International
IOR Rectifier

# DESIGN TIP

DT 99-8

International Rectifier • 233 Kansas Street, El Segundo, CA 90245 USA

## Using IR2171/IR2172 within DSP-controlled AC Drives

**By Dorin O. Neacsu**

### *TOPICS COVERED*

*Description of the control system including the linear current sensing ICs*
*Using Texas Instruments TMS320C240 DSP*
*Hardware description*
*Software description*
*Open-loop results*

### *1. DESCRIPTION OF THE CONTROL SYSTEM INCLUDING THE LINEAR CURRENT SENSING ICs*

Many applications require a three-phase AC drive ranging from hundreds of W to few MW. All of these cases are based on a three-phase inverter controlled by a microcontroller or DSP system. Measurement of at least two phase currents is desired in more advanced systems that are employing Field-Orientation Control of induction or synchronous machines.

There are several methods generally used for measurement of inverter currents:

- shunt resistors on the phase currents followed by linear optocouplers;
- Hall effect current sensors used on the phase currents;
- shunt resistors on the DC link current followed by an OpAmp and extensive reconstruction hardware/software.

IR2171/IR2172 are a promising alternative to these methods and they are based on sensing the phase currents through an external shunt resistor and transfer of the signal to the low side after converting it from analog to digital. The level shift function is realized using IR's high-voltage isolation technology. The output signal is a carrier pulse-width modulated wave with *fc* around 40kHz that can interface to a microcontroller system directly or through a galvanical separation. The duty ratio varies according to the shunt resistor voltage up to ±260mV.

This new approach has inherent advantages over the previous solutions in reduced cost and size as well as the possiblity of using the microcontroller system on the same ground level as the power system (especially in low power applications). Design note introduces the IR2171/IR2172 by explaining their use

as current measurement system within a micro-controlled AC drive. The *Texas Instruments TMS320F(C)240* has been chosen as an example due to its simple interface. Both system interface and software are explained and performance is analyzed based on some results that are finally detailed.

The PWM signal is applied to a counter input and the pulse widths are transformed in digital code by counting at the microcontroller clock frequency. The temperature drift of the output carrier voltage can be cancelled by measuring both intervals corresponding to the low and high level. A software routine is expected to calculate the duty ratio and to transform this data into an equivalent current value.

## *2. Using TMS320C(F)240 DSP*

*Texas Instruments TMS320C(F)240DSP* is a good performing computing engine with rich peripherals designed for AC motor drive applications. The Field-Orientation control structure based on this DSP contains

- the Event Manager for the generation of the Space Vector PWM signals;
- 2 capture channels for the encoder.

The other two channels of the capture unit are herein used to the phase current measurement.

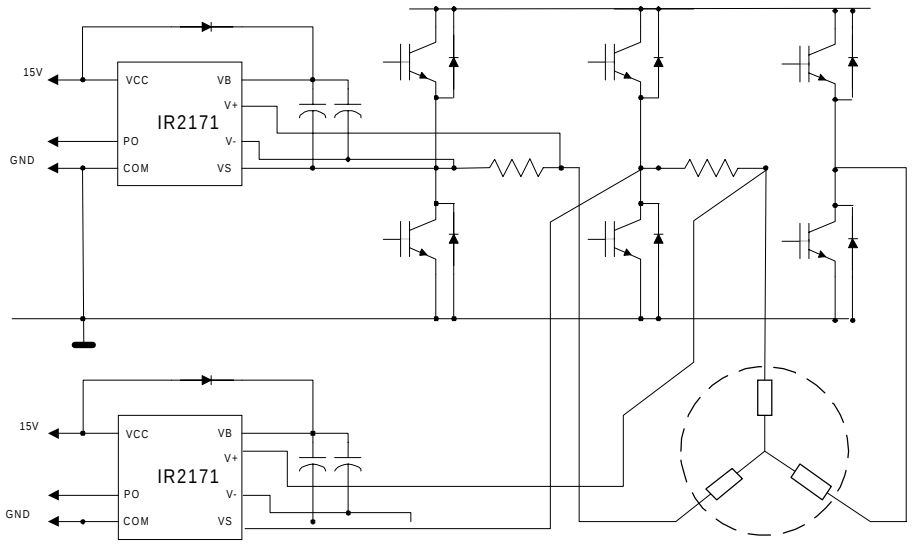



**Figure 1 - IR2171 current measurement in a three-phase inverter drive**

In this respect, a timer is selected to count at the CPU clock frequency of 20 or 30MHz and the counter content is saved into a stack at transitions on the capture input pins. The latency from a transition happens on an input pin and the moment the counter value is locked is 3.5-4.5 CPU clock cycles.

Since it is likely to measure both active and inactive state intervals, the timer content should be saved at both rising and falling edges. Unfortunately, the DSP system does not allow to separate the capture event from the interrupt generation event. In other words, we cannot capture both transitions

and ask the capture unit to generate an interrupt for reading data at each second transition. When employing interrupts on both edges, there is not enough time to serve those interrupts and to perform the main Field-Orientation control program. There would be 2 interrupts at each *1/fc* for each current measurement, each interrupt service routine lasting for at least 2µs. The DSP utilization factor would therefore be seriously affected.

Another possible problem is related to the minimum pulse width that is basically 2 µs, but can reach its minimum of τmin=1.87µs due to technological dispersion.

The delay between the interrupt request and its service is composed of:

- the synchronization time it takes for the interrupt to be recognized by the peripheral interface and converted into a request to the DSP. It takes:
  - **2 clock cycles** for internal interrupts
  - **4 clock cycles** for external interrupts.
- the CPU response time to recognize and acknowledge the enabled interrupt request. The minimum latency is **4 clock cycles** and longer for multicycle operations (memory access using wait states, repeat loops, pipeline operation, or serving other interrupt before the interrupts are unmasked again).
- the time needed by ISR to branch to get the specific event that produced the interrupt. possible case leads to minimum **4 clock cycles**.
- the time needed by the software routine designed to serve the interrupt to save the status registers, accumulator and sometimes other registers before doing the expected stack readings, savings of data to memory and other calculations. Minimum time expected for this interrupt routine initialization is **6 clock cycles**.

The overall time necessary to save the capture unit data to memory is very close to the value of the minimum pulse width of the input PWM signal.

Noticing these allows us to propose a solution based on having only one interrupt for each period (around 40kHz) at each falling edge of the capture input signal. During the appropriate interrupt servicing routine, the last two values of the capture stack are read and minor calculations are employed to defining the pulse widths. Such interrupt can be requested by a separate input of the DSP and acting as an external interrupt. The interrupt latency is about the same in this approach when compared to the solution of having two interrupts at each sampling interval, interrupts caused by the capture unit itself. However, the DSP utilization factor is improved allowing further development such as Field-Oriented Control or sensorless control.

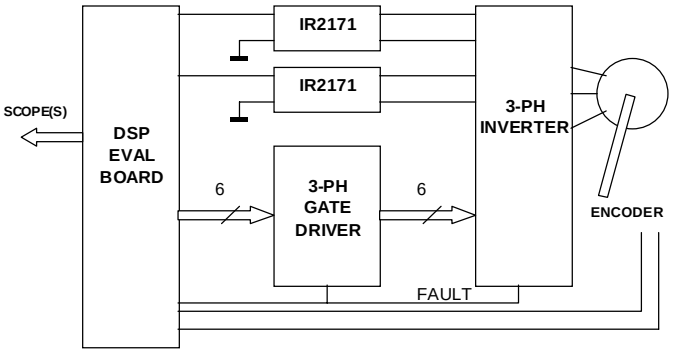



**Figure 2 - Block diagram of the control system**

## *3. Hardware description*

The interface of the two IR2171 devices with the DSP is shown in Fig.2. The TMS320C24x DSP controllers evaluation module has been used for testing the current acquisition. *Table I* provides some information about the interface with IR2171, but the power supply, clock and usual circuitry is not included.

| Name of the signal | | DSP TMS320C240 pin | Eval.board connector |
|---|---|---|---|
| PWM outputs to be connected to the inverter gate drivers | PWM1/CMP1 | 94 | P1-pin3 |
| | PWM2/CMP2 | 95 | P1-pin4 |
| | PWM3/CMP3 | 96 | P1-pin5 |
| | PWM4/CMP4 | 97 | P1-pin6 |
| | PWM5/CMP5 | 98 | P1-pin7 |
| | PWM6/CMP6 | 99 | P1-pin8 |
| | | | |
| Fault signal | PDPINT | 52 | P1-pin26 |
| | | | |
| Encoder signals | CAP1/QEP1 | 67 | P1-pin21 |
| | CAP2/QEP2 | 68 | P1-pin22 |
| | | | |
| 2171-ph.A | CAP3/QEP3 | 69 | P1-pin23 |
| 2171-ph.B | CAP4/QEP4 | 70 | P1-pin24 |
| | | | |
| 2171-ph.A | XINT2 | 54 | P4-pin19 |
| 2171-ph.B | XINT3 | 55 | P4-pin20 |
| | | | |
| Scope-ch.1 | DACOUT0 | - | P2-pin25 |
| Scope-ch.2 | DACOUT1 | - | P2-pin26 |
| Scope-ch.3 | DACOUT2 | - | P2-pin27 |
| Scope-ch.4 | DACOUT3 | - | P2-pin28 |

**Table I**

## *4. Software description*

This section describes the software routines that could be included in any control software for a 3-phase motor drive in order to make use of the IR2171 for current measurements.

This solution is not unique and it is presented herein as a test example. Moreover, only those parts of code that refer to the capture unit configuration and current calculations are included herein.

```
;********* initialization of variables ————————————————-
……
;******** Vector address declarations   ********************************
             .sect   ".vectors"
RSVECT        B      START   ; PM 0    Reset Vector          1 ;00
INT1          B      CRT_ISR  ; PM 2    Int level 1           4 ;02
INT2          B      PWM_ISR  ; PM 4    Int level 2           5 ;04
INT3          B      PHANTOM  ; PM 6    Int level 3           6 ;06
…..
       .text
```

```
;********* M A I N   C O D E  - starts here    ************************
START: ……..
;*********   SYSTEM CONFIGURATION         *********************************
…..
              LDP    #04h          ;
              MAR    *,1           ;
              LAR    AR0,#initar   ;
……..
;********* CONFIGURE EVENT MANAGER   ——————————————-
; Configure I/O pins as capture units ——————————————-
              LDP    #0E1h         ;
              SPLK   #0FF00H,OPCRA ;
              SPLK   #00F0H,OPCRB  ;
;********* INITIALIZE EVENT MANAGER   ——————————————-
              ……..
;——— Configure CAPCON and enable capture operation ———————-
              SPLK #1001101000001111b,CAPCON    ;
;——— External digital inputs used for interrupting at current measurement
              LDP    #0E0h                ;
              SPLK #1000000000000001b,XINT2_CNTL;
              SPLK #1000000000000001b,XINT3_CNTL;
……..
;********* Main background loop   ************************************
MAIN          LDP    #04h          ;
              NOP                  ;
              NOP
              B      MAIN          ;
;********* Routine Name: P W M _ I S R   *************************
PWM_ISR:      CLRC   INTM          ;
              SST    #0,stk1               ; save ST0 - Forced Page 0
              SST    #1,stk2               ; save ST1 - Forced Page 0
              LDP    #00h          ;
              SACL   stk3   ; save ACCL
              SACH   stk4   ; save ACCH
              LDP    #0E8h         ;
              SPLK   #0FFFFh,IFRA  ; Clear all Group A interrupt flags
;******** Phase Current calculation - Phase A = IaDelta2/[IaDelta1+IaDelta2]
              LDP    #04h          ;
              LACC   IaDelta2      ;
              ADD    IaDelta1      ;
              SACL   GPR2          ;
              ZALR   IaDelta2      ;
              RPT    #15           ; Place 16-bit pos. divisor in data memo
              SUBC   GPR2          ; Execute SUBC for 16 times
                                   ; L-ACC=quotient and H-ACC=remainder
              SACL   Ia            ; Save ‘Ia’ in D0 format (norm.)
;******** Phase Current calculation - Phase B = IbDelta1/[IbDelta1+IbDelta2]
              LACC   IbDelta2      ;
              ADD    IbDelta1      ;
              SACL   GPR2          ;
              ZALR   IbDelta2      ;
              RPT    #15           ; Place 16-bit pos. divisor in data memo
              SUBC   GPR2          ; Execute SUBC for 16 times
                                   ; L-ACC=quotient and H-ACC=remainder
              SACL   Ib            ; Save ‘Ib’ in D0 format (norm.)
……..
```

```
;******** configure DAC for scope *********************************
              LDP    #04h          ; Set data page pointer to 0000h
              LACC   Ia,12         ;
              SACH   DAC0_VAL      ; Load Ta to channel 1 of DAC
              LACC   Ib,12         ;
              SACH   DAC1_VAL      ; Load Tb to channel 2 of DAC
              LACL   IaTime1       ;
              SACL   DAC2_VAL      ; Load Tc to channel 3 of DAC
              LACL   IaTime2       ;
              SACL   DAC3_VAL      ; Load Tc to channel 3 of DAC
              OUT    DAC0_VAL,0000h       ; Write to the DAC0 register
              OUT    DAC1_VAL,0001h       ; Write to the DAC1 register
              OUT    DAC2_VAL,0002h       ; Write to the DAC2 register
              OUT    DAC3_VAL,0003h       ; Write to the DAC3 register
              OUT    DAC3_VAL,0004h       ; Start DAC conversions by writing DAC
update reg.
;********* Restore regs   ———————————————————
              LDP    #00h          ;
              ZALH   stk4          ; restore ACCH
              ADDS   stk3          ; restore ACCL
              LST    #1, stk2      ; restore ST1
              LST    #0, stk1      ; restore ST0
              RET                  ;
;********* Routine Name: C R T _ I S R   **********************
CRT_ISR       SST    #0,*-         ;
              SST    #1,*-         ;
              LDP    #0E0h         ;
              BIT    SYSIVR,13     ;
              BCND   Ib_ISR,TC     ; If TC=1, run XINT3_S_ISR (Ib)
Ia_ISR        LDP    #0E8h         ; Set data page pointer to 7400h
              LAR    AR6,FIFO3     ; Read the old stack data to AR6
              LAR    AR7,FIFO3     ; Read the newer stack data to AR7
              SACL   *-            ;
              SACH   *-            ;
              CLRC   INTM          ;
              LDP    #04h          ;
              SAR    AR6,IaTime1   ; Save first reading to ‘IaTime1’
              LACC   IaTime1       ;
              SUB    IaTime2       ;
              SACL   IaDelta1      ; Calculate the ‘off’ time.
              SAR    AR7,IaTime2   ; Save second reading to ‘IaTime2’
              LACC   IaTime2       ;
              SUB    IaTime1       ; Calculate the ‘on’ time
              SACL   IaDelta2      ;
              B      SAVEREG       ;
Ib_ISR        LDP    #0E8h         ; Set data page pointer to 7400h
              LAR    AR4,FIFO4     ; Read the old stack data to AR6
              LAR    AR5,FIFO4     ; Read the newer stack data to AR7
              SACL   *-            ;
              SACH   *-            ;
              CLRC   INTM          ;
              LDP    #04h          ;
              SAR    AR4,IbTime1   ; Save first reading to ‘IbTime1’
              LACC   IbTime1       ;
              SUB    IbTime2       ;
              SACL   IbDelta1      ; Calculate the ‘off’ time
              SAR    AR5,IbTime2   ; Save second reading to ‘IbTime2’
              LACC   IbTime2       ;
              SUB    IbTime1       ;
              SACL   IbDelta2      ; Calculate the ‘on’ time
```

```
SAVEREG     SETC   INTM          ;
            MAR    *+,AR0        ;
            LACC   *+,16         ;
            ADDS   *+            ;
            LST    #1,*+         ;
            LST    #0,*          ;
            CLRC   INTM          ;
            RET                  ;
```

## *5. Results*

Several tests have been made in order to validate both hardware and software. In the beginning, open-loop tests have been carried out by applying pure sinusoidal input signals to the IR2171/2172 development board. Most of the waveforms are obtained by reconstruction through 12-bit D/A converters. The data is sent to this converter when currents are calculated by DSP from input data during the interrupt routine for inverter switching PWM generation that occurs at each 4kHz. Effects of sampling could be noticed in most waveforms. However, they are inherent in a digital system and provide information about the real data used by the control loop. The worst case shows a delay of 38µs that corresponds to 13.68º for 1kHz input signal. This delay is measured from the original sinusoidal signal to the left-end point of each S/H level and it contains not only IR2171 phase delay (about 10º phase shift) but also delay due to DSP internal S/H.

### *5-A Waveforms (Fif. 3)*

Trace 3 (up) = input signal
Trace 4 (mid)=2171 to DSP signal (40kHz)
Trace 2 (dwn)= signal reconstructed through the D/A conv.

### *5-B Phase delays (Fig. 4)*

Trace 3 = input signal
Trace A = DSP signal

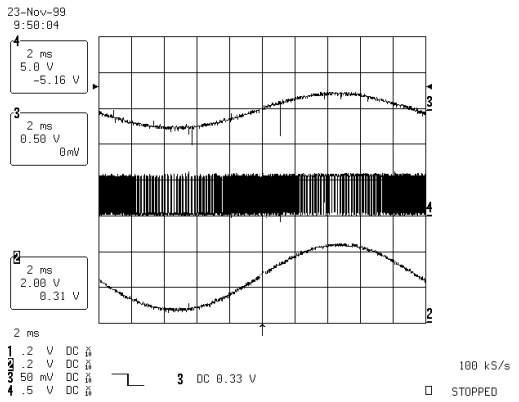



**Figure 3a**
**Input signal: freq.=50Hz, amplitude of 240mV**

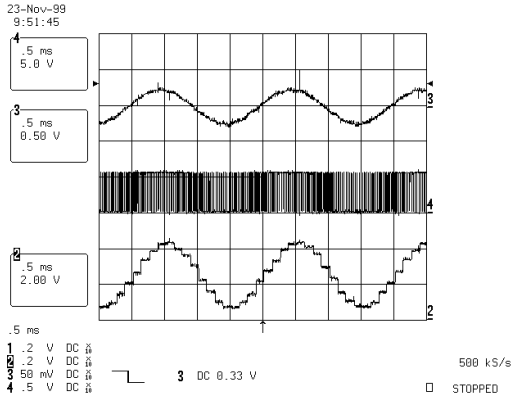



**Figure 3b**
**Input signal: freq.=500Hz, amplitude of 240mV**

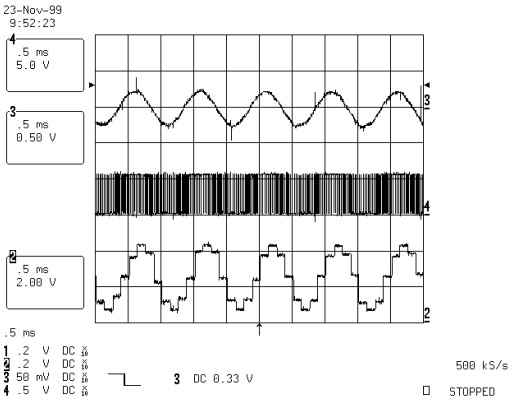



**Figure 3c**
**Input signal: freq.= 1kHz, ampl. of 240mV**

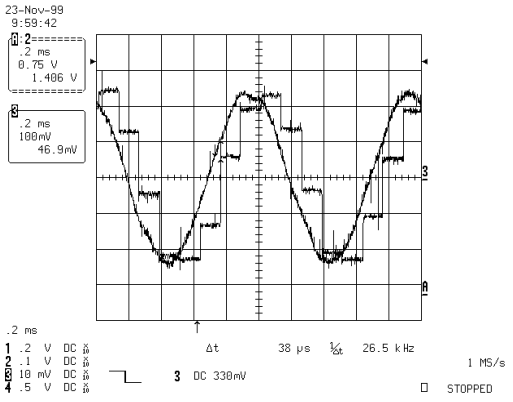



**Figure 4**
**Phase delay measurement for 1kHz signal**

### *5-C Capture operation*

A better understanding of the counter operation and capture units can be achieved by looking at the Fig.5. Both low and large duty ratio cases are presented. One has to keep in mind that the internal counter counts up to *0FFFFh* (16 bits) while the D/A converter allows us to see 12 bits only. As a result, the counter content would look truncated on the scope. This, however, does not affect the meaning of these waveforms. Also, the D/A signals are output during the very fast interrupt routine that occurs during current acquisition (at about 40kHz). Figure 5 presents:

Trace 3 (up) = input signal
Trace 4 (mid) = 2171 to DSP signal (40kHz)
Traces 2,1 (down) = counter at reading time

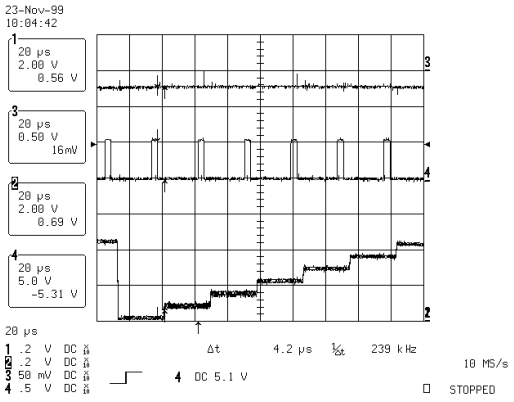



**Figure 5a Low duty ratio**

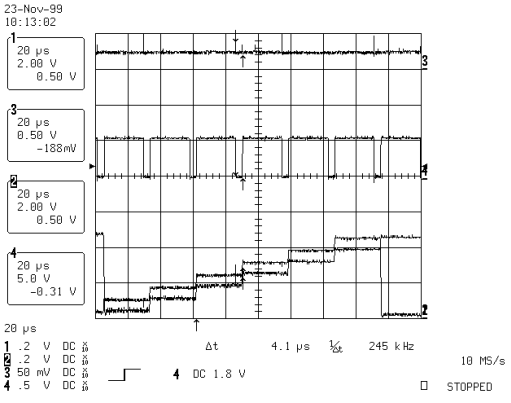



**Figure 5b Large duty ratio**

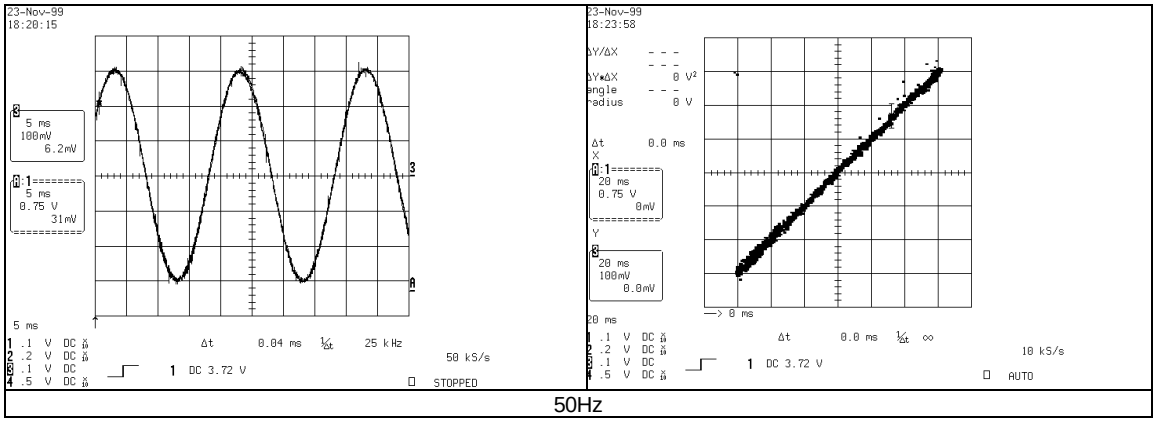
***5-D Transfer characteristics***

Trace 3 (up) = input signal

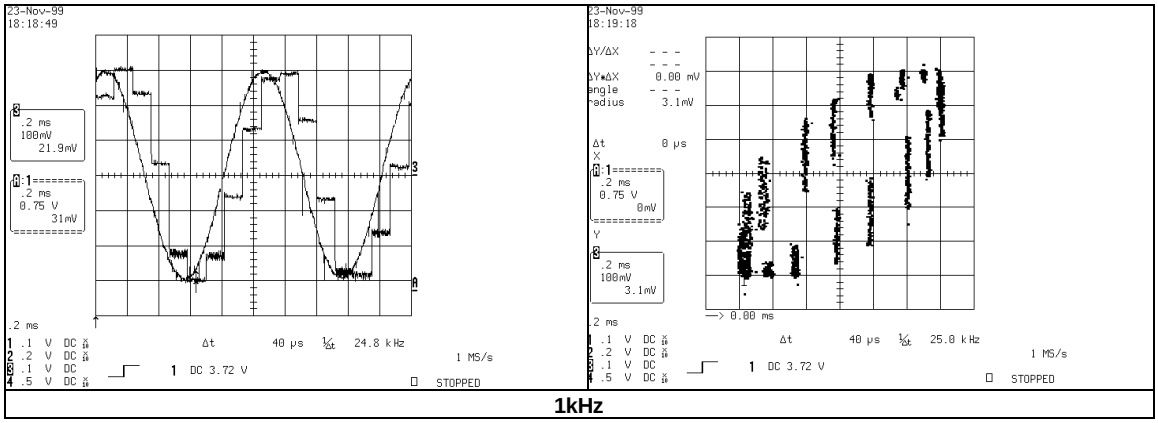
Trace A (down) = signal reconstructed through the D/A converter

The IN/OUT characteristics are presented at two different frequencies (50Hz and 1kHz) and it is possible to see the effect of sampling the current measurement at the PWM frequency.

The same characteristics are plotted as X-Y functions they are allowing us to observe their linearity.




50Hz




**1kHz**

**Figure 6 Transfer characteristics**

### *5-E Operation at overvoltage*

When the shunt resistor voltage is higher than the allowed maximum voltage at the IC input, IR2171/2172 automatically limits the pulse width and the resulting reconstructed signal is shown in Fig.7.
Trace 2=input voltage
Trace 3=D/A result

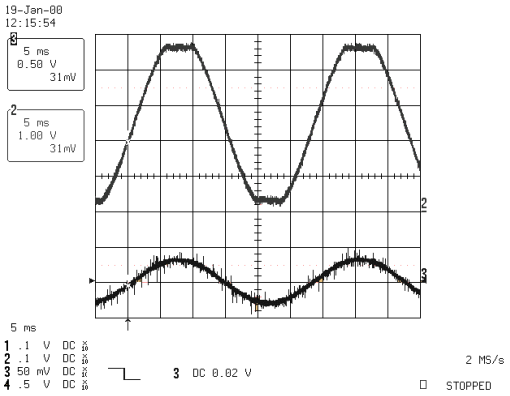



**Fig.7 Voltage limitation at IR2172(40kHz). Results for 36Hz and 300mV**

## *6. CONCLUSION*

The suitability of IR2171/2172 to the DSP-controlled three-phase AC drives is herein demonstrated and a possible software implementation of the phase current measurement is detailed