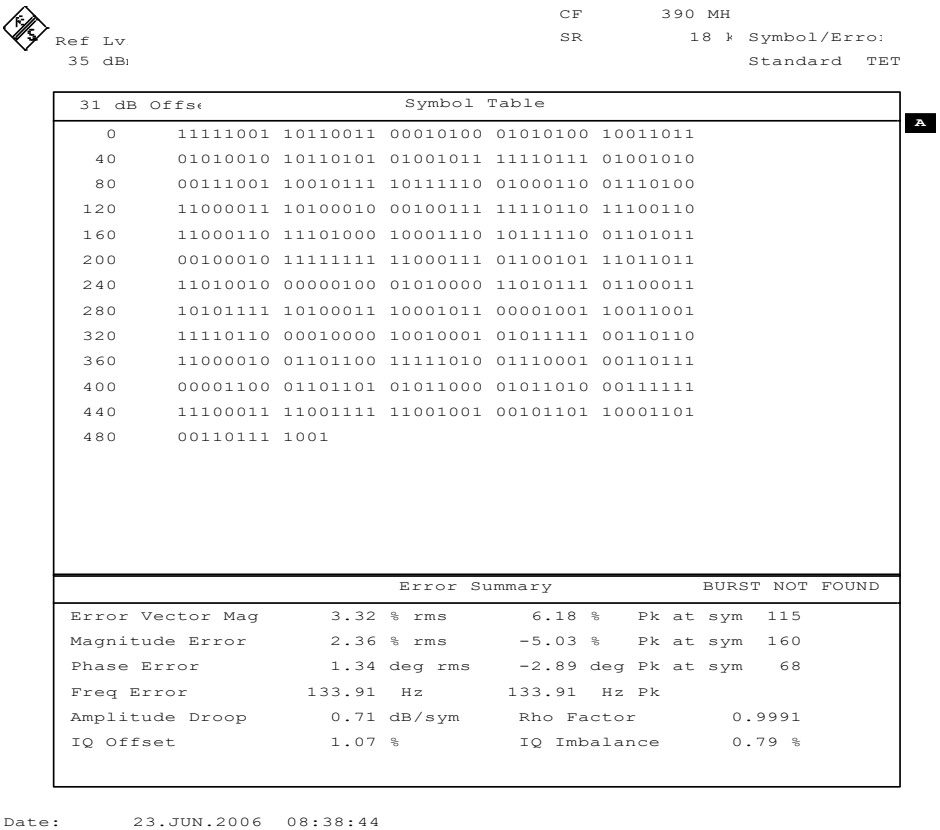


Figure 12. Symbol/Error Table after 2 iterations of sample and correct

6.2.1.2 Three Iterations of Sample and Correct

The resultant carrier level after calibration is ~-24dBm (full output power is ~+30dBm mean power with Pi/4-DQPSK modulation) and the software calculated offset values are I offset = -98 and Q offset = -57. Figure 13 shows it takes ~400µS to complete 3 iterations of sample and correct. Figures 14 and 15 show the resultant carrier level after calibration and the symbol/error table using TETRA modulation.

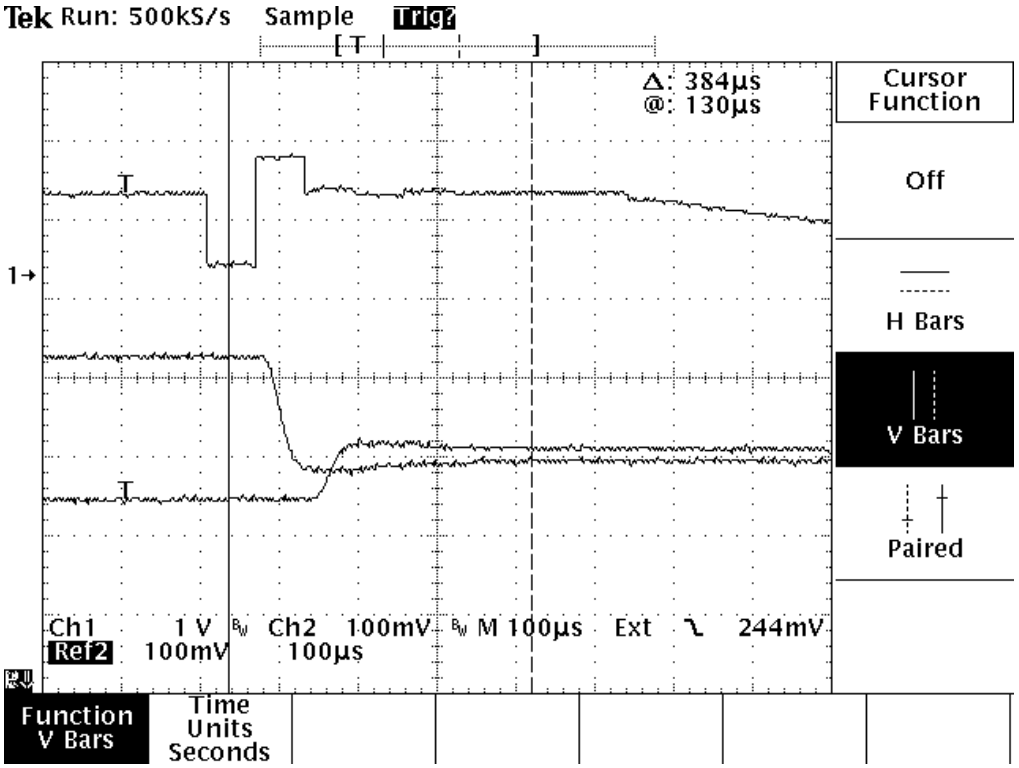


Figure 13. Plot of DCMEAS and I+ & Q+ inputs after 3 iterations of sample and correct

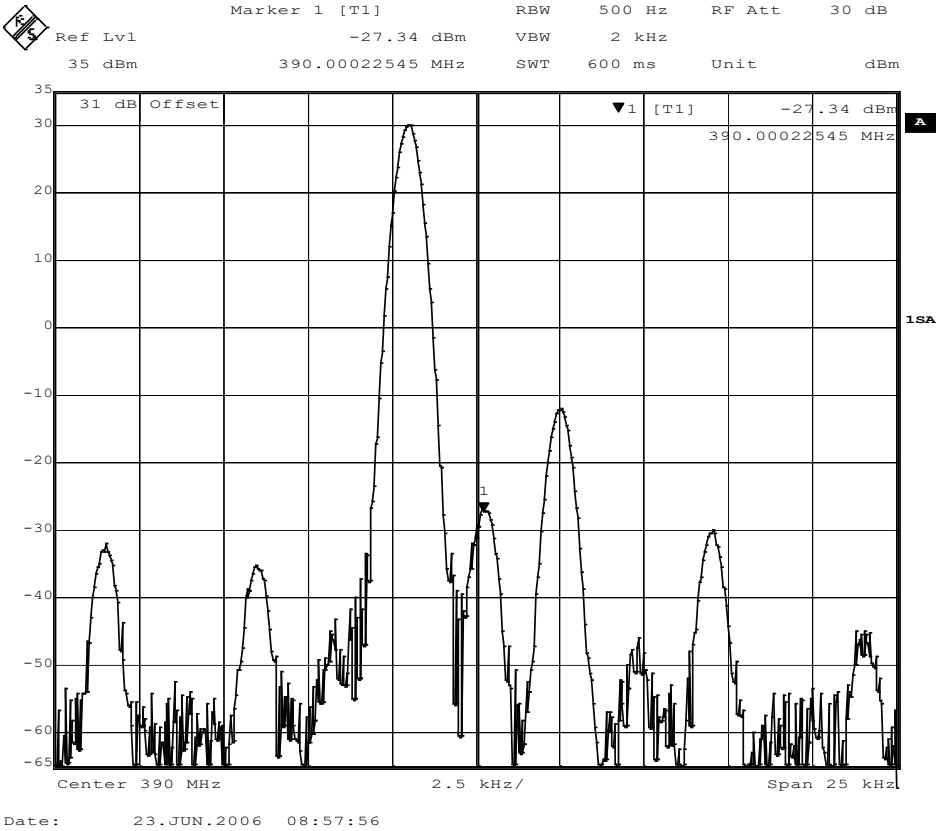


Figure 14. Resultant Carrier Level after 3 iterations of sample and correct

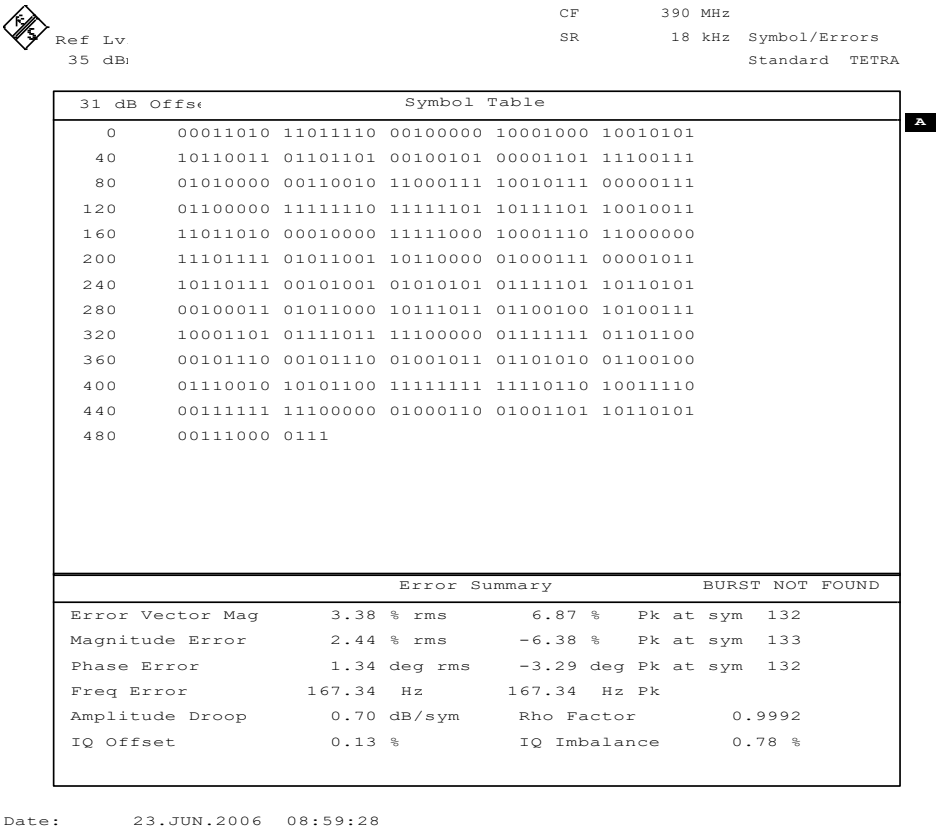


Figure 15. Symbol/Error Table after 3 iterations of sample and correct

So it can be seen that, as expected, more iterations of sampling the DCMEAS and applying correction to the I and Q inputs can improve the calibration accuracy, although with only 2 iterations the calibration is sufficient. The major consequence of doing more passes is the time taken: 2 passes takes ~290µS and 3 passes takes ~ 400µS.

6.2.2 Changing the ERROR_GAIN Factor

In this test scenario, the number of passes through the sample and correct is always 2. Common test conditions are:

1. Low gain mode
2. Closed loop mode
3. ERROR_OFFSET factor = 7
4. A forced error of 200 on I and –200 on Q (equivalent to ~90mV of DC offset on I and Q).

Table 3 shows the resultant carrier level after a calibration has been completed using different values of ERROR_GAIN factor.

ERROR_GAIN Factor	Resultant Carrier Level (dBm)
0	+21.1
0.6	-7
0.65	-20.5
0.7	-19
0.75	-11.4
0.8	-10.33
0.85	-11.5
0.9	-17.4
0.95	-13

Table 3. Varying ERROR_GAIN Factor

Figure 16 compares an ERROR_GAIN factor of 0.65 and 0.85. The lower traces of I+ show how changing the ERROR_GAIN factor alters the final I value calculated. It appears that an ERROR_GAIN factor of 0.65 or 0.7 achieves the most accurate calibration.

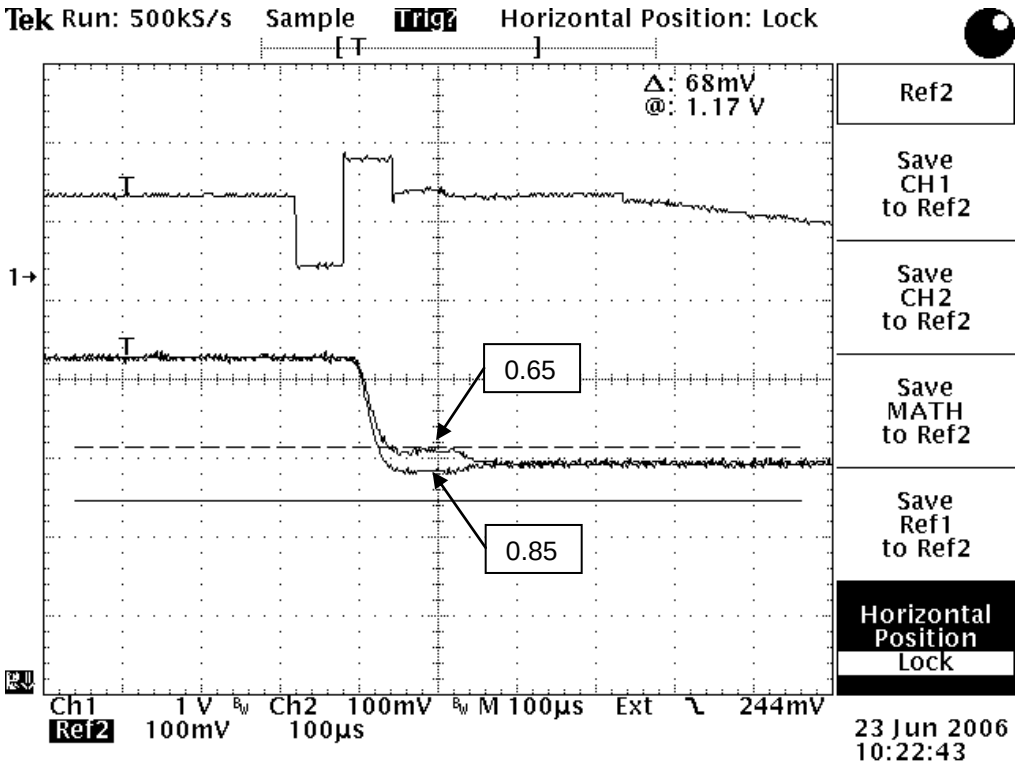


Figure 16. Varying the ERROR_GAIN Factor between 0.65 and 0.85

6.2.3 Changing the Amount of Forced DC offset on the I and Q Inputs

In this test scenario, the number of passes through the sample and correct is always 2. Common test conditions for the results are:

1. low gain mode
2. closed loop mode
3. ERROR_OFFSET factor = 7
4. ERROR_GAIN factor = 0.85.

Table 4 shows the resultant carrier level after a calibration has been completed using different levels of forced DC offset on I, Q, or both I and Q.

Forced I/Q Offset		Resultant Carrier Level (dBm)	DC Offset to be corrected (mV)	
I	**Q**		**I**	**Q**
0	0	-17.3	-12	-16
0	100	-22	-12	-60
100	100	-22	-56	-60
-100	100	-23	34	-60
-50	50	-21	12	-38
-150	150	-20	56	-84
-200	200	-16	78	-106
200	-200	-10	-102	74
150	-150	-16	-80	54
100	-100	-21	-56	28
200	200	-11	-102	-106
-200	-200	-14	78	74
300	-300	-10	-147	116
-300	300	-9	125	146

Table 4. Varying the amount of forced DC offset on I and Q inputs

With no forced DC offset on I and Q, it is possible to see the residual DC offset of the test system which is ~12mV on I and ~16mV on Q. With just the residual DC error, the carrier level after calibration is ~ -16dBm after 2 iterations of sample and correct and ~ -22dBm after 3 iterations. The results show the effect of adding varying amounts of DC offset to this. They indicate that the larger the amount of error to correct, the less accurate the calibration becomes although they also indicate that the DC can be corrected over quite a large range.

Note: adding a value of 100 on I/Q adds an error of ~44mV and, therefore, 300 corresponds to ~132mV.

Figure 17 is a plot of Q (lower two traces) being corrected with two different amount of DC offset added, one trace is Q = 50 and Q = 200. Figure 18 is a plot of I (lower two traces) being corrected with I = -50 and I = -350.

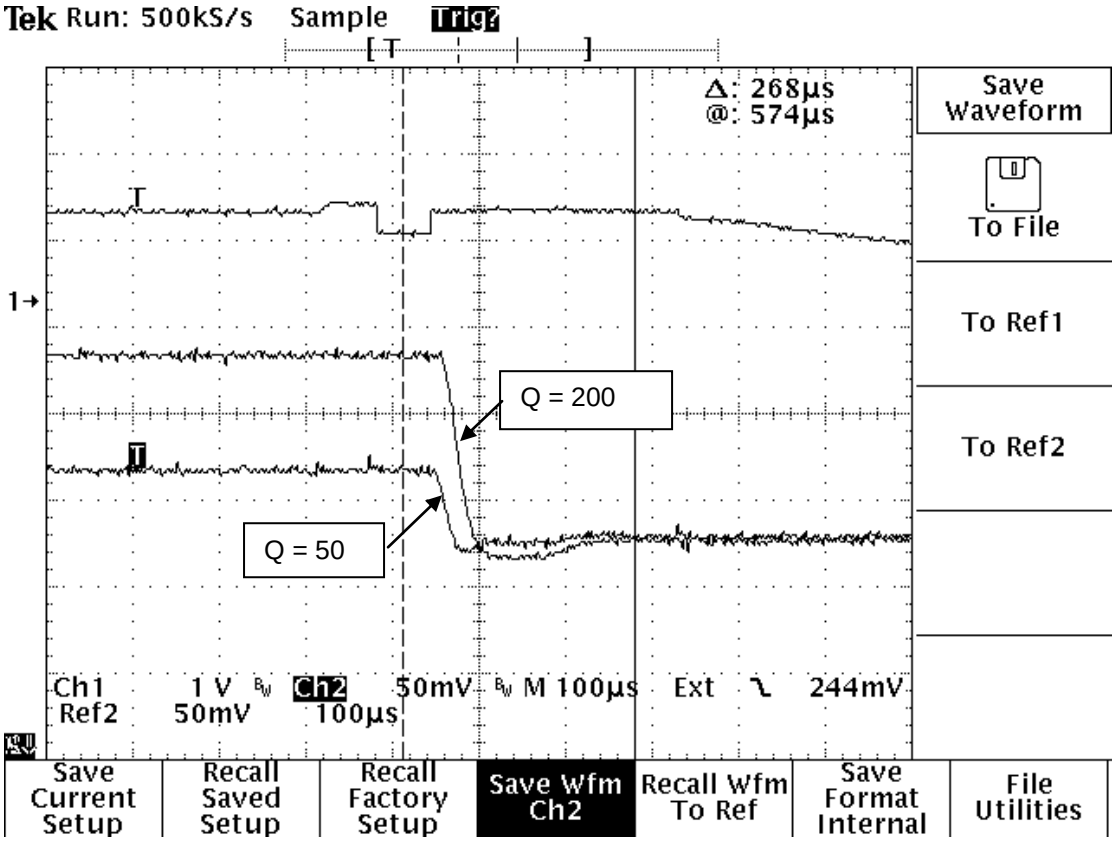


Figure 17. Varying the amount of forced DC offset on Q+ input

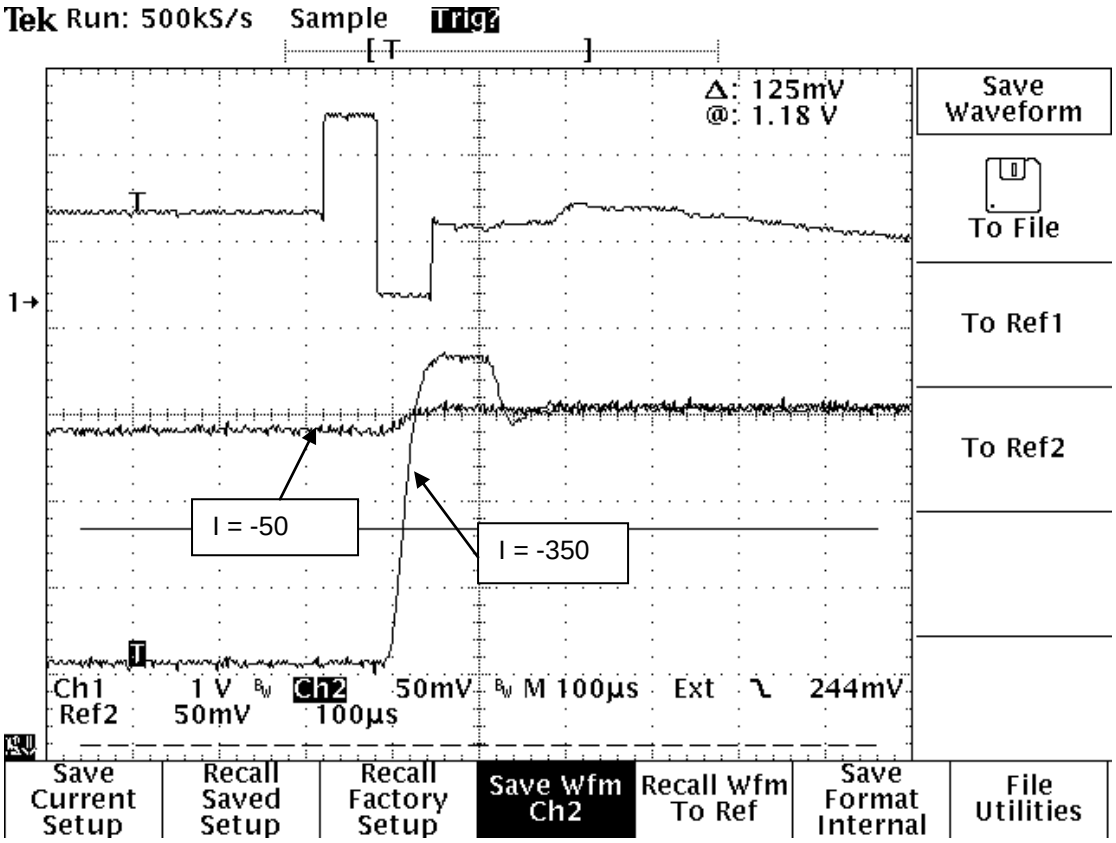


Figure 18. Varying the amount of forced DC offset on I+ input

6.2.4 Changing the ERROR_OFFSET Factor Value

In this test scenario, the number of passes through the sample and correct is always 2. Common test conditions are;

1. low gain mode
2. closed loop mode
3. an ERROR_GAIN factor of 0.85

An ERROR_OFFSET factor of 7 has been used previously. Table 5 shows the effect of changing the ERROR_OFFSET factor on the resultant carrier level. There is no forced I and Q DC offset but the system residual DC offset is present. Table 6 shows the case where a forced DC offset of I = 200 and Q = -200 is present. The results indicate that an ERROR_OFFSET factor value of 9 gives better calibration accuracy.

ERROR_OFFSET Factor	Resultant Carrier Level (dBm)
6	-13
7	-16.5
8	-20
9	-21
10	-22
11	-21
12	-19

Table 5. Varying the ERROR_OFFSET factor with just a residual DC offset

ERROR_OFFSET Factor	Resultant Carrier Level (dBm)
6	-11
7	-11
8	-11.5
9	-11
10	-10
11	-9
12	-8

Table 6. Varying the ERROR_OFFSET factor with forced offsets of I = 200 and Q = -200

6.2.5 Further Results

It has been shown that an ERROR_GAIN factor of 0.7 and an ERROR_OFFSET factor of 9 would give better calibration accuracy. Common test conditions for the results are:

1. Error Amplifier gain is low
2. The loop is closed
3. The number of passes through the sample and correct is always 2.

Table 7 shows the effect of changing the amount of DC offset on the I and Q inputs.

Forced I and Q Offset[8]		Resultant Carrier Level (dBm)
I	Q	
0	0	-22
50	-50	-24
100	-100	-25
150	-150	-18
200	-200	-16
300	-300	-15

Table 7. Varying forced I and Q offset

Figure 19 shows the correction of I and Q with the same conditions as above but with no forced DC offset. Figures 20, 21 and 22 show the nulled carrier, the TETRA ACP performance and the related symbol/error table respectively.

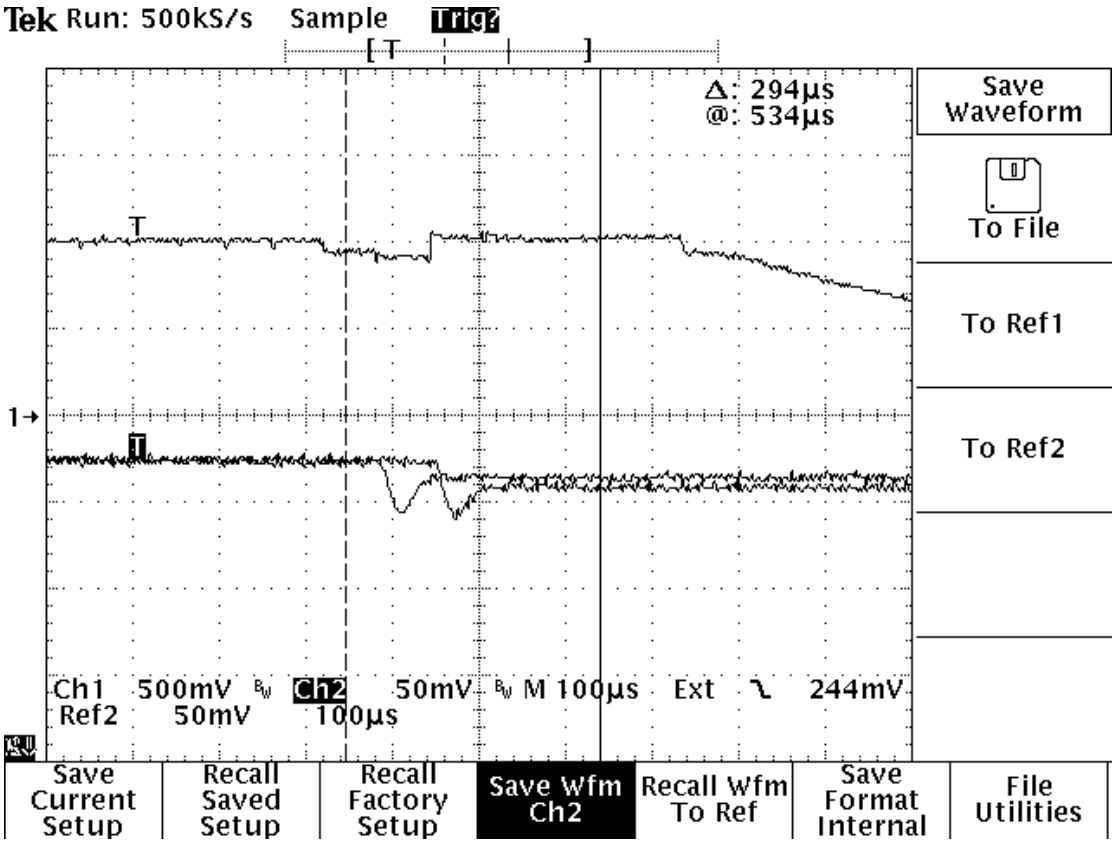


Figure 19. Correction of I and Q with residual error only

[8] Decimal value, see section 6.2.3 for more details.

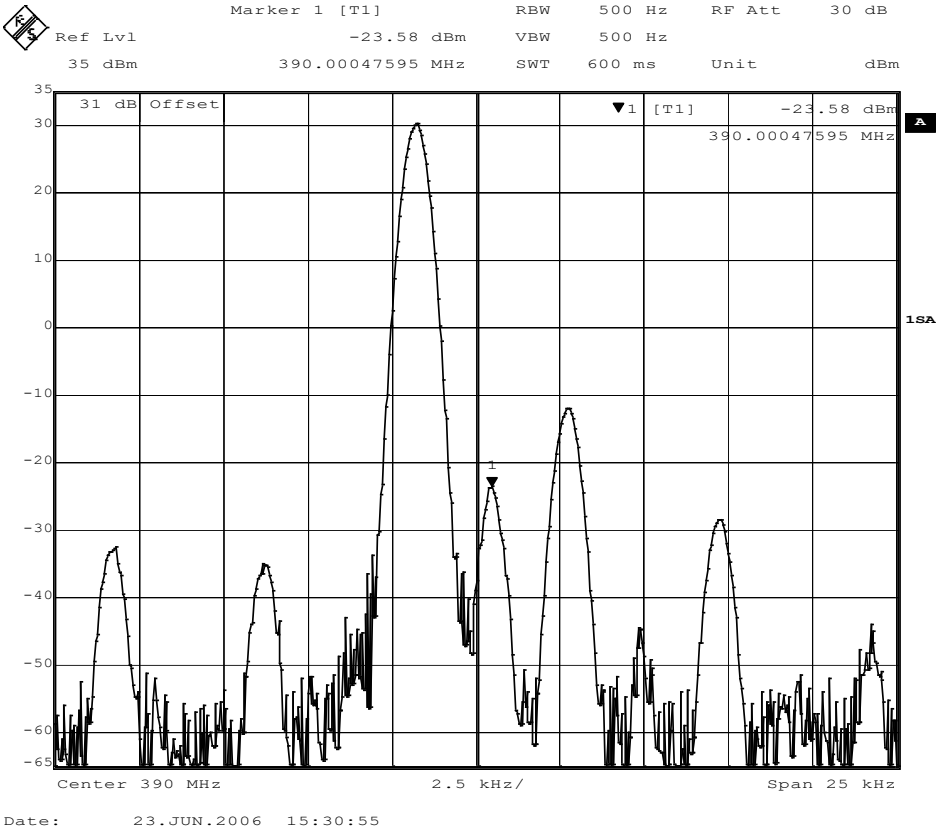


Figure 20. Nulled Carrier with residual error only

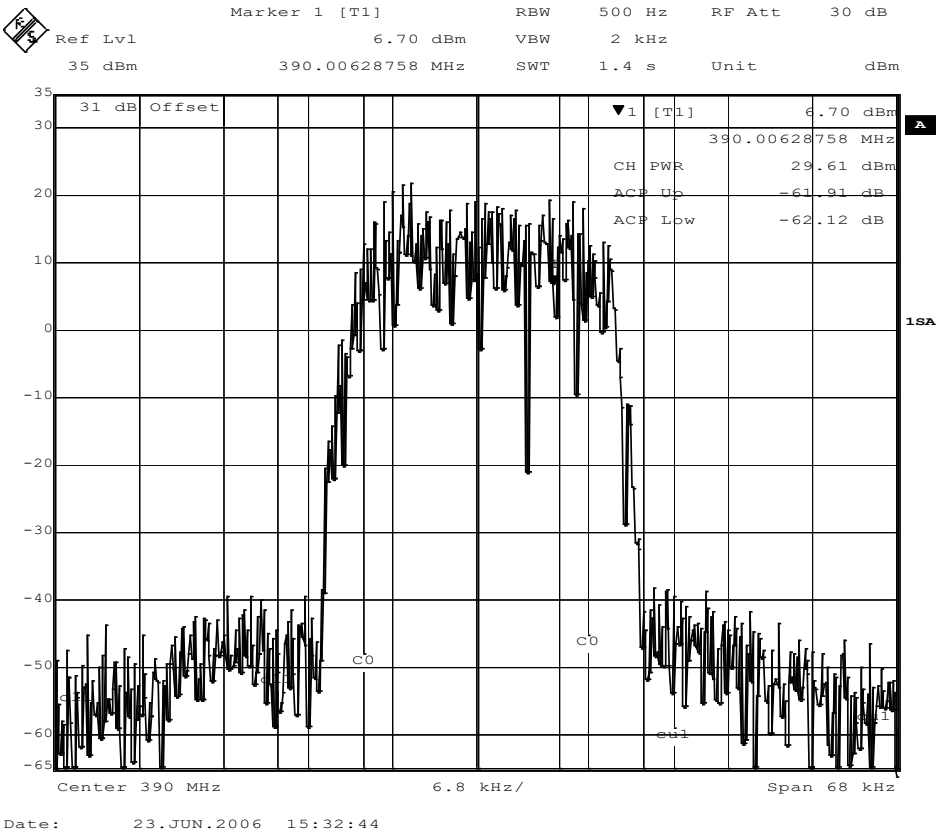


Figure 21. TETRA ACP Performance after calibration of residual error only

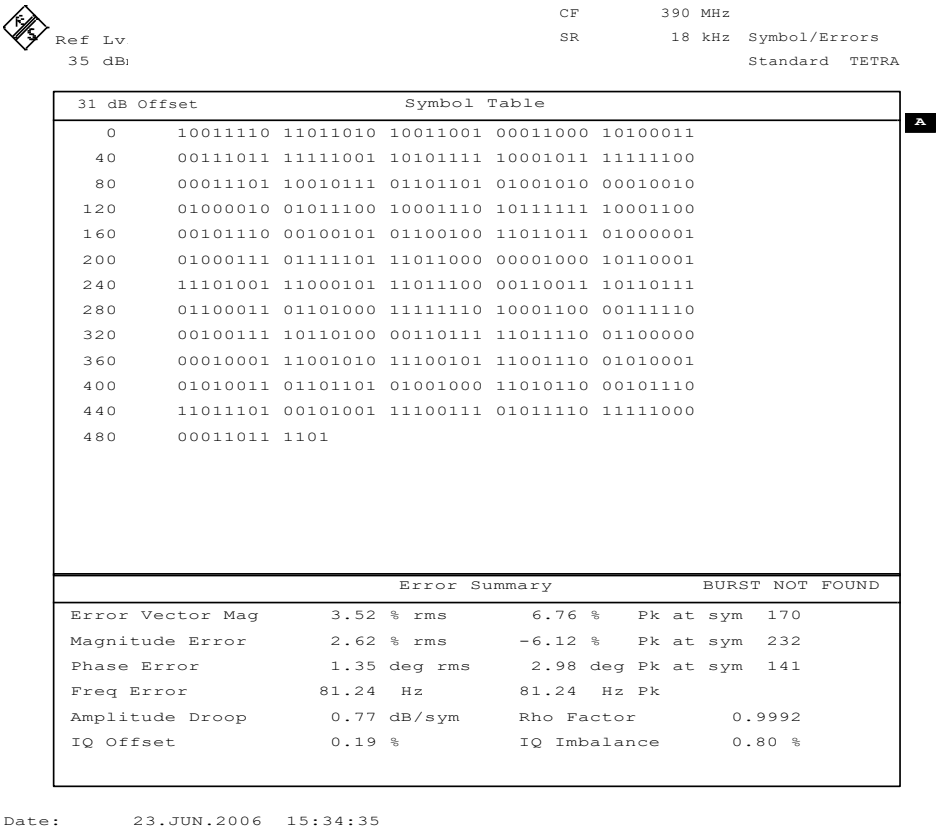


Figure 22. Symbol/Error table after calibration of residual error only

With the same test scenario as previously described a forced error of I = 200 and Q = -200 was included. Figure 23 shows the correction of I and Q, the upper trace is DCMEAS and lower traces are I+ and Q+. Figures 24, 25 and 26 show the nulled carrier, the TETRA ACP performance and the related symbol/error table respectively.

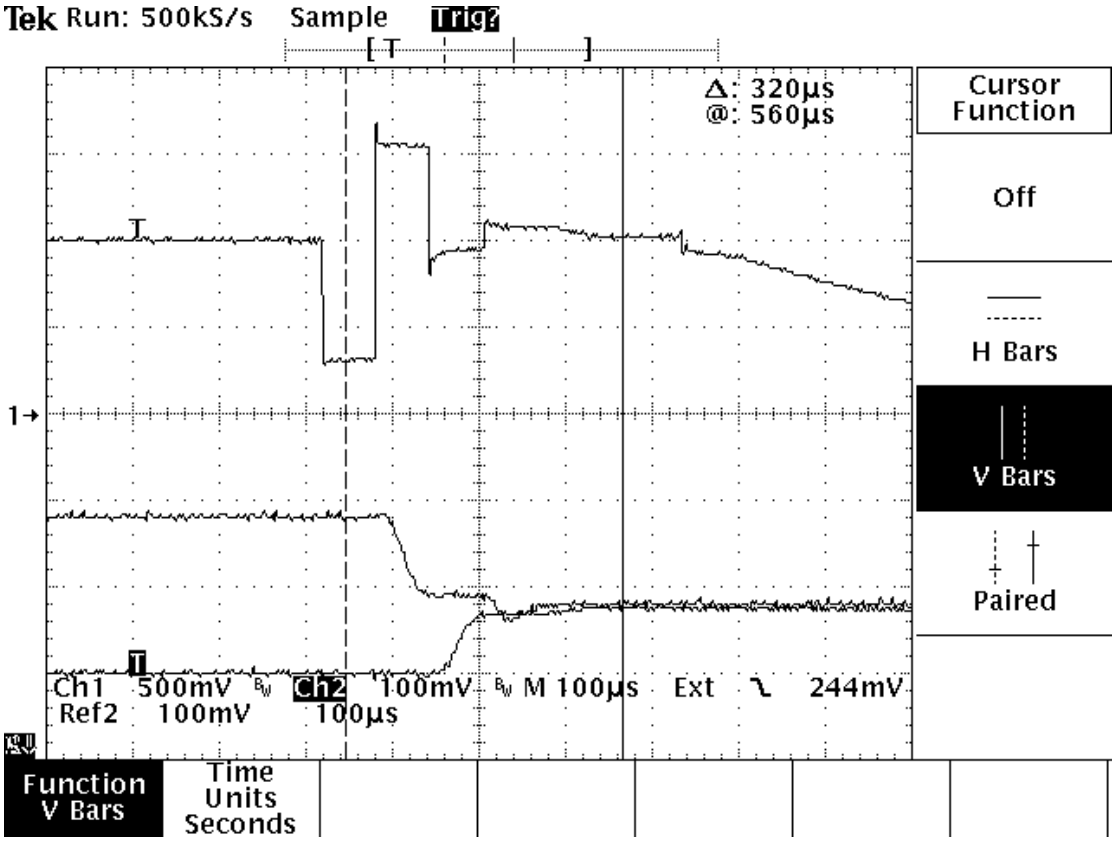


Figure 23. Correction of I and Q with forced offsets of I = 200 and Q = -200

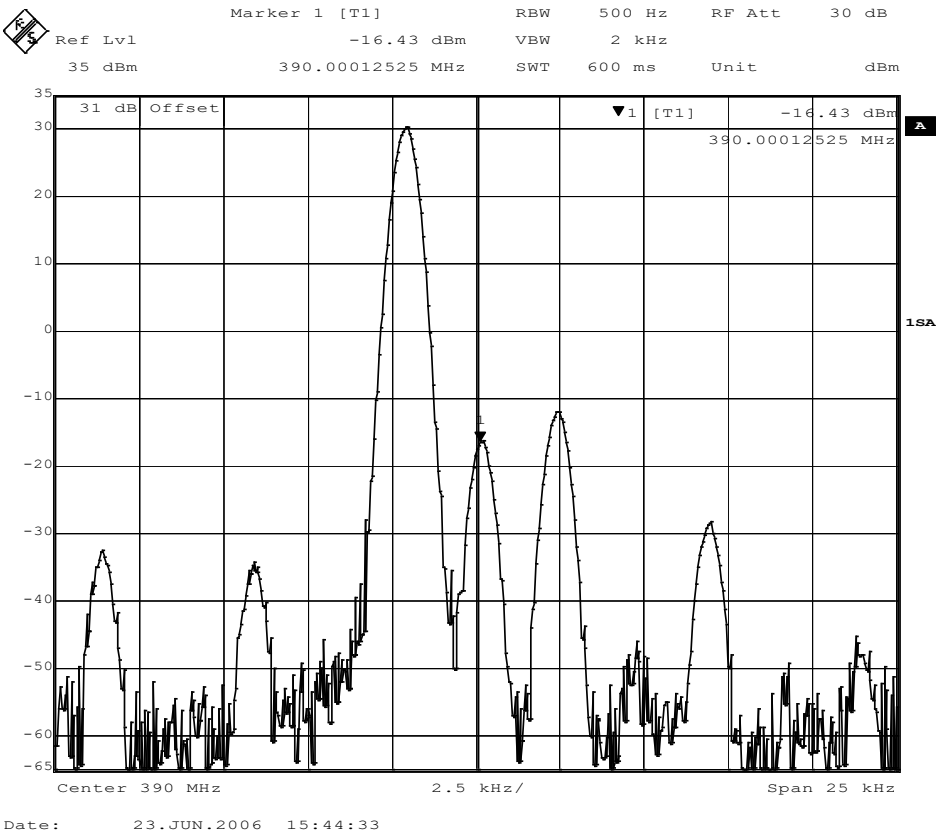


Figure 24. Nulled Carrier with forced offsets of I = 200 and Q = -200

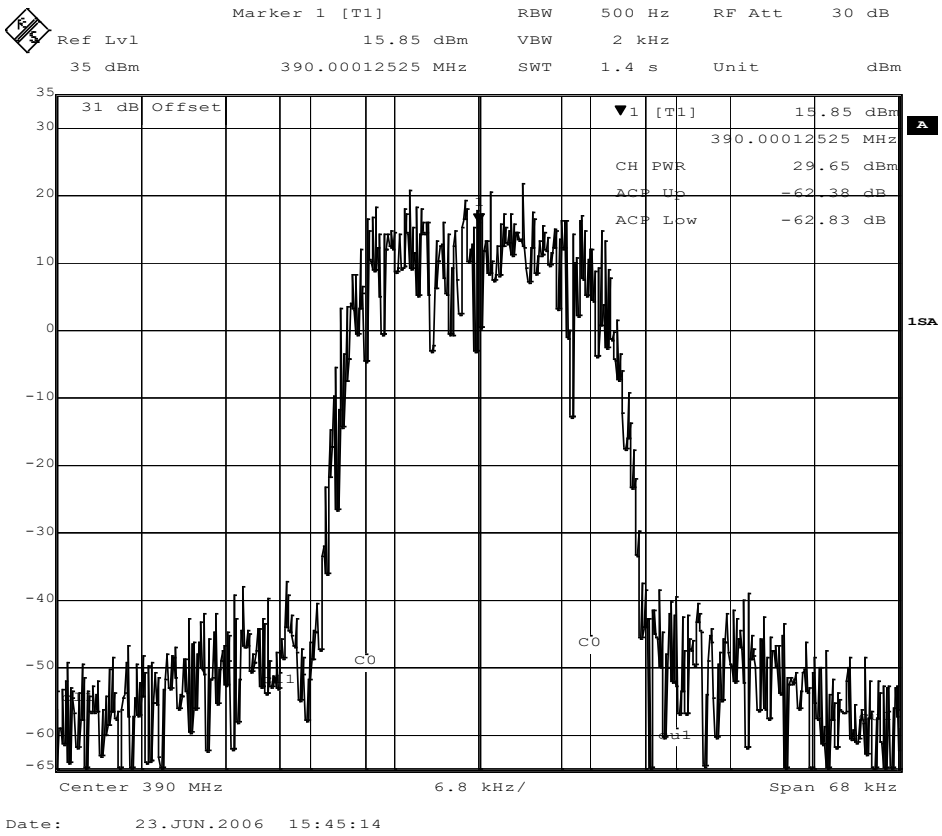


Figure 25. TETRA ACP Performance with forced offsets of I = 200 and Q = -200

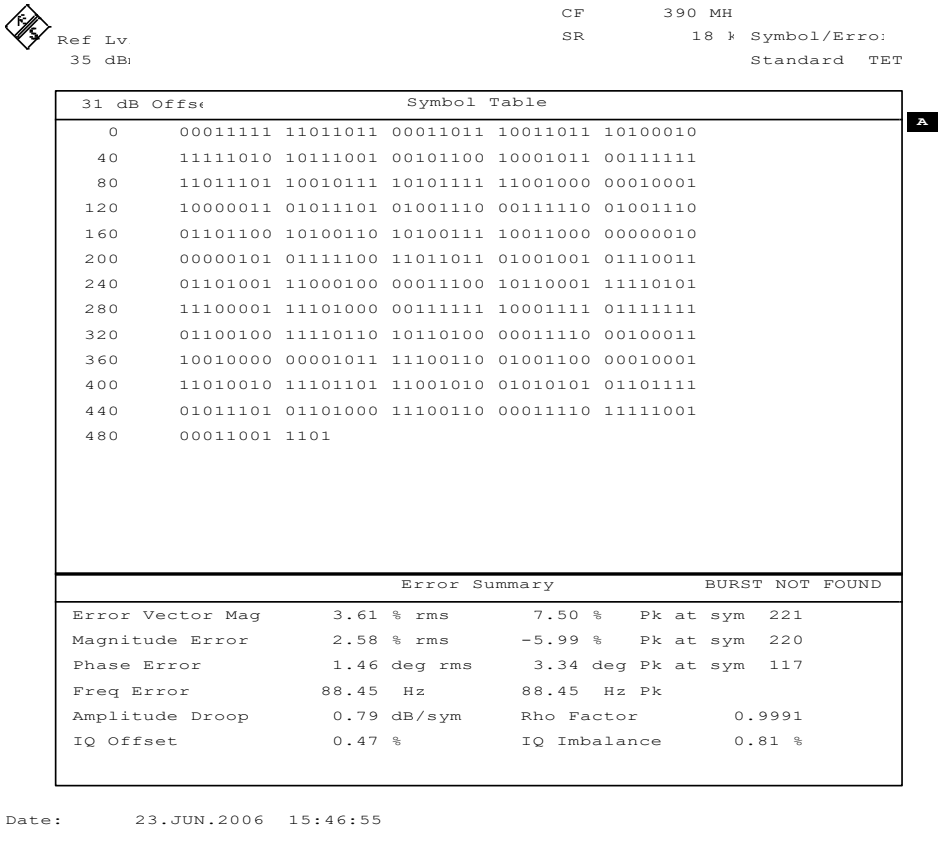


Figure 26. Symbol/Error table with forced offsets of I = 200 and Q = -200

6.3 Enabling the PA

The PA can be enabled after the calibration cycle so that it is ready to start transmission. The Error Amplifier gain should be returned to the high gain state before the PA is enabled. A full sequence listing can be found in Table 2.

Figure 27 shows the output of the Error Amplifier (trace 2) and the PA control line – AUXDAC1 output (trace 1). The time between events in the plot is artificially long so that the sequence can easily be observed. In normal operation, no delay is necessary between the completion of the calibration and the PA enable or between PA enable[9] and the start of modulation. Considering now the detail of Figure 27, the Error Amplifier output is initially forced to a significant offset. The calibration cycle can be seen to correct this error between 2.5 and 3.5 horizontal divisions. After this is complete, the Error Amplifier is switched to full gain and immediately power ramping is started (around 4 horizontal divisions). It can be seen that at this point a glitch in the Error Amplifier output occurs. This is because the output drifts, due to the high gain, before the PA has turned on and feedback applied. Once the PA is enabled, the Error Amplifier output sits at the corrected DC level until modulation starts at approximately 6.5 divisions.

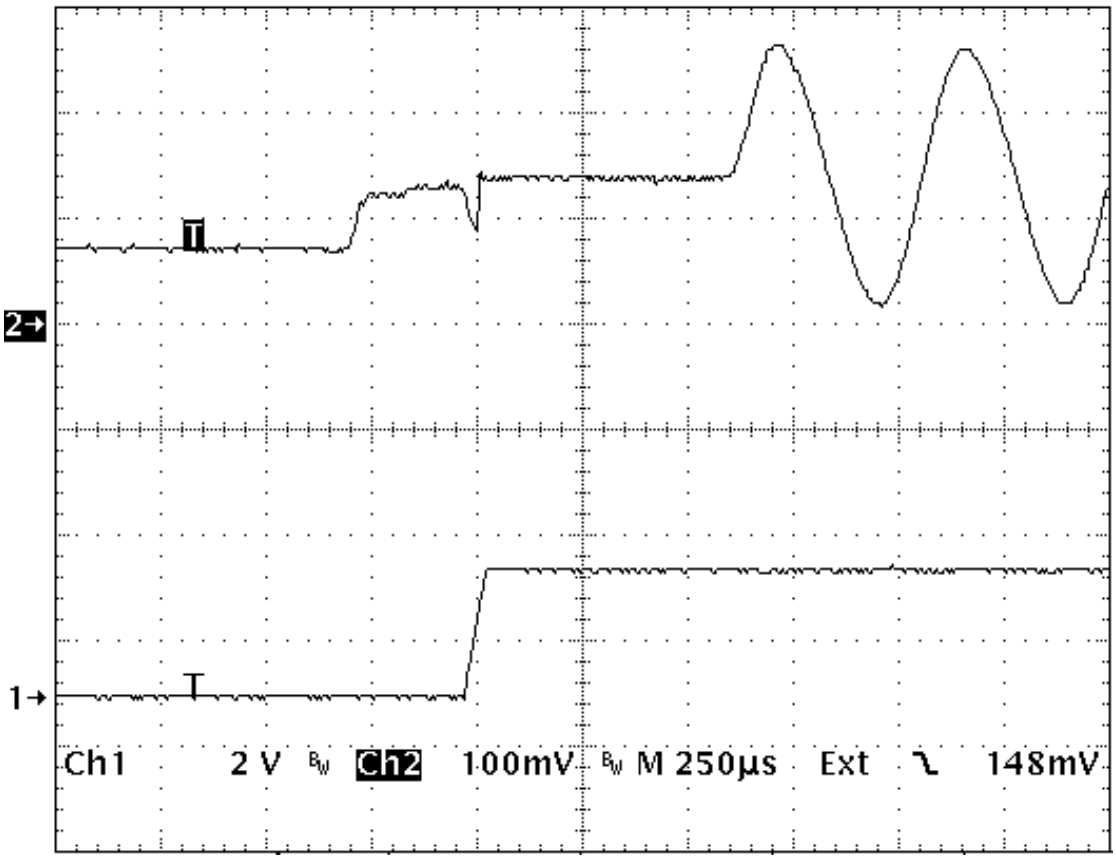


Figure 27. PA turn-on waveforms

[9] In this test system the PA is turned on quickly, in many situations the ramping facility of the CMX981 will be used to control the turn on of the PA over a period of several hundred microseconds to minimise switching transients generated by the transmitter.

6.4 Calibration and Hold Timing

The timing of the calibration routine is an important consideration because typical air interface standards allow only a limited time for calibration, typically less than 500us. The methods described easily meet this requirement by providing calibration times of between 300us and 400us. These calibration times are the sum of the time taken for the sampling and calculation process plus the hardware delay in applying the correction. The results are summarised in Table 8 for the test software and agree closely with the measured data. These times could be reduced to a total of 244us by the changes to the code discussed in Section 5.2.

Clock Tick	Period / us 6.9444	Operation	Hardware Delay / us	Action Complete / us
0	0.00	Set DCMEAS=Idc		
1	6.94			
2	13.89			
3	20.83			
4	27.78			
5	34.72			
6	41.67	Read ADC= Idc,		
7	48.61			
8	55.56	Apply Correction to I		
9	62.50	Set DCMEAS=Qdc	70	132.50
10	69.44			
11	76.39			
12	83.33			
13	90.28			
14	97.22			
15	104.17	Read ADC= Qdc		
16	111.11			
17	118.06	Apply Correction to Q	70	188.06
18	125.00	Set DCMEAS=Idc		
19	131.94			
20	138.89			
21	145.83			
22	152.78			
23	159.72			
24	166.67	Read ADC= Idc		
25	173.61			
26	180.56	Apply Correction to I		
27	187.50	Set DCMEAS=Qdc	50	237.50
28	194.44			
29	201.39			
30	208.33			
31	215.28			
32	222.22			
33	229.17	Read ADC= Qdc,		
34	236.11			
35	243.06	Apply Correction to Q	50	293.06

Table 8. Timing of DC Calibration for two passes.

An important feature of a DC calibration scheme is the hold time, that is the time after the completion of the calibration for which the correction remains valid. In TDMA systems (e.g. TETRA), the required hold time can be quite short (14ms in the case of TETRA). In other systems, typically FDMA systems or systems using analogue PTT based operation, a much longer hold time is required. The system described here has excellent hold time characteristics due largely to the digital implementation of the correction. The accuracy of the calibration is purely based in the change of analogue offset voltages, principally in the CMX998. As the CMX998 is very good in this respect, hold times are very long. In typical tests the system calibration levels, once established, remain valid for many minutes. As would be expected, calibrations of circa –50dBc tend to drift faster than ones at circa –30dBc as the relative impact of a few mV drift is much more severe. A reasonable calibration may often be maintained for hours in bench test conditions.

7 Conclusions

It has been shown with the use of the DCMEAS output on the CMX998 that DC offsets can be measured and a correction calculated and applied at the I and Q inputs resulting in satisfactory reductions in carrier leakage. It has also been shown that the CMX981 can be used to sample the DCMEAS signal, control the PA and apply the DC correction at the I and Q inputs. A calibration of DC offsets up to and in excess of 100mV on I and Q can be corrected successfully within 300µS.

An optimal calibration can be achieved using a correction factor but in the majority of the previous test scenarios the amount of calibration accuracy is in excess of that required for typical applications. For example, a carrier level of around –20dBc meets the requirements of the TETRA standard[4]. Manufacturers may wish to exceed this by some margin and a typical requirement is around –30dBc (or better). This level of performance is achieved by the basic calibration scheme (Section 6.2) without the need for 'correction factors'.

If increased precision is required and it is felt desirable to avoid 'correction factors', perform an initial calibration with the Error Amplifier in low gain mode, then switch to high gain mode and perform a further calibration cycle. Section 6.2 shows that results of better than –50dBc can be achieved.

To summarise, the DC calibration scheme described in this document:

- Comfortably exceeds the accuracy requirement of product standards such as TETRA
- Meets typical timing requirements for performing DC calibration (less than 500µS)
- Has an excellent repeatability and hold characteristic
- Uses a straightforward software implementation
- Has seamless hardware interface requirements when using the CMX981 for the ADC and DAC functions

8 References

1. CMX998 Cartesian Feed-back Loop Transmitter D/998/4 August 2006
2. CMX981 Advanced Digital Radio Baseband Processor D/981/5 January 2005
3. TMS320C5510 DSP Starter Kit (DSK)
4. ETSI EN 300 392-2 Terrestrial Trunked Radio (TETRA) voice and data. Part 2: Air Interface V2.4.2 (2004-02)
5. EV9980 Evaluation Kit User Manual UM9980/2 August 2006
6. EV9810 Evaluation Kit User Manual UM/9810/2 June 2005

9 Code Listing

```
//////////////////////////////////////////////////
// This code is intended to run on a Texas Instruments C5510 DSK.

// Initialisation requires setting up the DSK,
// the C5510 McBSPs and then we are able to talk to
// the CML devices.
// Initialise them and setup an ISR, which does the
// remainder of the work
//////////////////////////////////////////////////
U16 main(void)
{
   init_dsk();                 //initialise hardware environment external to DSP
   init_mcbsp();                 //Setup the serial ports to  talk to 981 and 998
                              //Sets up an interrupt on I,Q pair Rx from 981 called mcbsp0_rx_isr

   init_cmx981();               //Setup the 981 ready to calibrate and transmit
   init_ramdac();               //Initialise a basic ramdac ramp so we can ramp up

   init_cmx998(cmx998_phase,cmx998_gain);  //initialise the 998, load a phase and gain in so as to be calibrated

   while (1);                  //nothing further for the foreground to do, so wait forever
}

/*************************************************************
** Function: init_cmx998
** Initialises the CMX998.
*************************************************************/
void init_cmx998(U16 phase,U16 gain)
{
   set_cpld_misc_bits(0x0002);//MCbsp output onto header, not host

   cmx998_write8(CMX998_RESET,0);            //reset the 998

   cmx998_write8(CMX998_COMMAND_WRITE,0xfd);   //0xfc for closed loop //0xfd closed less gain
                                 //0xfe=open loop//0xff=open loop, less gain

   cmx998_write8(CMX998_PHASE_WRITE,phase);    //set the phase that was passed - to calibrate
   cmx998_write16(CMX998_GAIN_WRITE,gain);     //set the gain that was passed - to calibrate
   cmx998_write8(CMX998_AUX_WRITE,0x80);
}

/*************************************************************
** Function: init_cmx981
** Initialises the CMX981 , having firstly reset it.
** both transmit and receive are enabled, receive so that
** we can get an interrupt timing reference.  Also config some of the
** aux hardware to do calibration
*************************************************************/
void init_cmx981(void)
{
   U16 temp;

   asm(" bset INTM");            /*Ensure interrupts are disabled*/

   cmx981_reset();              /* Reset CMX981 */

   temp=*IFR0;                  /*clear any pending interrupts*/
   *IFR0=temp;
   temp=*IFR1;                  /*clear any pending interrupts*/
   *IFR1=temp;

   *IER0|=BRINT0_IMASK;          /*unmask McBSP0 Rx int only*/

   cmx981_cmd_write(CONFIGCTRL1 , 0x80);   /*Set Config1 SCLK=MCLK*/
   cmx981_cmd_write(CONFIGCTRL2 , 0x00);   /*Set Config2 */

   cmx981_cmd_write(TXIGAINLSB , 0x00);    /*Set 8 LSBs of TxI channel Gain*/
   cmx981_cmd_write(TXIGAINMSB , 0x00);    /*Set 4 MSBs of TxI channel Gain*/
   cmx981_cmd_write(TXQGAINLSB , 0x00);    /*Set 8 LSBs of TxQ channel Gain*/
   cmx981_cmd_write(TXQGAINMSB , 0x00);    /*Set 4 MSBs of TxQ channel Gain*/
```

```
    cmx981_cmd_write(TXRAMPUPINCMSB , 0xff);/*ramp up quickly*/
    cmx981_cmd_write(TXRAMPDNDECMSB , 0xff);/*ramp down quickly*/

    cmx981_cmd_write(RXIGAINLSB , 0xff);    /*set 8 lsbs of RxI channel Gain*/
    cmx981_cmd_write(RXIGAINMSB , 0x7f);    /*set 8 msbs of RxI channel Gain*/
    cmx981_cmd_write(RXQGAINLSB , 0xff);    /*set 8 lsbs of RxQ channel Gain*/
    cmx981_cmd_write(RXQGAINMSB , 0x7f);    /*set 8 msbs of RxQ channel Gain*/

    cmx981_cmd_write(CLKSTOPCTRL , 0x00);   /*enable FIFO - EDS rev 2 has bit inverted!*/
    cmx981_cmd_write(POWERDOWNCTRL , 0x1f); /*power up aux DACs and analogue bias chain. Digital outputs slow*/

    cmx981_cmd_write(LOOPBACKCTRL , 0x00);  /*no loopback*/

    cmx981_cmd_write(AUXADCCTRL1 , 0x1);   /*enable aux ADC 1 only */
    cmx981_cmd_write(AUXADCCTRL2 , 0x00);   /*no continuous conversion for aux ADCs */

    cmx981_cmd_write(CODECSETUP1 , SPEAKER|MIC1);   /*20dB input gain & 6dB output gain enabled by default*/
    cmx981_cmd_write(CODECSETUP2 , 0x09);   /*Enable voice codec, enable output*/
    cmx981_cmd_write(CODECGAIN1  , 0xc5);   /*+18dB input gain -10dB output gain*/

    cmx981_cmd_write(RXSETUP1 , 0x88);      /*Enable the receiver in 32 bit mode*/
    cmx981_cmd_write(TXSETUP , 0x8A);       /*Enable the transmitter*/

    temp = cmx981_cmd_read(STATUS1);        /*Do a dummy read of Status regs to clear any pending INTs*/
    temp = cmx981_cmd_read(STATUS2);
    temp = cmx981_cmd_read(STATUS3);

    cmx981_cmd_write(AUXADCCTRL2 , 0x04);   /*enable fast conversion for aux ADCs, but prepared to use single shot!*/
                                    //setup user selected, hardware specific gain and phase here
    cmx981_cmd_write(TXQGAINLSB, (U16)(tx_q_gain));
    cmx981_cmd_write(TXQGAINMSB, (U16)(tx_q_gain >>8));
    cmx981_cmd_write(TXIGAINLSB, (U16)(tx_i_gain));
    cmx981_cmd_write(TXIGAINMSB, (U16)(tx_i_gain >>8));
    cmx981_cmd_write((U16)TXIPHASELSB|(tx_i_phase & 0x1ff)>>8, (U16)(tx_i_phase & 0xff));
    cmx981_cmd_write((U16)TXQPHASELSB|(tx_q_phase & 0x1ff)>>8, (U16)(tx_q_phase & 0xff));

    asm(" bclr INTM");                  /* enable interrupts*/
}

/*************************************************************
** Initialise the ramdac - simple linear ramp
*************************************************************/
void init_ramdac(void)
{
    U16 i;

    cmx981_cmd_write(CONFIGCTRL2 , 0x00);   /*Set Config2 to ensure Aux RAM write ptr is reset*/
    cmx981_cmd_write(CONFIGCTRL2 , 0x04);       /*Set Config2 to access Aux RAM (write)*/

    for(i=0;i<32;i++)//2 data words per loop, 64 locations
    {
        cmx981_cmd_write(SRAMDATA,0);
        cmx981_cmd_write(SRAMDATA+1,i<<3);
        cmx981_cmd_write(SRAMDATA+2,0);
        cmx981_cmd_write(SRAMDATA+3,i<<3);
    }
    cmx981_cmd_write(CONFIGCTRL2 , 0x00);       /*Set Config2 back*/
}

/*************************************************************
** ISR: called on I,Q pair received interrupt from the McBSP
** rate expected to be 144kHz
*************************************************************/
interrupt void mcbsp0_rx_isr(void)
{
    static U16 measurement=0;       //counter for the measurement we are making. 0,1=Qref,Iref 2,3=Imeas,Qmeas
    static S16 cal_counter=-128;    //count for calibration step sequencing. -ve=not calibrating, previous tx ending
    static S16 tx_counter=0;        //count for tx sequencing - modulo 32 for 981 symbol tx
    static U16 cal_passes=6;        //number of passes of 9 steps to carry out to calibrate

                            //dummy read of McBsp so we get another interrupt
    dummy = io_read(DRR2_0);        //put Q data in buffer
    dummy = io_read(DRR1_0);        //put I data in buffer
```

```
   if((tx_counter%32)==0  && cal_counter<0)    //if not calibrating then do some transmitting
   {
      switch(tx_pattern)                    //user selectable pattern
      {                                 //cmx981_tx_control=4 before cal: ramp down,
                                        //set 7 after cal to ramp up and do 4 symbol transmit
         case (ZEROES):
         cmx981_cmd_write(cmx981_tx_control, 0x00);
         break;

         case (PETAL):
         cmx981_tx_write(cmx981_tx_control, 0xff);
         break;

         case(PRBS):
         cmx981_tx_write(cmx981_tx_control, prbs_data[symbol_index]);
         symbol_index++;
         if (symbol_index >= TX_PATTERN_LENGTH) symbol_index = 0;
         break;
      }
   }

   if(key_up!=old_key_up)                //user control code: watch key_up and change it to do another cal
   {                                 //key_up=1 is set by the code when cal is finished
      if(key_up!=1)                      //user set a value !=1 to trigger another cal
      {                                 //so reset the calibration system for another go
                                        //Dont care that this takes ages as it is just a user/demo function
         cmx981_tx_control=0x04;
         cmx998_write8(CMX998_COMMAND_WRITE,0xfd);   //for closed loop small gain during calibration
         cmx981_cmd_write(RAMDACCTRL,0x31);         //cause a ramp down to start
         key_up=0;                              //and switch on the calibration process

         //For test purposes, inject a DC error to undo any previous calibration.  Typically dont need this!!!
/*       i_info=FORCED_I_OFFSET;
         q_info=FORCED_Q_OFFSET;
         cmx981_cmd_write(TXQOFFSETMSB,0xf&(q_info>>8));
         cmx981_cmd_write(TXQOFFSETLSB,q_info&0xff);
         cmx981_cmd_write(TXIOFFSETMSB,0xf&(i_info>>8));
         cmx981_cmd_write(TXIOFFSETLSB,i_info&0xff);*/        }
      old_key_up=key_up;
   }

   if(!key_up)                              //If we are not keyed up then calibrate and key up!
   {
      if(cal_counter==0)                    //provide a useful scope trigger so we can see the waveforms
         set_cpld_user_bits(0x0008);
      if(cal_counter>=0)                    //dont calibrate while the transmit signal from the last burst is still present
      {
         switch(cal_counter%9)                 //9 stages to each pass
         {
            case 0:                         //to read back a DCMEAS. value from the 998 we need to:
               cmx998_write8(CMX998_AUX_WRITE,0x80+measurement);   //select our 998 measurement to make
               cmx981_cmd_write(AUXADCCTRL2 , 0x01);              //fire off an aux adc single shot
               break;
            case 6:                         //wait several stages and then read back the result
               monitor[measurement]=(0xff&cmx981_cmd_read(AUXADC1DATAMSB))<< 2;
               break;
            case 7:                         //2 goes to read the result
               monitor[measurement]|=(0x3&cmx981_cmd_read(AUXADC1DATALSB));
               break;
            case 8:                         //compute and apply a correction (which takes time to settle)
               if(measurement==2)
               {
                  compute_offset(&i_info,monitor[1],monitor[2]);
                  cmx981_cmd_write(TXIOFFSETMSB,0xf&(i_info>>8));
                  cmx981_cmd_write(TXIOFFSETLSB,i_info&0xff);
               }
               if(measurement==3)
               {
                  compute_offset(&q_info,monitor[0],monitor[3]);
                  cmx981_cmd_write(TXQOFFSETMSB,0xf&(q_info>>8));
                  cmx981_cmd_write(TXQOFFSETLSB,q_info&0xff);
               }
               //go to the next measurement of Qref,Iref,Imeas,Qmeas,Imeas,Qmeas....
               //for as long as we choose to run for
               measurement++;//0,1,2,3,2,3,2,3....
```

```
            if(measurement==4)
               measurement=2;
            break;
         };
      }
      if(cal_counter==(cal_passes*9)) //End of calibration after 6*9 step stages. 2 for reference measurement
                              //which is only needed the first pass and 4 for calibration
      {
         key_up=1;                        //key_up signals transmit mode, not calibration
         cmx981_tx_control=0x7;                  //transmit, ramped up in 4 symbol mode
         cmx998_write8(CMX998_AUX_WRITE,0x0);        //no more measurement required
         cmx998_write8(CMX998_COMMAND_WRITE,0xfc);   //0xfc for closed loop big gain
         cmx981_cmd_write(RAMDACCTRL,0x35);         //trigger a ramp up
         tx_counter=-1;                     //setup for the transmission to start
                                 //For next time:
         cal_counter=-128;                     //next time we will allow some time(count -128 to 0)
                                 //before calibrating so that signal have had time to go
         measurement=2;                        //begin on this measurement next time.
                                 //0=read references, 2 cal only
         cal_passes=4;                     //and change so that we dont do it for as long!
         clear_cpld_user_bits(0x0008);             //clear down a useful scope trigger for looking at cal signals
      }
      cal_counter++;
   }
   tx_counter++;
}

/*************************************************************
** given a pointer to a current correction, modify it based
** on the reference and measured value provided
*************************************************************/
#define ERROR_GAIN  0.95              //Gain applied to computed error before correction
#define ERROR_OFFSET 6                //Offset between error measured in low gain vs high gain modes
void compute_offset(S16 *offset,S16 reference,S16 measured)
{
   float error;

   error=reference-measured+ERROR_OFFSET;             //compute error between the reference and the measured signal
   *offset-=(S16)(error*ERROR_GAIN);                 //compute a correction from that and apply it (with a degree of damping)
}

//EOF
```

www.cmlmicro.com

For FAQs see: http://www.cmlmicro.com/products/faqs/index.htm

For a full data sheet listing see: http://www.cmlmicro.com/products/datasheets/download.htm

For detailed application notes: http://www.cmlmicro.com/products/applications/index.htm

CML Microcircuits (UK) Ltd COMMUNICATION SEMICONDUCTORS	CML Microcircuits (USA) Inc. COMMUNICATION SEMICONDUCTORS	CML Microcircuits (Singapore) Pte Ltd COMMUNICATION SEMICONDUCTORS	
Tel: +44 (0)1621 875500 Fax: +44 (0)1621 875600 Sales: sales@cmlmicro.com Technical Support: techsupport@cmlmicro.com	Tel: +1 336 744 5050, 800 638 5577 Fax: +1 336 744 5054 Sales: us.sales@cmlmicro.com Technical Support: us.techsupport@cmlmicro.com	Tel: +65 7450426 Fax: +65 7452917 Sales: sg.sales@cmlmicro.com Technical Support: sg.techsupport@cmlmicro.com	Tel: +86 21 63174107 +86 21 63178916 Fax: +86 21 63170243 Sales: cn.sales@cmlmicro.com.cn Technical Support: sg.techsupport@cmlmicro.com