Application Report

SPRA605A – February 2000 – Revised August 2017

TMS320F240 DSP Solution for Obtaining Resolver Angular Position and Speed

Martin Staebler, Ankur Verma

ABSTRACT

This application report first presents a TMS320F240 DSP solution and then another unique approach using PGA411-Q1 for obtaining the angular position and speed of a resolver.

A resolver is an absolute mechanical position sensor used, for example, in servo applications. It is basically a rotating transformer. The sinusoidal input reference is amplitude modulated with the sine and cosine of the mechanical angle, respectively. These two output signals have to be decoded to obtain the angular position.

For decoding the resolver output signals, a basic method is introduced. It uses undersampling and an inverse tangent function. To achieve a higher angular resolution an improved method, which adds oversampling techniques is used. Due to an integral invariant filter, which is a combination of an IIR and FIR filter, the digitized angular position does not suffer any velocity lag.

Thanks to its peripherals, the Texas Instruments TI™ TMS320F240 DSP can be ideally used for decoding the two resolver output signals as well as for generating the input reference frequency. The hardware interfacing of the resolver to the TMS320F240 is shown and the software realization of the improved method on the TMS320F240 is documented. All software routines are C-compatible. The angular resolution achievable with the TMS320F240 is up to 14 bits, at a CPU loading of 13%.

The TMS320F240 will, therefore, eliminate the cost for the external resolver-to-digital conversion IC because the algorithm runs in addition to the motor control task.

The alternative approach uses PGA411-Q1 that is a resolver-to-digital converter, with an integrated exciter-amplifier and boost-regulator power supply, that is capable of both exciting and reading the sine and cosine angle from a resolver sensor. The integrated boost converter and exciter amplifier of the PGA411-Q1 reduces system cost and board space compared to traditional RDC solutions. On-chip protection and diagnostic improve robustness against short-circuit and increase safety by detecting external fault conditions. A high-speed, 3.3-V SPI allows PGA411-Q1 configuration, diagnostics, angle, and velocity information. An example of a TI design using PGA411-Q1 is the *EMC Compliant Single-Chip Resolver-to-Digital Converter (RDC)*. The TI design also includes example firmware on a C2000™ MCU to allow easy real-time evaluation of the TI Design.

Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

Digital signal processors are going to become more common in digital motor control applications. They permit sophisticated real-time control applications to be implemented, which improve dynamic response, precision, and efficiency. In addition, they enable sensorless control, which reduces total system operating by eliminating mechanical sensors.

TMS320F240 has been successfully used in sensorless applications. Mechanical sensors, which provide information on speed or position of the drive, can be eliminated and replaced by high-sophisticated position and speed estimation algorithms. Typical estimation algorithms include Extended-Kalman-Filters, INFORM, and Sliding Mode Observers. A further cost reduction can be achieved by the elimination of phase current sensors by a new algorithm that estimates the actual value of the three phase currents using only the DC-link current with a shunt resistor. However, there are still applications where sensorless control cannot achieve the required accuracy and reliability. This is especially true with respect to the angular position. Examples include servo applications, like robotics and numerically-controlled machine tools. Mechanical sensors used there are usually incremental encoders and resolvers. Incremental TTL-encoders provide a pulse train, where each pulse is equivalent to an incremental step. Incremental encoders with sinusoidal output voltage allow interpolation between two incremental steps and achieve a higher accuracy. Incremental encoders are relative position sensors. In contrast, resolvers are absolute angle transducers. They provide two output signals that always allow the detection of the absolute angular position. This and the fact that they suppress common-mode noise, makes them especially useful in a noisy environment. Because decoding the output signals of a resolver is not straightforward, application specific ICs, called resolver-to-digital converters, have been used to track the resolver's angular position. This application report presents a TMS320F240 DSP solution for resolver-to-digital conversion, which runs in addition to the motor control task. This will eliminate the cost for the external IC and provide improved performance and flexibility, since parameters can be easily modified by software.

Section 2 gives an introduction to the functionality of a resolver. Section 3 introduces the algorithm to obtain angular position and speed with a resolver. Section 4 then describes the hardware and software implementation on the TMS320F240 DSP controller. Finally Section 6 shows the results achieved with this TMS320F240 DSP solution.

2 The Resolver – An Absolute Angle Transducer

Resolvers are absolute angle transducers and are mounted on the motor shaft to get the absolute angular position of the motor. The accuracy is typically in the range of 5 arcmin. Resolvers are often used for angle sensing in noisy environment, due to their rugged construction and their ability to reject common-mode noise.

Resolvers are basically rotating transformers. A simplified functional diagram of a resolver and its corresponding signals over one mechanical revolution is depicted in Figure 1.

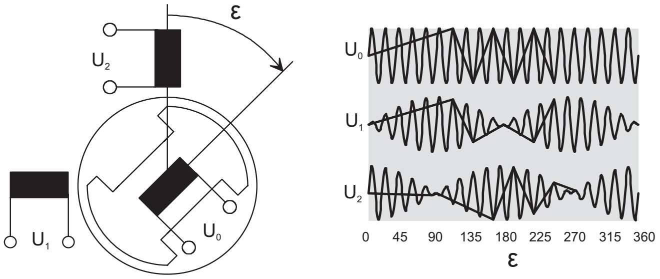


Figure 1. Resolver, Simplified Functional Diagram and Corresponding Signals

It basically consists of a rotor coil, with N windings and two orthogonal stator coils with usually N or N/2 windings. An alternating voltage, the reference signal, is coupled into the rotor winding and provides primary excitation. The reference signal is typically a fixed frequency in the range of 2 kHz to 10 kHz, with:

$$u_0(t) = \hat{u}_0 \times \sin\omega_{ref} t \tag{1}$$

The two orthogonal stator coils are wound, so that when the rotor shaft turns, the amplitude of the output signals is modulated with the sine and cosine of the shaft angle ε, relative to some zero, according to:

$$u_1(\varepsilon, t) = \hat{u}_0 \times k \times \sin\varepsilon \times \sin\omega_{ref} t \tag{2}$$

$$u_2(\varepsilon, t) = \hat{u}_0 \times k \times \cos\varepsilon \times \sin\omega_{ref} t \tag{3}$$

k is the transformation ratio of the resolver. Hence the shape of the resolver output signal u_1 and u_2 is equal to the sine and the cosine of the mechanical angle, respectively. Figure 2 shows the amplitude spectrum of the resolver output signals u_1, u_2, for a sinusoidal excitation signal u_0.

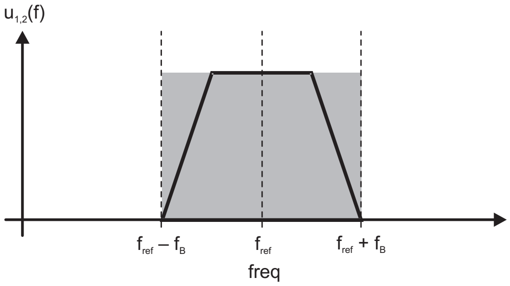


Figure 2. Amplitude Spectrum of the Resolver Signals u_1, u_2

As with amplitude-modulated signals, the spectrum is symmetrical to the reference frequency. The bandwidth f_B depends on the maximum angular speed according to:

$$f_B = \frac{1}{2\pi} \times \frac{d\varepsilon}{dt}\bigg|_{MAX} \tag{4}$$

3 Obtaining Angular Position and Speed of the Resolver

The method for obtaining and digitizing the angular position of a resolver is also known as resolver-to-digital conversion (R/D conversion). One basic method, using digital demodulation, is introduced in this section. It can be ideally implemented on the TMS320F240 DSP.

3.1 *Resolver-to-Digital (R/D) Conversion – Using Undersampling*

The basic method is depicted in Figure 3. The sine and cosine modulated output signals u_1 and u_2 must be sampled at the same frequency as the reference frequency. This, so called undersampling, demodulates both analog signals, so that the digitized samples $u_1(n)$ and $u_2(n)$ are sine and cosine of the angle e, respectively. This method can be ideally implemented on the TMS320F240. It incorporates dual ADCs, which can be synchronized to the reference frequency, generated by the on-chip PWM-unit.

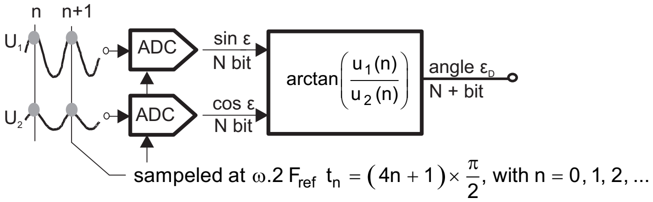


Figure 3. Resolver-to-Digital Conversion Using Undersampling and Inverse Tangent

The angular position can now be determined by a four quadrant inverse tangent function of the quotient of the demodulated sine and cosine samples. The inverse tangent function is ambiguous. Thus, the sign of the sampled signals has to be taken into account, to determine the absolute angular position according to:

$$\varepsilon(n) = \left\{ \begin{array}{c} \arctan\left(\frac{u_1(n)}{u_2(n)}\right), \text{ if } u_2(n) \geq 0 \\ \pi + \arctan\left(\frac{u_2(n)}{u_1(n)}\right), \text{ if } u_2(n) < 0 \end{array} \right\} \tag{5}$$

To be accurate, both signals, u_1 and u_2 have to be sampled simultaneously, at, or close to their maximum positive value, synchronized to the reference frequency according to:

$$\omega_{ref} t_n = (4n+1)\frac{\pi}{2},\ n = 0,1,2,..., \tag{6}$$

To avoid aliasing, the Nyquist criteria must be met. It requires the sample rate f_S to be at least twice the bandwidth f_B of the interesting signal. To meet that, an analog anti-alias filter has to remove any frequency components outside the band-of-interest $f_{ref} \pm f_B$. Referring to Figure 2, it is obvious that any DC offset has to be removed prior to sampling. Otherwise, it would be added to the demodulated sine and cosine signals and decrease accuracy.

For an N-bit ADC, the angular accuracy achievable is N+1-bit. With the dual 10-bit ADCs integrated on the TMS320F240, the angular accuracy achievable is 10 arc minutes. A higher resolution and better noise rejection are achievable by oversampling and averaging techniques, which are discussed in the following section.

3.2 R/D Conversion – Improved Method Using Oversampling

3.2.1 Overview

The total algorithm, using oversampling, is depicted in Figure 4. The high-resolution digital angle e_m is the output of the position-tracking closed loop.

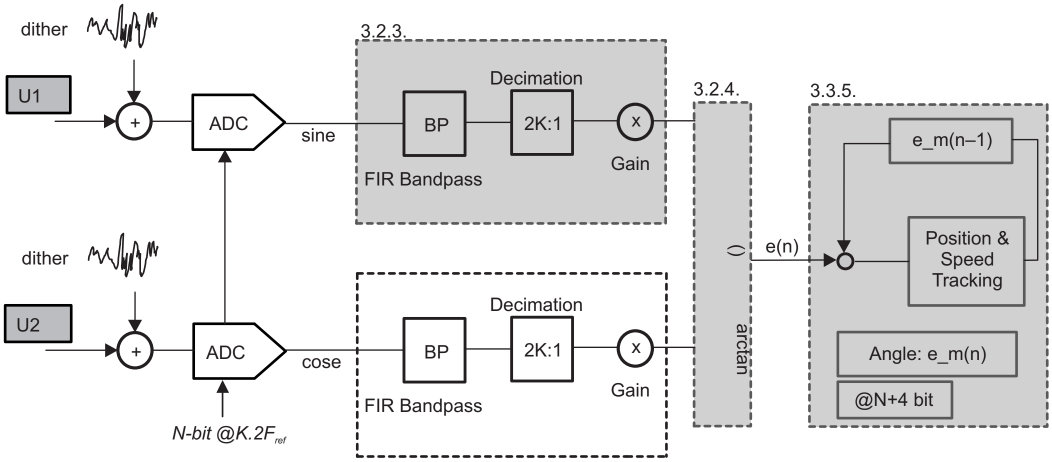


Figure 4. Block Diagram of the Improved Resolver-to-Digital Converter

The sine and cosine modulated resolver output signals u_1, u_2 are sampled at 2K-times the reference frequency. This is equivalent to K-times oversampling. Dither is added prior to sampling, to ensure that the quantization noise is not correlated to the resolver output signals. The FIR bandpass filter acts as a digital anti-alias filter and improves the resolution of the samples by reducing the bandwidth to $f_{ref} \pm (f_{ref}/2)$. The decimation is done taking only every $2K^{th}$ sample. This demodulates the bandpass-filtered samples and reduces the sample rate to f_{ref}. At that point, the (filtered) sine and cosine samples of the resolver angular position are available. The angular position is now derived by the arc tangent of the quotient of the demodulated sine and cosine samples. Due to the averaging FIR filter, the resolution of the angular position has been improved. On the other hand, it shows a velocity lag. The position and speed interpolation closed loop is added to further improve the resolution of the angular position. The achievable improvement depends on the bandwidth of the closed loop. Another important task of the closed loop is that it exactly compensates the group delay of the FIR bandpass filter. Thus, the estimated angle e_m shows a higher resolution, but does not suffer any velocity lag.

3.2.2 Dither – Adding Random White Noise

To apply oversampling or averaging techniques, the quantization error e(n) must not be correlated with the input signal . This is obviously not true for example, for DC signals, where the quantization error e(n) remains constant. In that case the quantization error is correlated and averaging can not improve the resolution. For example, a solution to that problem is to add random white noise to the analog signal, if noise is not already present. The additive noise must have a root-mean-square value of:

$$rms(noise) = \sqrt{q/12},\ q = 1LSB \tag{7}$$

This ensures that the quantization error is not correlated with the input signal. Note that this does not increase the total quantization error.

3.2.3 FIR Decimation Bandpass-Filter

The decimation bandpass filter is a symmetrical FIR filter, which has got a linear phase. The number of taps depends mainly on the oversampling ratio. The center frequency of the FIR filter is set equal to the resolver reference frequency, as shown in Figure 5.

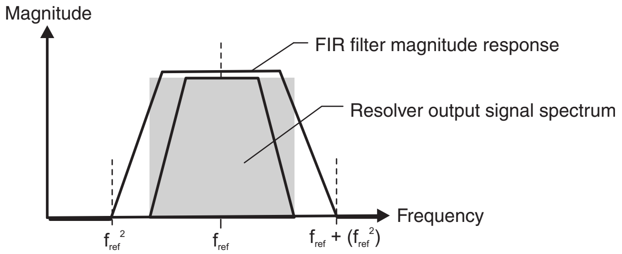

Magnitude
FIR filter magnitude response
Resolver output signal spectrum
Frequency
$f_{ref}{}^2$
f_{ref}
$f_{ref} + (f_{ref}{}^2)$

Figure 5. FIR Decimation Bandpass Filter Magnitude Response

The task of the FIR filter is to reduce the bandwidth to [f_{ref}±(f_{ref}/2)]. This is required prior to decimation to fulfill the Nyquist criteria. The FIR filter is, in fact, a digital anti-alias filter, which additionally improves the resolution within the band-of-interest. For K-times oversampling, decimation in time is achieved by taking only every 2Kth output of the FIR filter. That is equivalent to calculating the filter for only every 2Kth input sample. Decimation in time will demodulate the signals down to the base-band [0-f_{ref}/2]. Then the sine and cosine of the resolver angular position are at a higher resolution. As a rule of thumb the resolution increases by 3 dB (equivalent to 0.5 bit), each time the bandwidth is halved. On the other hand the signals are delayed. This is due to the constant group delay of the symmetrical FIR filter. For N taps, the group delay is exactly (N–1)/2 samples. When the FIR filter has got (4K+1) taps, the group delay is equivalent to one period of the reference frequency.

3.2.4 Arc Tangent Function

The angle is derived by taking the arc tangent of the quotient of the FIR-filtered sine and cosine signals, as outlined in Section 3.1.

$$\varepsilon_{FIR}(n) = \left\{ \begin{array}{l} \text{arctan}\left(\dfrac{u_{1,FIR}(n)}{u_{2,FIR}(n)}\right), \text{ if } u_{2,FIR}(n) \geq 0 \\ \pi + \text{arctan}\left(\dfrac{u_{2,FIR}(n)}{u_{1,FIR}(n)}\right), \text{ if } u_{2,FIR}(n) < 0 \end{array} \right\} \tag{8}$$

Compared to the basic method, shown in Section 3.1, the digitized angle now has got a higher resolution, due to the averaging FIR filter. However, as depicted in Section 3.1, the digitized angle shows a velocity lag, due the group delay of the FIR filter. The error of the digitized angle is proportional to the rotational speed.

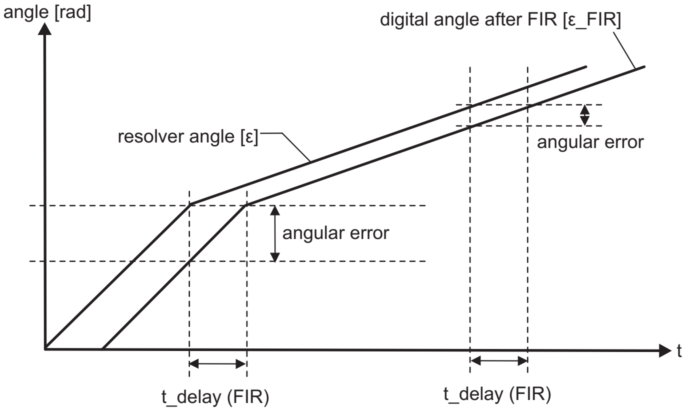


Figure 6. Angular Error as Function of the Rotational Speed After the FIR Filter

A closed-loop position and speed interpolator will be used to compensate for the angular error and to further improve the angular resolution.

3.2.5 Closed-Loop Angular Position and Speed Interpolator

The closed-loop, shown in Figure 7, consists of a PI controller, a 1st order IIR filter and an integrator. It is basically a lowpass filter. It is a type 2 closed-loop (two integrators) and hence does not have an integral error. The parameters of the closed loop define bandwidth and characteristic and are discussed in the next section.

The task of the closed loop is to:

- Improve the accuracy of the angular position e_m. The accuracy depends on bandwidth selected.
- Compensate the delay of the FIR filter, so that e_m suffers no velocity lag
- Derive the angular speed w_m

The result will be a higher accuracy of the interpolated angular position e_m. Additionally, e_m does not suffer a velocity lag. This is achieved by the following. The output signal e_m is delayed with the same time as the FIR filters group delay. Hence the delayed output signal e_m(n–1) is also compared with the delayed FIR filtered angle e_FIR. The closed loop forces both delayed angles to be identical. For constant speeds, the output e_m of the closed loop is then identical with the real resolver shaft angle.

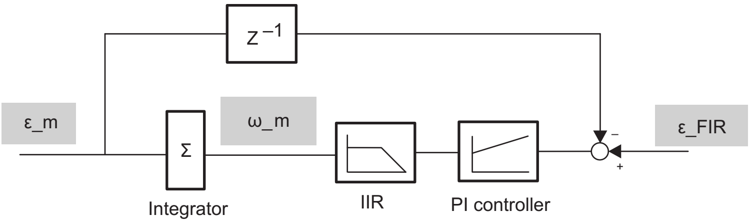


Figure 7. Closed-Loop Position and Speed Interpolator

4 TMS320F240 DSP Implementation

4.1 *Overview*

The TMS320F240 DSP controller incorporates two 10-bit A/D converters, which sample both analog inputs simultaneously. The A/D sampling point can be triggered by any timer event, hence synchronously to the on-chip PWM. One PWM channel (part of the event manager) is used to generate the sinusoidal reference frequency.

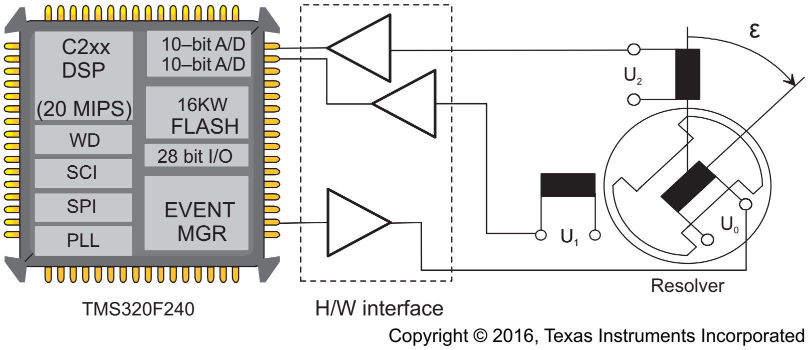


Figure 8. Functional Block Diagram of the TMS320F240 R/D Converter Realization

Figure 8 shows the TMS320F240 single-chip DSP solution for the resolver-to-digital conversion. The DSP performs the following tasks:

- Generates the sinusoidal reference frequency [Event Manager]
- Sample both resolver output signals simultaneously [ADC]
- Synchronizes the sampling point to the sinusoidal frequency [ADC, Event Manager]
- Performs the improved R/D conversion algorithm outlined in Section 3 [CPU]

4.2 TMS320F240 Hardware Interface

Figure 9 shows the hardware interface of the TMS320F240 to a resolver.

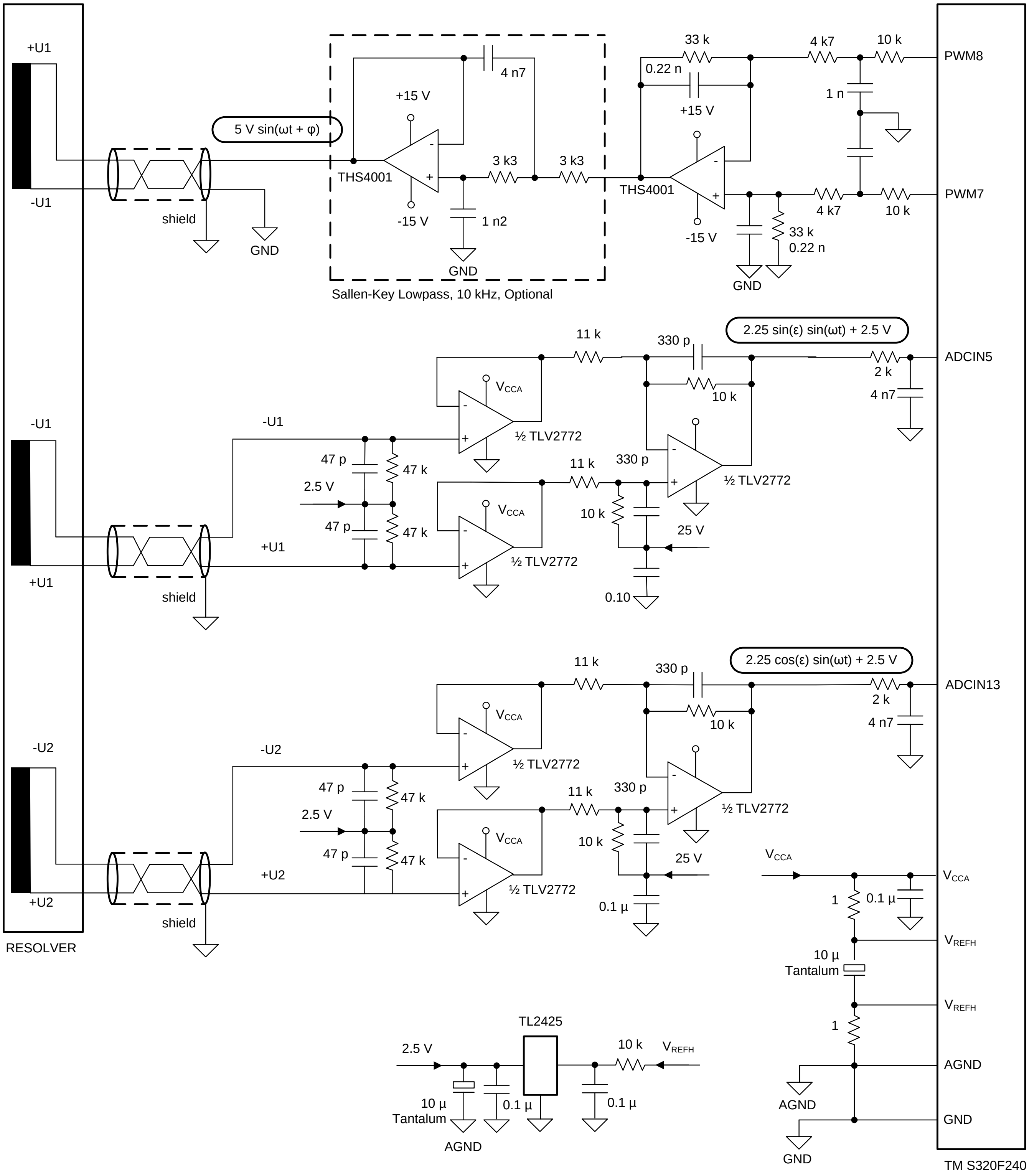


Figure 9. Signal Conditioning for Resolver to TMS320F240 Interface

peak-to-peak voltage and a resolver with a transformation ratio of 2:1. Thus, the maximum peak-to-peak voltage of the two modulated resolver output signals is 5 VPP. The circuit performs the following:

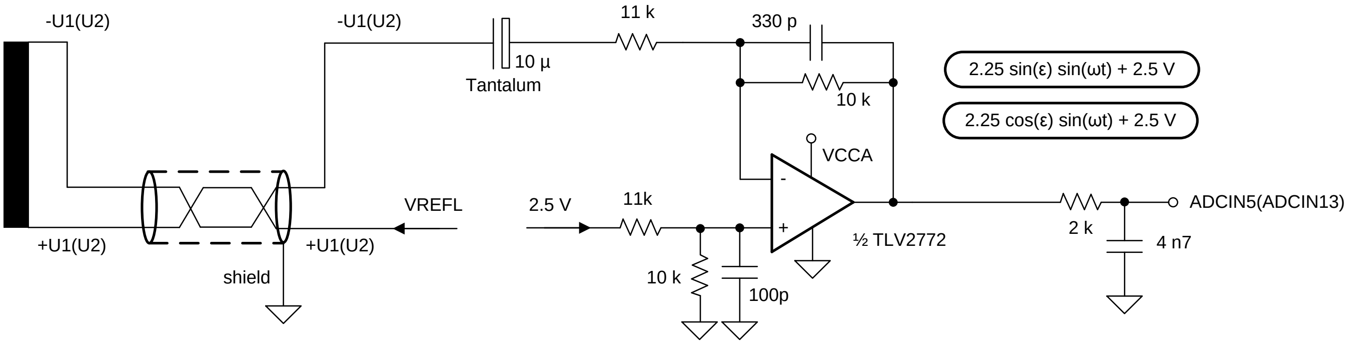
- Signal conditioning of the PWM output to drive the resolver with a sinusoidal reference frequency u_0.

The PWM channels PWM7 and PWM8 (differential signals) are used to generate the sinusoidal reference frequency. The first op amp THS4001 is used to filter out the PWM carrier signal, to do the level shifting and amplification to get a sine wave at 10-Vpp output voltage swing. The second op amp performs a Sallen-Key lowpass filter and further reduces harmonic distortion.

- Signal conditioning and level shifting of both resolver output signals, u_1, u_2 to drive the TMS320F240 A/D converters.

  The differential output signals of the resolver are fed into noninverting buffers TLV2772 and referenced to the virtual ground. The following differential amplifier TLV2772 suppresses common-mode noise and performs the level shifting to meet the 5-V A/D input range. The TL2772 are single supply, rail-to-rail output CMOS op amps. The RC network performs the anti-alias filter. The capacitor is chosen to drive the A/D module switch capacitor input, rather than the op amp. Therefore the capacitor should be as close as possible to the analog input pins ADCIN5 and ADCIN13, respectively.
- The TL2425 provides the 2.5-V virtual ground, to operate with single 5-V supply only.
- TMS320F240 A/D input voltage range is 0 – 5 V.

  Figure 10 shows a simpler solution for single-ended resolver output signals. It requires only one single op amp to do buffering and level shifting. However, due to AC-coupling the gain of each channel is a function of the line impedance.



Copyright © 2017, Texas Instruments Incorporated

Figure 10. Signal Conditioning for Single-Ended Resolver Signals

4.3 TMS320F240 Software Implementation

4.3.1 Program Structure

The software is mainly written in ANSI C language to achieve a modular and readable software structure. Only time critical functions, like the FIR filter, arc tangent, and so forth, are written in run-time optimized assembler, but provide a C-compatible interface. Hence these functions can also be called from any C program.

The software integrates all modules required to obtain the resolvers angular position and speed. It can be easily integrated, for example, in a standard field-oriented control. The source files and a short functional description are given in Table 1.

The software is basically structured into three main functional blocks. Each functional block is realized as a C function, which might call other functions. The following three sections give a detailed description on these functions.

Table 1. Source Modules and Functional Description

SOURCE MODULE	C CALLABLE FUNCTIONS	FUNCTIONAL DESCRIPTION
MAIN.C	main()	Main program
INIT.C	Resolver_Init()	’F240 initialization
T3_INT.C	timer3_int()	Interrupt service routine
CONTROL.C	control()	R/D conversion, improved method

Table 1. Source Modules and Functional Description (continued)

SOURCE MODULE	C CALLABLE FUNCTIONS	FUNCTIONAL DESCRIPTION
FIR_DEC8.ASM	int FIRBandpass_17A_Dec8 (* int)	17-tap FIR decimation bandpass filter
CL.C	Resolver_ClosedLoop()	Closed-loop position and speed interpolator
ATAN_DIV.ASM	int atan_div(int nom, int denom)	Inverse tangent of the quotient sin/cos

4.3.2 TMS320F240 DSP Initialization

Function: void Resolver_Init(void)

This initialization routine is called from the main program and sets up the TMS320F240 and its peripherals. The function does not pass any argument and does not return a value.

The TMS320F240 is set up with a CPU clock of 20 MHz. The peripherals simple compare PWM unit, timer 3 and the ADC module and the timer 3 period interrupt are initialized.

- Timer 3 period interrupt is the main time base. The cycle time is 25 µs, hence the reference frequency period is 200 µs, or 5 kHz.
- The simple compare units are used to generate the sinusoidal reference frequency and noise, respectively. Timer2 is selected as the PWM time base. The timer2 period is set to 1/4 of the timer 3 period. This will give a PWM period of 6.25 µs. The reason for that is to reduce the ripple on the sine wave. This relaxes the requirements to the analog reconstruction filter.
- The resolver output signals, connected to ADCIN5, ADCIN13 are automatically sampled on a timer3 compare event. This ensures that the sampling points are synchronized to the PWM unit, which generates the reference frequency. Triggering the conversion by a timer3 compare event allows optimizing the phase at which the signal is sampled. The timer3 compare value is initially set to 12.5 µs. This means that the signals are sampled 12.5 µs before the interrupt occurs. Any modification of the timer3 compare value shifts the sampling point and hence the phase at which the resolver signals are sampled within ±22.5 degrees. However, the conversion time of 6 µs, prior to the timer3 period interrupt must still be met.
- All timers are synchronized and started synchronously

NOTE: The corresponding header file INIT.H contains the values for the reference frequency and the gain error of the two input channels at the analog inputs ADCIN5 and ADCIN13.

4.3.3 Interrupt Service Routine

Interrupt: void interrupt timer3_int(void)

The timer 3 period interrupt is the only interrupt used. The interrupt cycle time is set to 25 ms. This is equivalent to 1/8 of the sinusoidal reference period of 200 ms or a 5-kHz frequency. The flowchart is depicted in Figure 11.

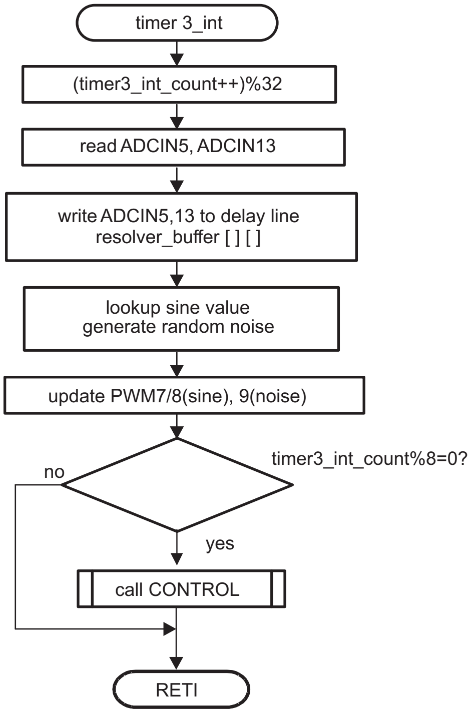


Figure 11. Flowchart of timer3_int

The variable timer3_int_count is used as a pointer and as a scheduler. It is incremented each timer 3 interrupt within the range 0–31. This modulo 32 operation is done by simply masking the upper bits 5–15, respectively.

The tasks of the timer3_int interrupt routine are as follows:

- 4 × oversampling of the sine and cosine modulated resolver signals u_1, connected to ADCIN5 and u_2, connected to ADCIN13 The ADC sample rate is synchronized with the sinusoidal reference frequency, which is generated by the TMS320F240 on-chip simple compare unit. The ADC sampling point is triggered automatically by a timer3 compare event. This allows compensating a phase shift of up to ±22.5°C between the reference frequency u_0 and the modulated resolver output signals u_1 and u_2. Thus, a modification of the timer3 compare value shifts the phase at which the signals are sampled. Note, that the timer3 compare event must occur no later than 6 µs before the timer3 period interrupt, to meet the ADC conversion time of 6 µs. The data of ADC1 FIFO (result of channel 5) and ADC2 FIFO (result of channel 13) is divided by two to get a headroom of 6 dB. An offset is added to convert to a 2s complement or Q15 fractional number. These Q15 numbers are stored into the delay line resolver_buffer[][]. The variable timer3_int_count is used to point to the latest position in the delay buffer according to:

```
resolver_buffer[0][timer3_int_count] = ADCFIFO1/2 + 32768
 resolver_buffer[1][timer3_int_count] = ADCFIFO2/2 + 32768
```

  The lower array resolver_buffer [0] [] always stores the latest 32 sine modulated samples in Q15 notation with a 6-dB headroom. The upper array resolver_buffer[1][] stores the latest 32-cosine modulated samples, respectively.

- Resolver reference frequency generation through the simple compare PWM unit The simple compare PWM unit, used to generated the sinusoidal reference frequency, is updated in this interrupt — every 25 ms. This is equivalent to 8 updates during one period of the sine wave. Therefore a sine wave lookup table is used, which stores the corresponding 8 PWM duty cycles. The elements are accessed according to:

  PWM = sine_table [(timer3_int_count + sine_phase_lag) % 8]

  The variable sine_phase_lag is used as an offset to the pointer timer_int_count and is initialized, so that the output of the FIR filter is at a maximum.

To reduce harmonics the simple compare PWM period is 6.25 ms, which is 1/4 of timer3 period. This reduces the amount of analog filtering and leads to a sine wave with lower harmonic distortion.

- Random noise generation and update through the simple compare PWM9 unit

  The noise generator is a random number generator, which calculates a new random number each timer 3 interrupt. The new random number is stored to the compare register of PWM9.

- Scheduling of the control() routine on every 8th interrupt.

  The function control() is called on every 8th timer 3 interrupt. It performs the resolver-to-digital conversion algorithm and is explained in the next section.

4.3.4 R/D Conversion Using Oversampling

This algorithm is realized using a level one function control(), written in C. This function calls three level two functions, written in run-time optimized assembler. These are FIR_Bandpass_17A_Dec8 (* int), atan_div(int, int), Resolver_ClosedLoop(). Resolver position and speed are stored in the two global variables

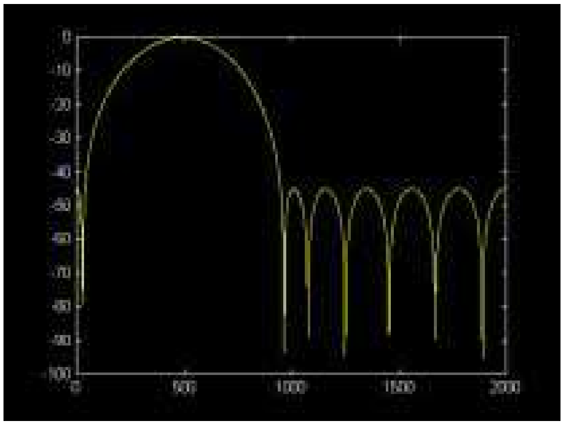
- epsilon_m = high-resolution position without velocity lag
- omega_m = angular speed

Function: void interrupt control(void)

This function is called on every 8th timer 3 interrupt. With a 5-kHz reference frequency it is called every 200 µs. It performs the resolver-to-digital converter for obtaining the high-resolution angular speed and position without any velocity lag. The listing of this function is shown below.

```
void interrupt control(void)
{
int r_sin, r_cos;
asm(" eint"); /* enable interrupt */
/*--------------------------------*/
/* 17-tap FIR decimation bandpass */
/*--------------------------------*/
index = &resolver_buffer[0][timer3_int_count] /* Ptr to latest */
/* element in delay line */
r_sin = FIRBandpass_17A_Dec8(index)
r_cos = FIRBandpass_17A_Dec8(index+BUFFER_SIZE)
/*--------------------------------*/
/* Gain compensation (optional) */
/*--------------------------------*/
r_sin = (int) ((long)R_SIN_GAIN_Q15 * (long)r_sin >> 15);
r_cos = (int) ((long)R_COS_GAIN_Q15 * (long)r_cos >> 15);
/*----------------------------------------------------*/
/* Angle determination using arc tangent function */
/*----------------------------------------------------*/
epsilon = atan_div(r_sin,r_cos); /* 200us delayed angle */
/*------------------------------------------------------*/
/* closed-loop angle and speed interpolation */
/*------------------------------------------------------*/
/* epsilon_m = actual position WITHOUT velocity lag */
/* omega_m = actual scaled speed */
/*------------------------------------------------------*/
Resolver_ClosedLoop();
}
```

Note: The function control() could be used to include a PI current controller. If a faster current controller cycle time is required, control() might be called on every 4th timer 3 interrupt and the resolver-to-digital algorithm is skipped every one, respectively. Function: int FIR_Bandpass_17A_Dec8 (* int) This function performs the bandpass filtering and decimation by 8, because it is called only every 8th input sample. The variable sine_phase_lag is used as an offset to the pointer timer_int_count. It is initialized so, that the FIR filter takes always the window of 17 samples, which leads to the maximum output amplitude. The argument passed to the filter is the pointer to the latest sample (start of the window) in the delay line. The return value is the filtered signal. The filter coefficients are designed with the Digital Filter Design Package [7]. It is a symmetrical FIR bandpass filter with 17 taps and a 5-kHz center frequency. The magnitude response is shown in Figure 12.



Frequency scaled by 10.

Figure 12. Magnitude Response of the 17-Tap FIR Filter

This FIR filter reduces the S/N ratio within the signal band-of-interest 5 kHz ± 2.5 kHz by 9 dB. This is equivalent to a bit gain of 1.5-bit. Another advantage of the FIR bandpass filter is that noise outside the band-of-interest 5 kHz ± 2.5 kHz is attenuated by more than 40 dB. This is especially required to remove any DC offset.

Function: int atan_div(int nom, int denom)

Arguments: nom, denom: Q15 (–1.0,..., 0.999)

Return: angle, Q15 (–1.0,..., 0.999), scaled by PI

Error: < 3 LSB

Function: void Resolver_ClosedLoop(void)

This function compensates the velocity lag of the angle, which is derived after the inverse tangent function. It further improves the accuracy of the angular position and also calculates the angular speed. The achievable resolution depends on the bandwidth. The function requires the following global input and output variables: Global Input variables: epsilon (FIR filtered angle with velocity lag) Global output variables: epsilon_m (high-resolution angle without velocity lag) omega_m (high-resolution speed).

Figure 13 shows the block diagram of the closed loop. The settings of the parameters, which define the bandwidth, are shown in the next section.

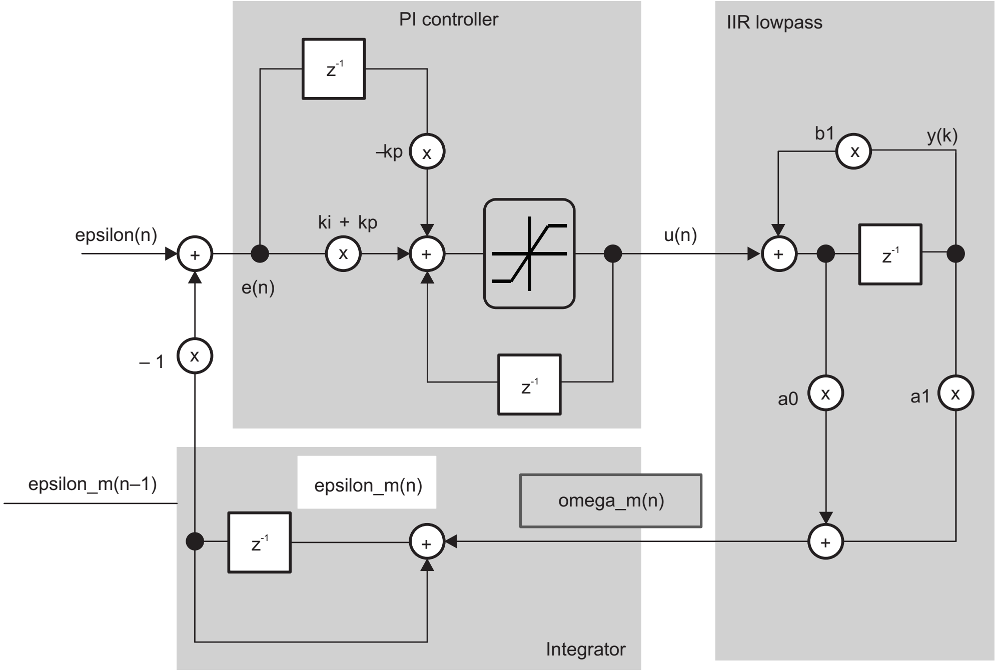


Figure 13. Closed-Loop Angle and Speed Interpolator

Figure 14 shows the normalized step response of the closed loop. The overshot is inherent in a critically damped 2nd order integral invariant (type-2) filter.

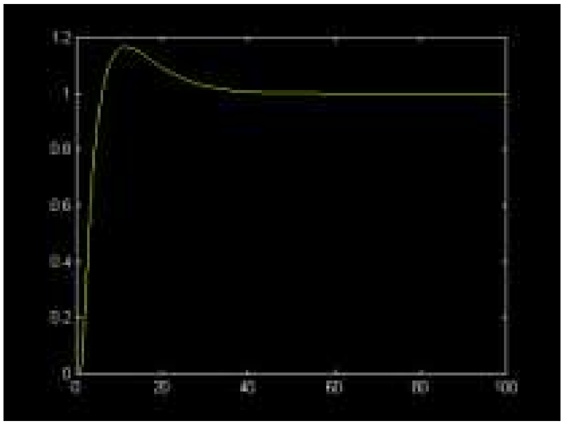


Figure 14. Normalized Step Response of the Interpolated Angle

4.4 Changing Parameters

4.4.1 Reference Frequency

The resolver reference frequency and the gain compensation of the input signals at ADCIN5 and ADCIN13 can be changed in the header file INIT.H

```
/*=============================================================*/
/* RESOLVER CONSTANTS */
/*=============================================================*/
#define R_SIN_GAIN 0.976 /* [0 .. +1.0] */
#define R_COS_GAIN 0.976 /* [0 .. +1.0] */
/* Reference frequency period in [us]*/
```

```
#define REF_FREQ_PERIOD 200 /* [us] */
/* Note: only 4[us] steps allowed */
```

4.4.2 Angular Resolution and Bandwidth

The achievable angular resolution depends on the bandwidth selected. It can be changed in the file CL.C, where the function Resolver_ClosedLoop() is located. Only two parameters KI and KP have to be changed, as outlined in Table 2:

```
/*---------------------*/
/* PI controller variables */
/*--------------------------*/
int KP = (int) (0.30 * 32767);
int KI = (int) (0.026 * 32767);
```

Table 2. Angular Resolution and Bandwidth as Function of KI and KP

RESOLUTION	BANDWIDTH	KI (Q15)	KP [Q15]
13 bit	600 Hz	0.026	0.3
13.5 bit	300 Hz	0.005	0.1
14 bit	150 Hz	0.001	0.05

5 PGA411-Q1 Implementation

5.1 *Overview*

The PGA411-Q1 is the highest integrated RDC accepting up to 20-kHz sine and cosine signals from the resolver sensor to output a 10- or 12-bit digital word representing angle or velocity. The resolver can be excited by the PGA411-Q1 on-chip excitation amplifier with sine wave input referenced to internally or externally generated clock. Optionally, an external amplifier can be used to extend the output current capability beyond 150 mA. The PGA411-Q1 is also equipped with a boost regulator supplying power to the onboard excitation amplifier. A tracking loop employing a Type-II PI Controller is integrated to determine the angle and velocity value based on the input signals. The maximum tracking rate of the device is 200,000 rpm. The integrated continuous diagnostics monitor and the internal diagnostics engine can signal a fault condition by a dedicated pin, which can be used as an MCU interrupt.

5.2 *PGA411-Q1 System Architecture*

PGA411-Q1 comprises of RDC architecture that converts analog resolver signals into digital angle and velocity outputs. The analog front-end (AFE) consists of programmable gain amplifiers and a comparator. The AFE block conditions the output signals of the resolver by removing noise, sets correct input DC bias, and appropriately gains up the AC signal to be used by the subsequent blocks. A digital feedback loop is the main part of the RDC conversion. It starts by assuming a digital angle, phi. This angle is digitally processed using the sine and cosine lookup tables stored in memory. This in turn is fed to the corresponding sine and cosine digital-to-analog converters (DACs). Texas Instruments, Design considerations for resolver-to-digital converters in electric vehicles, Technical Brief (SLYT661) describes the system architecture in more detail.

5.3 *PGA411-Q1 Software Design*

Use the *Software Developer Guide for PGA411-Q1* (SLAA708) to understand high-level software flow for platform development with the PGA411-Q1 device.

5.4 *PGA411-Q1 Troubleshooting Guide*

Use *Troubleshooting Guide for PGA411-Q1* (SLAA687) as the first step in troubleshooting any issues while using the PGA411-Q1 device. This guide includes example waveforms and information on faults that assist in the process to troubleshoot issues.

5.5 PGA411-Q1 System Modelling in MATLAB Simulink

Different automotive subsystems have diverse requirements. For example, the belt-driven alternator/starter system can have an acceleration of 50,000 revolutions per minute per second (RPM/s); whereas for crank-driven systems, acceleration can be in the order of 20,000 RPM/s. Similarly, industrial applications may have a highly dynamic servo motor that can accelerate from 0 to 5,000 RPM within as little as two to three milliseconds. Some applications involve hard braking and instant acceleration where the motor stops because of an obstacle. For this application, acceleration or deceleration can be as high as 200,000 rad/s2. In this case, higher position errors are temporarily tolerated, however, returning to true position values is required within a few milliseconds. The delay from reading the input position to outputting the angle has to be known accurately for proper motor control. To simulate the performance of a RDC in the powertrain system and analyze real events such as hard braking and sudden acceleration in the automobile, take a look at document here.

6 TMS320F240 Results

6.1 Processor Utilization

The R/D conversion algorithm can be subdivided into four functions. Table 3 gives an overview on the total CPU cycles for each subfunction.

Table 3. TMS320F240 CPU Loading

FUNCTION	CPU CYCLES
timer3_int()	76
atan_div()	103
Resolver_ClosedLoop()	56
FIRBandpass_17A_Dec8()	36

All functions, except timer3_int(), are called every 200 µs, for a 5-kHz reference frequency. The interrupt routine is called 2-times the selected oversampling rate K. Thus, the total CPU loading depends mainly on the oversampling rate K.

The CPU loading, for the algorithm using 4-times oversampling and a 5-kHz reference frequency is approximately 13%. The CPU loading for two times oversampling will be approximately 8%.

6.2 Angular Accuracy and Transient Response

The angular position resolution achievable with the TMS320F240 ranges from 12 to 14 bits, depending on the closed-loop bandwidth. The digital angular position does not show a velocity lag, thanks to the closed loop. The normalized angular step response is shown in Figure 15. The rise time depends on the bandwidth selected.

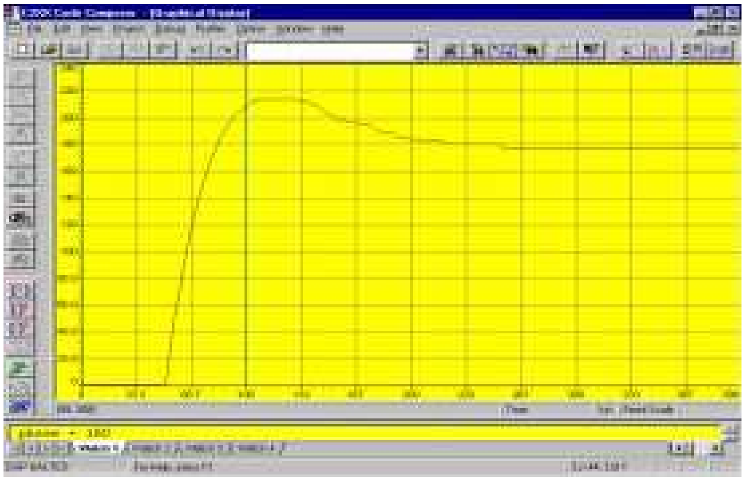


Figure 15. Measured Normalized Step Response for a 1º Angular Step (1º ~ 180)

The achievable angular accuracy and the settling time are shown in Table 4 for a 1º angular step.

Table 4. TITLE NEEDED

BANDWIDTH	SETTLE TIME	RESOLUTION
600 Hz	8 ms	13-bit
300 Hz	18 ms	13.5-bit
150 Hz	40 ms	14-bit

To test the dynamic error and accuracy of the R/D converter implementation on the TMS320F240, the two output signals of a resolver where generated with a [LeCroy] signal generator. The function realized was an angular speed reversal from –180 rpm to +180 rpm, as shown in Figure 16.

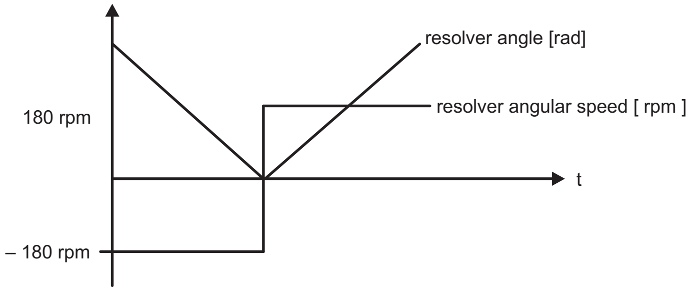
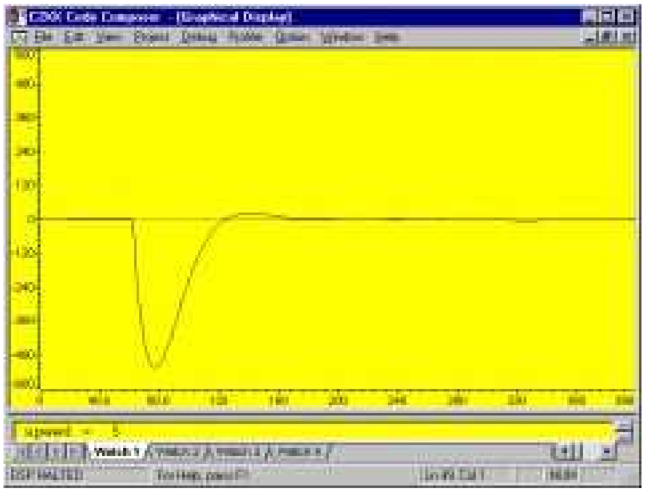


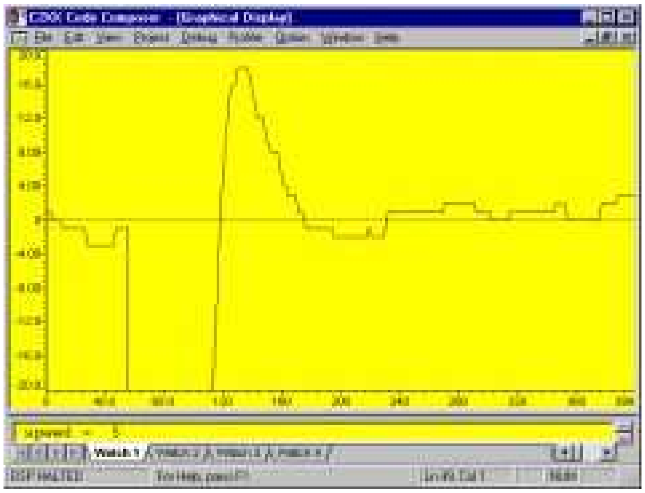
Figure 16. Resolver Angular Position and Angular Speed

Figure 17 and Figure 18 show the absolute angular position error for a speed reversal from –180 rpm to 180 rpm. The closed-loop bandwidth was set to 300 Hz. The digitized angle is normalized to a 16-bit value, where 0x0000 represents 0° and 0xFFFF represents 360°, respectively. The y-axis corresponds to the angular error in bit, with respect to the 16-bit value. 1-bit is equivalent to 0.33 arcmin. The x-axis one step corresponds to 200 ms.



Closed-loop bandwidth = 300 Hz
Y-axis in bit; 1 bit = 0.33 arcmin

Figure 17. Angular Position Error for a Speed Reversal From –180 rpm to +180 rpm



Closed-loop bandwidth = 300 Hz
Y-axis in bit; 1 bit = 0.33 arcmin

Figure 18. As Figure 17, but Y-Axis Zoomed

It can be seen that the angular error, after the bandwidth specific settling time, does not depend on the angular velocity. The digital angle does not suffer any velocity lag. The angular error remains within ±3 LSB which is equivalent to an accuracy of approximately 13.5 bit.

7 Conclusion

The resolver-to-digital converter has been successfully implemented on the TMS320F240 DSP controller. The angular accuracy achievable is up 14 bits. Thanks to the integral invariant filter, realized as a combination of FIR and IIR filter, the digitized angle does not suffer any velocity lag. The angular accuracy is comparable with application specific ICs. The digital filter realization allows changing the bandwidth by software, allowing the user to determine a suitable trade-off time between bandwidth and resolution. Adaptive bandwidth as a function of the rotation speed is possible. The CPU loading is approximately 13%. Hence, the DSP has got enough power to perform the R/D conversion in parallel to sophisticated digital control algorithms, like the field-oriented control. Typical applications are numerical control and robotics in noisy environment, using a synchronous motor (PMSM) with a resolver as absolute position sensor. For these applications this TMS320F240 DSP solution reduces total system cost, due to the elimination of an external R/D converter. The TMS320F240 solution even does not require an external oscillator.

The alternative approach discussed uses PGA411-Q1 that is a resolver to digital converter, with an integrated exciter-amplifier and boost-regulator power supply, that is capable of both exciting and reading the sine and cosine angle from a resolver sensor.

8 References - make clickable links

1. TMS320C24x DSP Controllers – Reference Set: Vol.1 and 2, SPRU160 and SPRU161 Texas Instruments Inc., 1998.
2. TMS320F240 Data Sheet, SPRS063 and SPRS064, Texas Instruments Inc., 1998.
3. TMS320C2x/C2xx/C5x Optimizing C Compiler User's Guide, (SPRU024), Texas Instruments Inc., 1995.
4. TLV2772 Advanced LinCMOS Rail-to-Rail Dual Op Amp, Data Sheet, SL0S209 Texas Instruments Inc.,1998.
5. Goepel, W.: Sensors A Comprehensive Survey, VCH, Weinheim, 1989.
6. Jonacha,H.: Integralinvariantes Antialiasing-filter fuer inertiale Messysteme, Technisches Messen 59 (1992), 7/8, Oldenburg Verlag.
7. TMS320 ASPI Digital Filter Design Package for PC, Atlanta Signal Processors Inc., 1990
8. Ankur Verma and Anand Chellamuthu, “Design considerations for resolver-to-digital converters in electric vehicles,” Texas Instruments Analog Applications Journal (SLYT661), 1Q 2016.
9. Ankur Verma and Amanda Weise, “Rotary position sensing for electric vehicles,” Part 1 and Part 2, EDN, October 2, 2015.
10. Ankur Verma and J. Panacek. “PGA411-Q1 PCB Design Guidelines,” Texas Instruments Application Report (SLAA697), March 2016.
11. Ankur Verma and F. Xu. “Troubleshooting Guide for PGA411-Q1,” Texas Instruments Application Report (SLAA687), February 2016.
12. Ankur Verma and Brian Rodriguez, “Electrical design considerations for industrial resolver sensing applications,” TI White Paper (SLYY100), June 2016.
13. Ankur Verma, “Software Developer's Guide for PGA411-Q1,” Application Report (SLAA708) July 2016.
14. Ankur Verma, Fei Xu and Jiri Panacek, “PGA411-Q1 Step-by-Step Initialization with Any Host System,” TI Application Report (SLAA688), March 2016.
15. Ankur Verma, “Accuracy? Resolution? Arc minutes? How to take charge of your motor control design,” TI E2E™ Motor Drive & Control blog, August 17, 2016.

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (February 2000) to A Revision	Page
• Added new content to the abstract	1
• Added *PGA411-Q1 Implementation* section	7
• Added new content to the *Conclusion* section	19
• Added references	20

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated (‘TI”) technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, “TI Resources”) are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI’s provision of TI Resources does not expand or otherwise alter TI’s applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED “AS IS” AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI’s standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265