J Electr Eng Technol.2018; 13(1): 192-200
http://doi.org/10.5370/JEET.2018.13.1.192

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

Enhancing the Accuracy for the Open-loop Resolver to Digital Converters

Fikret Anıl Karabeyli* and Ali Ziya Alkar†

Abstract – In this study, improvements for error correction, speed, position, and rotation calculation algorithms have been proposed to be used in resolver to digital conversion (RDC) systems. The proposed open-loop system drives the resolver and uses the output signals of the resolver signal to estimate the real time position, the instant speed, and the rotation count with high resolution and accuracy even at high speeds and noise. The proposed solution implements strong features of both closed and open loop based systems while eliminating their weak points. The improvements proposed is resistant to noise owing to digital FIR filter and data averaging techniques. The implementation used for proof of concept is implemented on a hardware using an FPGA and configurable to be used by any resolver.

Keywords: FPGA based RDC, Resolver to digital converter, RDC, feed-forward RDC, open-loop RDC

1. Introduction

Motion is a direct result of motors in systems such as planes, rocket or missile wings, seekers, vehicle wheels, and robotics. In motor controlled systems, the angular position feedback data is extremely important therefore the mechanical position of the controlled part should be sensed and expressed electrically by sensors. Depending on the task to be performed, the angular position data should be calculated with high accuracy even at high speeds and accelerations and needs to be robust. When the environmental conditions are harsh and the resolution is expected to be high, resolvers are the best choice instead of using encoders, hall sensors or other sensors. Therefore, resolver to digital converter (RDC) systems are needed to be built that drive the resolver, receive the resolver outputs and obtain the position information accurately.

Resolvers are one of the few kinds of angular position feedback devices such as; encoders and hall sensors. Resolvers are resistant to harsh environmental conditions like vibration, temperature, and moisture. They are known to be rugged and robust over extended usage periods. A resolver is basically a rotating transformer which is generally located on the motor shaft (some types have their own shaft and it is fastened to the rotor of the motor). Resolvers have one input (reference signal) and two output signals. As the rotor of the motor rotates, the rotor of the resolver rotates as well hence the mechanical position of the winding which the reference signal applied changes. This can be thought as the primary winding of a transformer. Compared to a standard transformer, a resolver has two secondary windings which stand at 90 degrees apart from each other and are not attached to the primary winding part. As the rotor rotates; since the stator of the resolver is stationary, the coupling ratio of the secondary windings changes according to the angle of the rotor. These induced voltages across the secondary windings are the outputs of the resolver. According to the position of the rotor and the transforming ratio of the resolver (a parameter specific to the resolver), the magnitude of the reference signal is transferred to the secondary windings with different ratios due to their 90 degrees apart position. Therefore the magnitudes of the outputs carry the absolute angular position information about the rotor position within 360 degrees.

A schematic cross-section of a resolver is shown in Fig. 1 and resolver input-output signals related to angular position is given in Fig. 2.

Reference signal and outputs can be expressed as:

$$V_R = A \times \sin(\omega_t t) \quad (1)$$

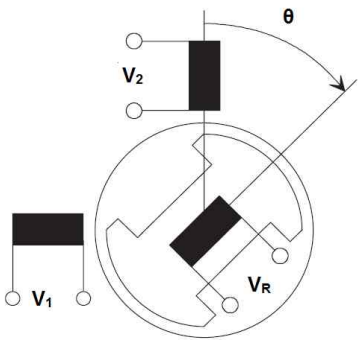

Fig. 1. Cross-section of a resolver [23]

† Corresponding Author: Hacettepe University, Ankara, 06800, Turkey. (alkar@hacettepe.edu.tr).
* TUBITAK SAGE, Ankara, Turkey (anilkarabeyli@gmail.com).
Received: April 10, 2017; Accepted: August, 23 2017

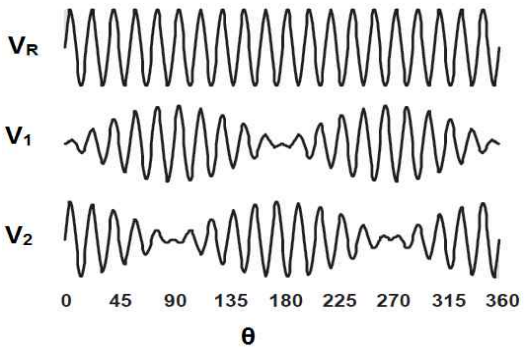

Fig. 2. Resolver signals [22]

$$V_1 = TR \times V_R \times (\sin(\theta) + \omega_r / \omega_t \times \sin(\omega_t t) \times \cos(\theta)) \quad (2)$$

$$V_2 = TR \times V_R \times (\cos(\theta) - \omega_r / \omega_t \times \sin(\omega_t t) \times \sin(\theta)) \quad (3)$$

Since ω_r is the mechanical speed of the shaft and normally 1000 times smaller than ω_t (angular frequency of the resolver excitation signal) even for very fast motors around 20000 rpm, V_1 and V_2 can be approximated [11] as shown below.

$$V_1 = TR \times A \times \sin(\omega_t t) \times \sin(\theta) \quad (4)$$

$$V_2 = TR \times A \times \sin(\omega_t t) \times \cos(\theta) \quad (5)$$

where V_R is the reference signal, V_1 is the sine output, V_2 is the cosine output, A is the amplitude, TR is the transforming ratio of the resolver and θ is defined as the angle of the rotor.

Ideal reference signal amplitude and frequency and the transforming ratio change according to the resolver but the mathematical approach would be the same. Since the output is analog, the resolution of the resolver is theoretically infinite but in practice the methodology used in the system would be the determining factor in the resolution.

In the literature, studies have been done on resolver to digital conversion to obtain the angular position, both in the analog and the digital domains [1]. There are two main methodologies, widely used in the RDC designs: closed-loop and feed-forward systems. Closed-loop systems generally use the ATO (Angle Tracking Observer) algorithm and the PLL's (Phase Locked Loop) with a few modifications [2-14]. These basically compare the calculated angular position with respect to the new position and always try to compensate the difference between them. In closed loop-systems, the calculated angular position changes smoothly and systems track the shaft angle fairly good except at acceleration periods. However, due to the nature of the closed loop systems, there is a settling time to obtain the right angle information accurately, generally in the order of more than a few milliseconds. Also they have relatively long starting times with respect to the starting shaft angle position and instability risk at high speeds and accelerations.

Another main resolver to digital conversion methodology is a feed-forward system which is based on comparing the read signals from the resolver and calculating the arctangent value [15-29]. Designs proposed in [15-22] are mainly analog systems. In [18], resolver signals are sampled at points which are more likely to be linear rather than the peak points. Other designs sample the resolver outputs at peak points and propose techniques for low cost and simple designs. These feed forward analog based designs are not suitable for high speed systems and do not have high accuracy. Also results are more dependent on temperature and the tolerances of the components used.

Designs proposed in [23-29] are heavily digital, feed-forward systems. One of the first designs among digital systems is [23], which is a milestone work on feed-forward systems. The design analogously generates and applies the excitation signal to the resolver; obtains, amplifies, and rectifies the output of the resolver signals synchronously along with the excitation signal to remove the carrier. Subsequently, the signal is applied a low pass filter and converted to digital to be processed by a microprocessor. In the microprocessor, the arctangent value is obtained by a look up table (LUT). Other designs are mainly similar to the work done in [23] with some improvements. In [23], a digital FIR band pass filter is used to increase the resolution along with a PI controller, an IIR filter, and an integrator to compensate the phase lag. In the design, the angular position is obtained by an open loop system but the phase lag compensation part is implemented as closed loop. It has a 14 bits resolution and a maximum of 40ms settling time. In [26] the resolver output signals are sampled not only at peak points of the excitation signal but 12 times in one period. In addition, a method is proposed to decrease the usage of the LUT memory. In work by [28], the emphasis is given on filtering and comparing the impacts of different types of digital filters on the signal quality and the resolution.

Feed-forward digital designs are stable, easy to modify, and robust against external factors. However, since they are LUT based, these need relatively more memory proportional to the required resolution in angle calculations. Also for both analog and digital based feed forward systems, to the best of our knowledge there is no work done to compensate the phase lags and sense the revolutions of the rotor and the proposed methods are not suitable for high speed and high acceleration systems.

The aim of this paper is to achieve a real time position data of a rotating motor from a resolver with maximum accuracy by combining the techniques that work well and to eliminate the weak points of both the closed loop and feed forward designs. We propose novel improvements for calculating the exact real time position using angle position estimations and error correction methods. In contrast to other approaches published in the literature, the proposed system is capable of sensing rotation at angle values beyond 360 degrees. The proposed system delivers the rotation count, the angular position, and the speed with high accuracy.

In the proposed system the FIR filter and averaging techniques are used which are known to be resistant against noise. We also propose methods such as data elimination, correction, and estimation which yield high performance in challenging conditions such as low signals, high noise, and fast rotations. In addition, the CORDIC (COordinate Rotation DIgital Computer) algorithm is implemented in the design to eliminate the use of LUT's, i.e. the additional memory which is known to be a bottleneck in portable systems.

In the next section, the general feed-forward RDC methodology and proposed improvements are described. In section 3 simulation results and performance is shown and in section 4 conclusions are presented.

2. Overview of Typical Rdc and The Proposed Improved Method

Typically a sine wave generator is inputted to a resolver. A sample and hold circuit latches and keeps the peak points of the resolver output signals which are then sampled with an ADC with the frequency of the resolver reference signal or at twice the frequency. Then by using an FPGA, microprocessor or microcontroller and a memory which keeps a Look-up-Table (LUT) for the arctangent conversion, the calculation of the positions are carried out. A block diagram of a general RDC architecture is shown in Fig. 3.

The current RDC functionality can be improved by incorporating a hardware that can detect and eliminate the problems commonly encountered. A general block diagram of the proposed system is shown in Fig. 4 and internals of the FPGA are detailed further in Fig. 5.

In the proposed design, the resolver reference signal is generated on an FPGA and after being converted by a DAC, this value is fed into a resolver. Outputs of the resolver are sampled and transmitted to the FPGA with an ADC. In the

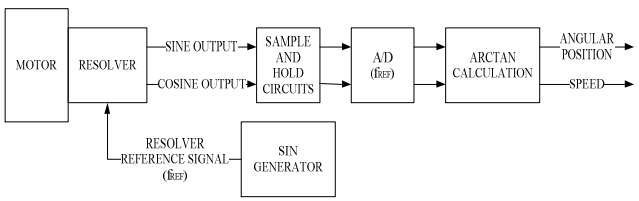

Fig. 3. Block diagram of the general RDC system

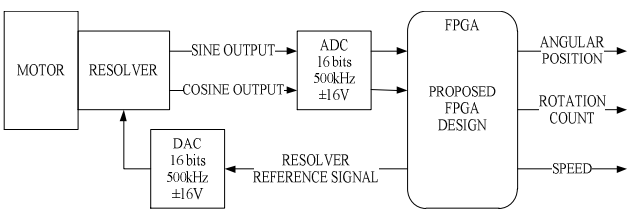

Fig. 4. Block diagram of the proposed RDC system

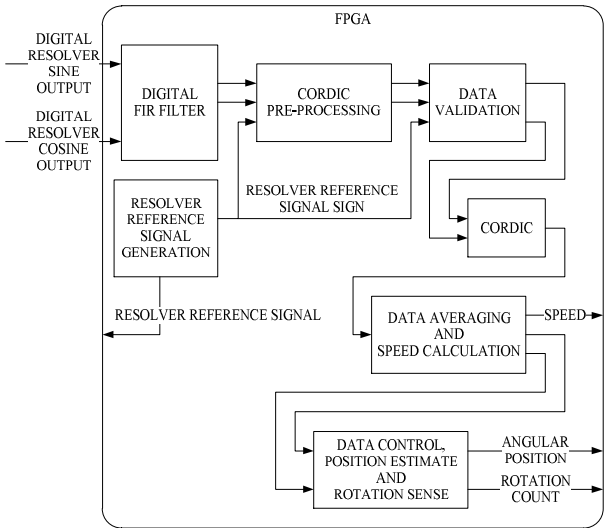

Fig. 5. Block diagram of the proposed FPGA design

FPGA signals are filtered with a digital low-pas FIR filter. Then a CORDIC pre-processing block is used to determine the quadrant of the signals by using reference signal sign. Data validation block aims to block the signals with low Signal to Noise Ratio (SNR) to prevent working with data that may be misleading. Then, a CORDIC algorithm is used to process the signals and calculate the position data. After the data averaging and speed calculation block, the average of the last 16 position is taken to increase the accuracy and to calculate the speed. In the next block, reliability of the position data is checked by an algorithm that utilizes the position and speed information. As a result of this, real time position estimation is made that compensates the calculation delay. A full rotation can also be detected by using the rotation sense algorithm.

Details of the blocks of the proposed RDC system and improvements according to general RCD method are detailed in further.

2.1 Resolver reference signal generation

Generally a sine wave is used as a reference signal in resolvers. The ideal working resolver reference signal frequency and amplitude varies according to the resolver used. Generally frequency of the sine wave is between 100Hz to 20 kHz and the amplitude is between 5V to 30V. A sine signal can be generated from an analog sine wave generator or it can be also generated in the FPGA from a LUT and a DAC. The input impedance of a resolver is typically between 50 Ohms to 1.5K Ohms therefore an amplifier may be required depending on the voltage and the current needs of the resolver and an analog low-pass filter may be used to smooth out the sine wave.

The system is designed and tested for a 5 kHz sine wave signal and the output of the resolver is assumed to be 16Vp-p but the design is compatible to be used with any frequency and voltage rating with some modifications. If

the signal is not generated by the FPGA then the sign of the reference signal has to be determined by a comparator and transmitted to the FPGA. For the proposed design, the sine wave is generated by the FPGA thus eliminating the use of an external analog signal generator and a comparator hence a more compact design is achieved.

2.2 FIR filter

In the proposed design a digital FIR filter is used to suppress noise. This filter reduces the peaks and other high frequency noises to transmit more clear signals which are critical for accuracy.

The response type of the used FIR filter is low pass. In the filter, the pass-band gain is set to be as low as possible since setting it at a higher value distorts the original signal, which produces higher angle errors especially while dealing with signals with small amplitudes. The pass-band gain in our system is set to be 0.001 dB maximum. The frequency of the resolver signals are around 5 kHz which jitters slightly according to the speed of the rotor so a cut off frequency of 10 kHz (where the gain falls below 0.001 dB) is set for the pass band frequency. Order of the filter directly affects the delay therefore other parameters of the filter are chosen to be low order and set to 14. The stop band is held at 60 dB. The resulting FIR filter is shown in Fig. 6. In order to measure the effect of the FIR on the performance, the filter block is removed from the system and the results are compared for a constant motor which

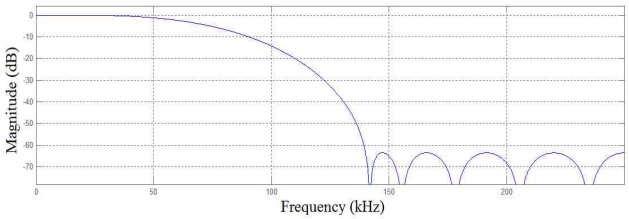

Fig. 6. FIR filter magnitude response

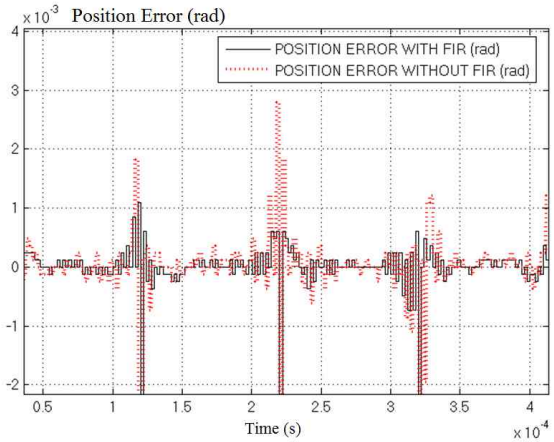

Fig. 7. Position error with and without the FIR filter for a constant motor

stays at π/90. The angular position error with and without FIR filter block is illustrated in Fig. 7.

As it is seen from Fig. 7 FIR block suppresses the peaks, provides more clear data to the next block and decreases the error of position data.

2.3 CORDIC preprocessing

Before transmitting the data to the CORDIC block a preprocessing is needed since the resolver output signals are correlated with the sign of the reference signal. When the reference signal is negative, both of the resolver output signals should be multiplied by -1. The multiplication should be delayed to match the arrival times of the read resolver signals for synchronization purposes to prevent error. Therefore the delays due to the DAC, the analog filter (if used), ADC and the FIR filter should be calculated.

CORDIC block accepts signed (2's compliment) fixed point data between -1 and 1. Therefore the signal should be scaled for this range. This scaling is performed by simply moving the position of the floating point so that it fits to the desired range without losing accuracy. The CORDIC preprocessing block produces an 18 bits data with a fraction point at 16th bit to the next block.

2.4 Data validation

In general method only the peak points of the resolver outputs are used. However in the proposed design there are 100 times more samples in a period to stay more close to real time and deal with noise. Therefore we do not only have signals with large amplitude but also the small ones. Signal to noise ratio changes according to the signal amplitude so to prevent calculations based on noise, signals with low SNR are eliminated to keep accuracy.

When dealing with very small signals, small errors in the CORDIC inputs result in large angle errors. Considering the formula (4) and (5); when the Vref (A×sin (ωt)) is very small, the signals in the range of 4μsec before and after the zero crossing of the reference signal are ignored and the CORDIC input is not updated. Consequently the block output is updated 92% of the whole period. The output that is not refreshed is only 8 μsec at maximum for a 5 kHz resolver reference signal. This is more than sufficient to keep track of a position in real time especially at high speeds and to keep the Signal to Noise Ratio (SNR) at an acceptable level.

After 18 μs of the reference signal zero crossing, which is equal to the DAC, the ADC, and the FIR filter total delay in the system, the input signals become extremely small and the noise predominates the signals hence the CORDIC cannot calculate the angle correctly. In the data validation block, at the time of the zero crossing of the reference signal, a counter is initiated. The block ignores the input values during 14 μs to 22 μs which corresponds to 4 μs before and after the zero crossings of the resolver output

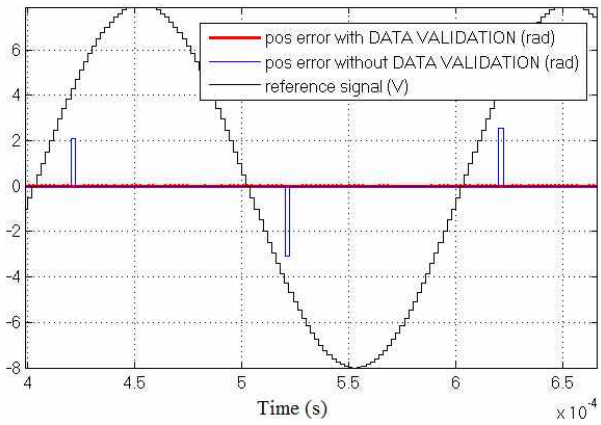

Fig. 8. Position error with and without data validation

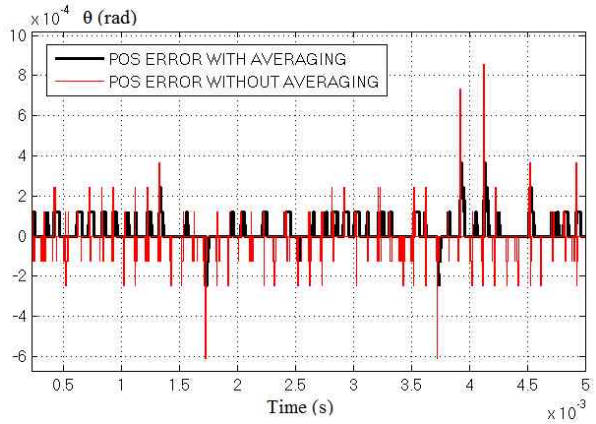

Fig. 9. Error with and without data averaging for a stationary motor at π/8

signals. In this way the CORDIC will ignore the small inputs and will not calculate angles for these values thus large errors at the output of the CORDIC are avoided. In Fig. 8, the impact of data validation block on the position error at the system output is shown. It is seen that 18 µs after the zero crossing of the reference signal, a large position error occurs when the data validation block is not used.

2.5 CORDIC

In order to translate the digitally transmitted voltage inputs to angle information the CORDIC algorithm is used. In general methodology a LUT is being used but in the proposed design CORDIC is preferred to reduce memory space needed.

The CORDIC module takes two inputs fitted between -1 to 1, implements the arctangent algorithm and calculates the angle in 18 clock cycles. The output is in radians between $-\pi$ to π. The input of the CORDIC block is 18 bits wide and the output is adjusted to be signed 16 bits, where the 13 bits are in floating point.

2.6 Data averaging and speed calculation

In general methodology position data is transmitted to the controller directly at this point. However in the proposed design to reduce noise affects, improve accuracy and calculate speed 'Data averaging and speed calculation' block is used.

In order to smooth out the short term fluctuations and eliminate the noise effects, a window based moving average method has been used. This method produces better position and speed data with respect to the methods that do not use averaging. The last 16 positions received from the CORDIC algorithm is averaged. In every 2 µs, the last 16 data are summed and shifted 4 bits to the right (divided by 16) to obtain an average. Then the window is shifted again one data position to obtain another 16 bit average. Position error with and without data averaging for a stationary motor at π/8 is shown in Fig. 9.

Table 1. Speed calculation errors for different speeds

Motor speed (rpm)	Expected speed (rad/38µs)	Maximum speed error (rad/38µs)	Maximum speed error (degree/38µs)
300	0.0012	0.0003	0.017
1000	0.00398	0.0005	0.029
10000	0.03977	0.002	0.115

From Fig. 9 we can observe the improvement of averaging on position error.

In order to calculate the speed, the last 19 averaged data is stored. The difference between the first and the last data corresponds to the position change in 38 µs which is equal to the total system delay of the proposed system and will be used in position estimation. To increase reliability, the speed estimation results are bounded by upper and lower values. Approximately, speeds below 150rpm and above 12000rpm are ignored. For speeds up to 150 rpm, the position change below 0.0006 radians in 38 µs (system calculation delay) is considered as acceptable. The upper limit should be chosen according to the maximum speed of the motor. This limitation is crucial in order not to be effected from misleading CORDIC outputs. CORDIC outputs can be very deceptive when the position is close to 0, π/2, 3*π/2, π (rotation point) and where the reference signal is extremely small since the SNR is extremely low.

For a 10000 rpm rotating motor, position change in 38µs is 0.0398 rad. As seen from Fig. 10, the error of the speed calculation is below 5%. This corresponds to a maximum of speed calculation error below 0.002 rad/38µs for a 10000 rpm rotating rotor and improves as the rotation speed is decreased. Table 1 shows the errors for some different speed values.

2.7 Data control, rotation sense and position estimation

In this block of design the position data is verified for accuracy. The received position data is compared with

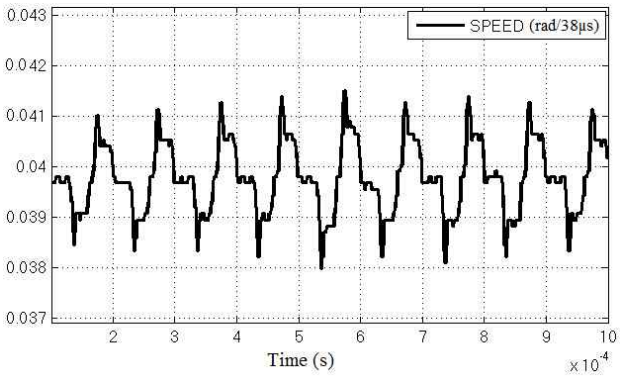

Fig. 10. Speed (rad/38μs) for a 10000 rpm rotating motor

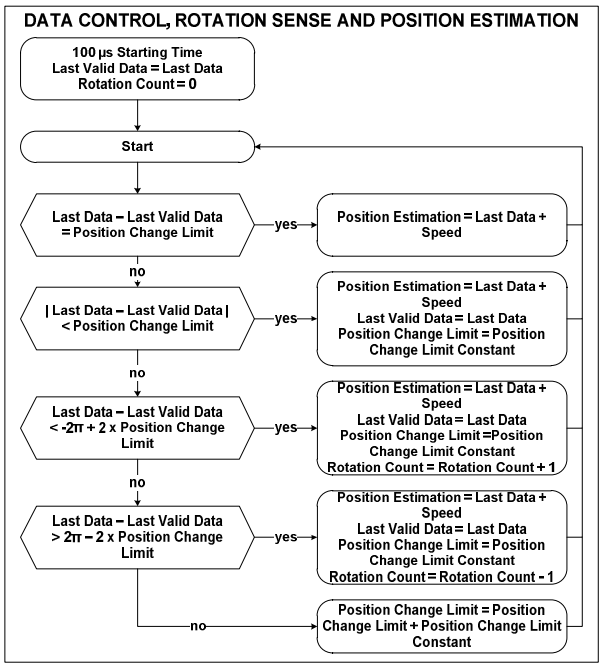

Fig. 11. Data control, rotation sense and position estimator block algorithm

respect to the previously obtained position information and if the difference is zero or within acceptable limits, data is considered as valid then the speed information is used to estimate the real time position to compensate the calculation delay. Acceptable position change limits are calculated according to the maximum speed of the motor. For a 10000 rpm motor, maximum position change in 2 μs is 0.0022 radians. To stay safe 0.003 radians position change in every 2 μs is used as acceptable position change limit. The real time position is estimated by adding the speed data (rad/38μs) to the last position information. If the difference is beyond acceptable limits then it will be ignored or this may be considered as a full rotation of the motor. Fig. 11 illustrates the algorithm explained above and the algorithm is explained in detail below.

According to the speed of the motor, the delay related error occurs at the output of the CORDIC. Although the

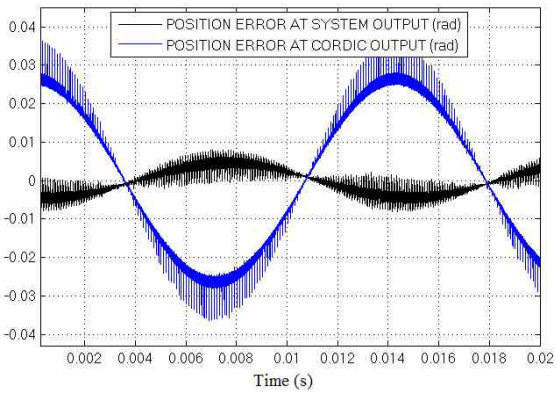

Fig. 12. Position error for a 70Hz ±π sine position input at the CORDIC and the system output

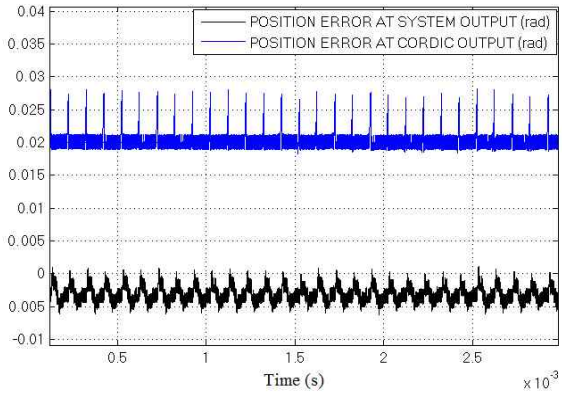

Fig. 13. Position error for a 10000 rpm rotating motor at the CORDIC and the system output

delay calculation is only around 18μs at the CORDIC output, it may cause errors up to 0.035 radians (2 degrees). With the suggested data averaging and position estimation methodologies applied, the error is decreased to 0.008 radians (0.46 degrees) or less. The proposed methods compensate the fluctuations caused by low SNR and phase lags due to the calculation delays and yield better results. For a 70 Hz -/+π sine position input and for a 10000 rpm motor, errors at the CORDIC and the system output are shown in Fig. 12 and Fig. 13 respectively.

If the difference between the last and the previous data is beyond allowable limits, the final data may be distorted and should not be accepted as a trustable information or a full rotation may have occurred. The high sampling rate used in the system allows the full rotation of the motor to be sensed. The amount of rotation is not limited to only π to $-\pi$ and the proposed algorithm supports multiple revolutions of the rotor. The last valid data and last sampled data are compared and if the change is approximately 2*π radians, it means that a rotation has occurred. When this difference in the position is more than the allowed limit, the new position is accepted and the rotation count is updated. Rotation can be sensed both when it occurs clockwise or counter-clockwise. This

feature brings a lot of benefits in applications where multiple rounds of rotations are possible in a motor. To the best of our knowledge, this has not been addressed for any closed loop system in the literature before. In Fig. 14 it can be seen that the estimated position (position out) is between $-\pi$ to π. However, by sensing the rotation and keeping the rotation count, angular positions greater than π can be tracked.

System is designed, optimized, and tested for 16 bit position output resolution and 5 kHz, 16 V_{p-p} reference signal but with minor changes, the design can be adapted to any resolver. By implementing these algorithms the design achieves fairly low position errors depending on the speed. System has a high tracking rate, gives fast responses to acceleration, and step changes. Also it is capable of sensing the revolution and keeping the revolution count. Some numerical properties representing the performance of the proposed design is given in Table 2.

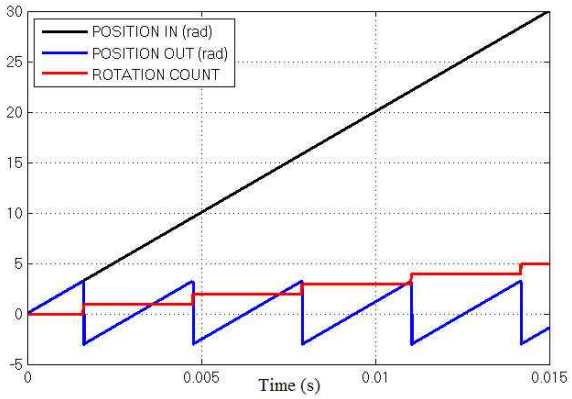

Fig. 14. Position input, estimated position output and rotation count for a motor rotating at 20000 rpm

Table 2. Properties of the proposed method

	[7]	[8]	Proposed Design
Resolution (bit)	16	16	16
Reference Signal Frequency (kHz)	2 - 20	0.05 - 20	Adaptable (5 is used)
Reference Signal Amplitude (Vp-p)	12.8 - 16	2 - 16	Adaptable (16 is used)
Input Signal Amplitutes (Vp-p)	2.3 - 4	5 - 6	Adaptable (8 is used)
Angular Accuracy - constant position (arcmin)	5	2	1
Angular Accuracy - 1-1000 rpm constant speed (arcmin)	5	2	1.5
Angular Accuracy - 1000-2000 rpm constant speed (arcmin)	5	Fail	2.75
Angular Accuracy - 2000-3500 rpm constant speed (arcmin)	5	Fail	5
Angular Accuracy - 3500-9375 rpm constant speed (arcmin)	5	Fail	12
Angular Accuracy - 9375-20000 rpm constant speed (arcmin)	Fail	Fail	27
Maximum Tracking Rate (rpm)	9375	975	50000
Acceleration Error with 125 rps² (arcmin)	<30	N/A	<10
179 Degrees Step Input Settling Time (ms)	45	12	0.37
Revolution Sense	No	Yes	Yes

In [7] and [8], two of the most accurate designs popularly used in industry and in military systems are described. Compared to these, the proposed method is more flexible in terms of resolution, reference signal frequency, reference signal amplitude, and sin-cos input signals amplitudes. The accuracy is better for constant position and below 3500 rpm speed. Also it is comparable at other speeds. The maximum tracking rate, acceleration error, and settling times are also superior. In addition the design is capable of driving more than one resolver thus it is very advantageous in cost and space.

2.8 Hardware cost

The total RDC design approximately consumes 1800 flip-flops, 1500 LUTs, 75 Memory LUTs, and 2 DSP48 blocks. Utilization consumes less than 1% of a Kintex 7 series FPGA (XC7K160T).

3. Simulation Results and Performance

To see the performance of the design, system is tested with the setup as shown in Fig. 15 by using MATLAB System Generator tool. Resolver ADC, DAC and FPGA model of the proposed design is created and resolver implementation is fed by artificial motor position data. Outputs of the FPGA model are saved in a file and graphs are drawn in MATLAB. Detailed information about resolver, ADC, DAC and FPGA model implementations, performance tests and comments on results are given below.

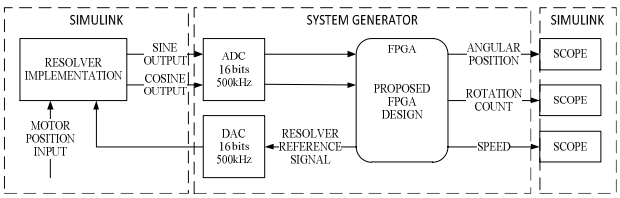

Fig. 15. Simulation block diagram

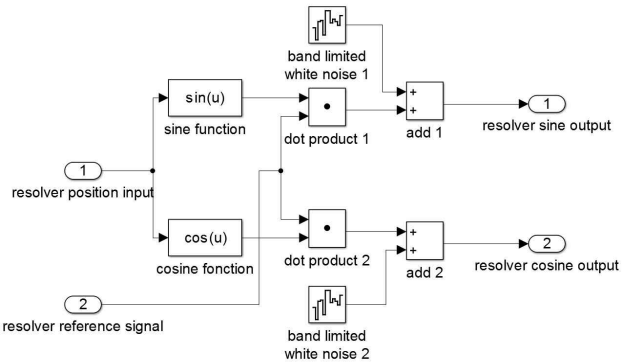

Fig. 16. The resolver mathematical model

3.1 Resolver implementation

The resolver implementation block shown in Fig. 15 is the implementation of the mathematical expressions given in (3), (4), and (5). The details of the resolver implementation are shown in Fig. 16. Rotor position data is an input to the simulation tool to be processed by the resolver. The output of the resolver is sampled at 500 kHz with 16 bits resolution. The converted result is between -16V to 16V so the Least Significant Bit (LSB) voltage corresponds to 480μV. In order to generate more realistic results, a $2mV_{p-p}$ white noise signal is added on both channels.

3.2 ADC and DAC implementation

Resolver implementation outputs between -16 and 16 are sampled with 500 kHz and expressed with 16 bits to feed the FPGA model. Resolver reference signal created by FPGA model is converted to continuous data and resolver is fed with the signal.

4. Conclusion

In this paper an improved design of feed forward resolver to digital conversion methodology has been proposed. The proposed design adds up the strong features of both closed and open loop based systems in the literature and eliminates their weak points. The design is compatible with different kinds of resolvers. It is resistant to noise owing to digital FIR filter and data averaging techniques. Since the design uses CORDIC algorithm instead of a LUT based approach, the memory requirement is extremely low. With the suggested position estimation methodology, calculation delay is compensated, and the error is substantially decreased. The proposed design also does not have long start up time and is highly stable. It is optimized for fast real time tracking and short response delay. In addition, the design includes a novel data sensing and rotation estimation algorithm implemented in hardware which is capable of detecting a rotation and returning the position data with high resolution and accuracy. In addition, the proposed design can be used with multiple resolvers simultaneously such as systems which control several motors and hence provides cost and area advantage. The proposed algorithm is implemented on an FPGA and tested with system generator tool.

References

[1] C. Mohan, R. Sivappagari, and N. R. Konduru, "Review of RDC Soft Computing Techniques for Accurate Measurement of Resolver Rotor Angle," *Sensors & Transducers*, vol. 150, no. 3, pp. 1-11, 2013.

[2] C. H. Yim, I. J. Ha, and M. S. Ko, "A resolver-to-digital conversion method for fast tracking," *IEEE Trans. Ind. Electron.*, vol. 39, no. 5, pp. 369-378, 1992.

[3] A. O. Di Tommaso, and R. Miceli, "A new high accuracy software based resolver-to-digital converter," in *IECON Proceedings (Industrial Electronics Conference)*, 2003, vol. 3, pp. 2435-2440.

[4] R. Hoseinnezhad, and P. Harding, "A novel hybrid angle tracking observer for resolver to digital conversion," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*, 2005, vol. 2005, pp. 7020-7025.

[5] J. Onno, K. Heiko, and S. Marcel, "FPGA Based Resolver to Digital Converter Using Delta-Sigma Technology," *Delta*, pp. 931-936, 2006.

[6] K. Bouallaga, L. Idkhajine, A. Prata, and E. Monmasson, "Demodulation methods on fully FPGA-based system for resolver signals treatment," in *2007 European Conference on Power Electronics and Applications, EPE*, no. 1, 2007.

[7] Analog Devices, "Variable Resolution, 10-Bit to 16-Bit R/D Converter with Reference Oscillator," AD2S1210 datasheet, 2010.

[8] Analog Devices, "Variable Resolution, Monolithic Resolver-to-Digital Converter," AD2S80A datasheet, 2014.

[9] L. Idkhajine, a. Prata, E. Monmasson, K. Bouallaga, and M. W. Naouar, "System on Chip controller for electrical actuator," *IEEE Int. Symp. Ind. Electron.*, pp. 2481-2486, 2008.

[10] L. Ben-Brahim, and M. Benammar, "A new PLL method for resolvers," in *2010 International Power Electronics Conference ECCE Asia, IPEC 2010*, no. 4, pp. 299-305, 2010.

[11] J. Bergas-jané, S. Member, C. Ferrater-simón, G. Gross, R. Ramírez-pisco, S. Galceran-arellano, and J. Rull-duran, "High-Accuracy All-Digital Resolver-to-Digital Conversion," *IEEE Trans. Ind. Electron.*, vol. 59, no. 1, pp. 326-333, 2012.

[12] D. A. Khaburi, "Software-based resolver-to-digital converter for DSP-based drives using an improved angle-tracking observer," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 4, pp. 922-929, 2012.

[13] N. Noori, and D. A. Khaburi, "A new software-based method for rotor angle calculation," in *the 5th Power Electronics, Drive Systems and Technologies Conf.,* Tehran, pp. 305-310, 2014.

[14] Y.-H. Kim, and S. Kim, "Software resolver-to-digital converter for compensation of amplitude imbalances using D-Q transformation," *J Electr Eng Technol* Vol. 8, No. 6, pp. 1310-1319, 2013.

[15] M. Benammar, L. Ben-Brahim, and M. A. Alhamadi, "A novel resolver-to-360° linearized converter,"

IEEE Sens. J., vol. 4, no. 1, pp. 96-101, 2004.
[16] M. Benammar, L. Ben-Brahim, and M. A. Alhamadi, "A high precision resolver-to-DC converter," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 6, pp. 2289-2296, 2005.
[17] M. Benammar, L. Ben-Brahim, M. A. Alhamadi, and M. El-Naimi, "A Novel Converter for Sinusoidal Encoders," *in IEEE Sensors Conference*, pp. 1415-1418, 2006.
[18] L. Ben-Brahim, M. Benammar, M. A. Alhamadi, N. A. Al-Emadi, and M. A. Al-Hitmi, "A new low cost linear resolver converter," *IEEE Sens. J.*, vol. 8, no. 10, pp. 1620-1627, 2008.
[19] A. Kaewpoonsuk, W. Petchmaneelumka, A. Rerkratn, S. Tammaruckwattana, and V. Riewruja, "A novel resolver-to-DC converter based on OTA-based inverse-sine function circuit," in *Proceedings of the SICE Annual Conference*, pp. 609-614, 2008.
[20] L. Ben-Brahim, M. Benammar, and M. a. Alhamadi, "A resolver angle estimator based on its excitation signal," *IEEE Trans. Ind. Electron.*, vol. 56, no. 2, pp. 574-580, 2009.
[21] M. Benammar, M. Bagher, and M. Al Kaisi, "Novel linearizer for tangent/cotangent converter," in *2009 16th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2009*, pp. 575-578, 2009.
[22] H. Schmid, "Direct resolver to digital converter," U.S. Patent 5 912 638, June 15, 1999.
[23] Texas Instruments, Appl. Report SPRA605, pp. 1-23.
[24] C. Attaianese, and G. Tomasso, "Position measurement in industrial drives by means of low-cost resolver-to-digital converter," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 6, pp. 2155-2159, 2007.
[25] S. Sarma, V. K. Agrawal, and S. Udupa, "Software-based resolver-to-digital conversion using a DSP," *IEEE Trans. Ind. Electron.*, vol. 55, no. 1, pp. 371-379, 2008.
[26] Y. Zhu, J. M. Wang, and M. Zhu, "An approach based on AD converted resolver demodulation," in *ICACTE 2010 - 2010 3rd International Conference on Advanced Computer Theory and Engineering, Proceedings*, vol. 5, no. 1, pp. 192-195, 2010.
[27] Z. Ming, W. Jianming, D. Ling, Z. Yi, D. Ruzhen, and L. Yu, "A software based robust resolver-to-digital conversion method in designed in frequency domain," in *Proceedings - 2011 International Symposium on Computer Science and Society, ISCCS 2011*, pp. 244-247, 2011.
[28] A. K. S. Baasch, E. C. Lemos, F. Stein, A. S. Paterno, J. De Oliveira, and A. Nied, "Resolver-to-digital conversion implementation - A filter approach to PMSM position measurement," *in International Conference on Power Engineering, Energy and Electrical Drives*, no. May 2011.
[29] S. C. M. Reddy, and K. N. Raju, "Inverse tangent based resolver to digital converter - a software approach," *International Journal of Advances in Engineering & Technology,* vol. 4, no. 2, pp. 228-235, 2012.

Fikret Anıl Karabeyli received his B.S. degree in 2011 at Electrical and Electronics Engineering Department, Bilkent University and M.S. degree in 2015 at Electrical and Electronics Engineering Department, Hacettepe University, Ankara, Turkey. He has worked in TUBITAK SAGE. Mr. Karabeyli has research and experience in military systems, electronic card design, motor drive systems and FPGA coding.

Ali Ziya Alkar (M'93) received his MSc. and Ph.D. degrees in Electrical and Computer Engineering from University of Colorado at Boulder, U.S.A. in 1991 and 1995 respectively. He is currently a full time Professor in Department of Electrical and Electronics Engineering at Hacettepe University, Ankara, Turkey. Dr. Alkar has worked as a consultant for the government as well as the industry. He has supervised and been a part of government and internationally funded projects. He also has guided several graduate theses during his academic studies at the university. His research interest includes computer architecture and design, security issues, instrumentation, image processing, networking and solving embedded system requirements on various platforms. Dr. Alkar's awards and honors include the best thesis award from the Technology Development Foundation of Turkey and Ihsan Dogramaci High Achievement Award. He is also a member of the IEEE Consumer Electronics Society.