Algorithm for real-time estimation of absolute angular position

Veljko Mihajlović

***Abstract* — This paper presents a new algorithm for the absolute angular position estimation in rotary applications, with real-time correction included. The algorithm is intended to be used in combination with sensors whose output is a sine/cosine function of an angle of rotation, which is the case with majority of sensors in rotary applications: Hall-effect sensors, resolver-type sensors, rotary capacitive/inductive sensors etc. Demanded hardware resources are minimized. On the other hand, the algorithm successfully utilizes increase in available hardware resources: additional memory locations, hardware multipliers, parallelism. Thus, it can be purposefully implemented on a wide range of the platforms, from the simplest ASIC calculation blocks, through all commercial microcontrollers and microprocessors, up to the FPGA chips and DSPs.**

***Keywords* — angular position estimation, limited hardware resources, real-time correction algorithm.**

I. Introduction

In rotary applications determination of the angular position is an inevitable issue. Nowadays, systems in charge of that process generally consist of two sub-systems: sensor part and signal processing part [1]. The quality of both sub-systems influences final accuracy. This paper contains a new model of the algorithm that is implemented in the signal processing part. The complete physical surrounding environment, where the algorithm would be implemented, is simulated in order to provide enough freedom for its characterization.

II. Calibration procedure

A. Measurement setup

The generalized measurement setup consists of four sensors that are placed at the angular displacements of π/2 between each, on the measured rotating object.

The calibration procedure is done in the laboratory conditions, which allows two setups: the calibrated measurement setup and the referent measurement setup [1]. The referent measurement setup is verified and its accuracy and precision are well determined.

The referent measurement setup gives the output that is comprised of four variables with sinusoidal behavior. Since the accuracy and the precision of the measurements from the referent measurement setup are known, these curves are considered to be ideal referring to the calibrated measurement system, and can be described as:

$$\begin{aligned} c_1^r &= \sin(\theta) \\ c_2^r &= \sin(\theta + \pi/2) = \cos(\theta) \\ c_3^r &= \sin(\theta + \pi) = -\sin(\theta) \\ c_4^r &= \sin(\theta + 3\pi/2) = -\cos(\theta) \end{aligned} \quad (1)$$

On the other hand, the calibrated measurement system suffers from errors, which can be classified in two groups: a systematic error e_s and a random error e_r [2]. In practice, the systematic error of the measurement originates from the production imperfectness and the systematic error of the measurement instrument. The random error includes various causes. For the purpose of analyzing the calibration procedure the random error is split in two categories: measurement imprecision e_{rm} and a temporal variation in the sensor behaviour, e_{rv}.

The analytical description of the taken measurements that come from the calibrated measurement system, and which are the inputs into the calibration procedure, is given as:

$$\begin{aligned} c_1^m &= A_1 \sin(\theta) + e_{1s} + e_{1r} \\ c_2^m &= A_2 \cos(\theta) + e_{2s} + e_{2r} \\ c_3^m &= -A_3 \sin(\theta) + e_{3s} + e_{3r} \\ c_4^m &= A_4 \cos(\theta) + e_{4s} + e_{4r} \end{aligned} \quad (2)$$

B. Calibration

The aim of the calibration is to measure the difference between the measured and the ideal inputs and process the difference so that the parameters for the correction and angle calculation can be determined accurately to a feasible extent.

The difference is to be fitted by a polynomial function. Since the correction algorithm is real-time, the arguments of the polynomial function can only be samples of the measurements.

The calibration procedure consists of the following four steps that are taken in order to preprocess measurements so that polynomial coefficients can be more accurately estimated and to obtain polynomial coefficients for the correction:

Pre-filtering. The redundancy of the simultaneous equations in (2) is used for removing the temporal variation of the sensor's behaviour (e_{rv}). The temporal variation of the sensor's behaviour is equal for the all curves in the equation (2). Thus, the first step is substituting opposite curves:

Veljko Mihajlović, TES Electronic Solutions d.o.o., Belgrade, Serbia (phone: 381-11-3370017; e-mail: veljko.mihajlovic@tesbv.com).

$$
\begin{aligned}
c_{13}^m &= c_1^m - c_3^m = A_{13}\sin(\theta) + e_{13s} + e_{13rm} \\
c_{24}^m &= c_2^m - c_4^m = A_{24}\cos(\theta) + e_{24s} + e_{24rm} \\
c_{31}^m &= c_3^m - c_1^m = -c_{13}^m \\
c_{42}^m &= c_4^m - c_2^m - c_{24}^m
\end{aligned} \tag{3}
$$

Scaling. It has been empirically proven in the research phase for this paper that the fitting polynomial can be more accurately determined if the offset and the amplitude of the measured curves from the Eq. (3) are roughly calculated and removed. The rough estimation of the offset is determined in the calibration phase and is simply calculated as a mean value of the curves given in (3).

The rough estimation of the amplitude is given by the formula:

$$
\hat{A} = \sqrt{\frac{2}{N_{cal}} \sum_{k=1}^{N_{cal}} [c^m(k) - \hat{O}]^2}
$$

where

N_{cal} – number of taken measurements in calibration (4)

$c^m(k)$ – kth value of c^m

$\hat{O}$ - roughly estimated offset

After this, we have scaled measurement curves, which are adapted for correction.

$$
\begin{aligned}
c_{13}^s &= c_{13}^m - \hat{O}_{13} - \hat{A}_{13} \\
c_{24}^s &= c_{24}^m - \hat{O}_{24} - \hat{A}_{24} \\
c_{31}^s &= c_{31}^m - \hat{O}_{31} - \hat{A}_{31} \\
c_{42}^s &= c_{42}^m - \hat{O}_{42} - \hat{A}_{42}
\end{aligned} \tag{5}
$$

Calculating domain of the fitting polynomials. The third step of preprocessing can be done by using the most linear segment as the argument of the fitting polynomial (x) in the particular region. This is allowed by the redundancy of the measurement system. The most linear region is determined by the algorithm in the Fig.1.

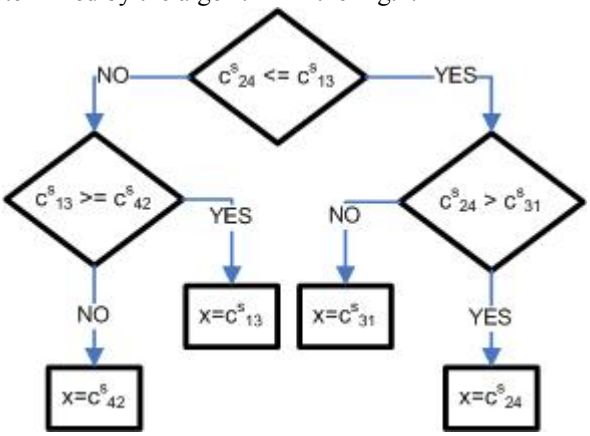

Fig. 1.The algorithm for selecting the argument of the fitting polynomial

Obtaining coefficients for the fitting polynomials. All curves from Eq. (5) are to be used as arguments of the fitting polynomial, depending on the angular position at which the fitting is done.

The position is to be determined by the *arctan* function and only two curves from the system (5) need to be corrected. The chosen curves in this implementation are c^s_{13} and c^s_{24}. Because of the transformation in the equation (3) it is obvious that choosing any other pair will give the same accuracy of the correction algorithm.

Finally, Eq. (6) gives the differences that are to be fitted and the corresponding fitting polynomials. The argument x is used from a curve that is chosen by the algorithm depicted in Fig. 1. Accordingly, coefficients a and b are chosen, depending on the region (i represents the region of correction and takes values 1..4); n represent a degree of the fitting polynomial.

$$
\begin{aligned}
d_1^s &= c_{13}^s - c_1^r \approx a_{(n-1)i}x^{n-1} + a_{(n-2)i}x^{n-2} .. + a_{0i} \\
d_2^s &= c_{24}^s - c_2^r \approx b_{(n-1)i}x^{n-1} + b_{(n-2)i}x^{n-2} .. + b_{0i}
\end{aligned} \tag{6}
$$

III. ALGORITHM FLOW

After the measurement setup is calibrated in the laboratory conditions, it is used for the real-time angular position estimation. The algorithm for position estimation basically contains two parts: *correction* and *angle calculation*. The complete flow of the algorithm is depicted in Fig 2.

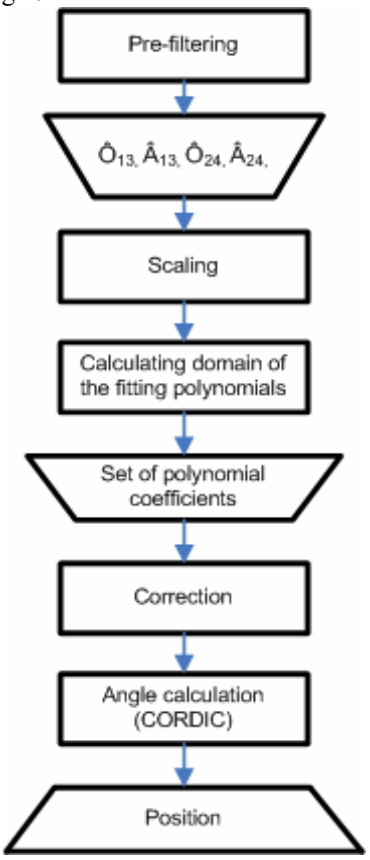

Fig. 2.Complete flow of the algorithm

The first three steps of the algorithm are done in the same way as in the calibration procedure. These steps are: *pre-filtering, scaling* and *calculating domain for the fitting polynomials*. The domain of the fitting polynomials determines which set of the polynomials is to be used. According to the domain, the corresponding coefficients are loaded from a non-versatile memory and the *correction* procedure is done. After correction, the *angle calculation* is done. The angle is calculated by the CORDIC algorithm [3, 4].

IV. HARDWARE REQUIREMENTS

The main advantage of the algorithm is the minimum required hardware resources for its implementation.

A. Calculation requirements

The algorithm can be executed with only two mathematical/logical operations: addition and bit shifting. Extreme simplicity makes it suitable for a very simple hardware where the power consumption and occupied space on a die (chip) are inevitable factors.

On the other hand, the algorithm is not strictly predicted for simple hardware. It is adaptable and can take advantage of additional hardware performances.

First, the hardware multiplier improves the speed of the algorithm in the correction phase. Second, the algorithm takes advantage of the hardware with parallelism in instructions executions (e.g. FPGA chips), which allows significant improvement of the speed in both correction and angle estimation phases.

B. Memory requirements

The memory resources could also be minimized in a case of the limitations. Provided additional memory locations the algorithm could easily utilize them in two ways. First, by increasing a degree of the fitting polynomial, which improves performances of the correction. Second, by increasing a number of CORDIC iterations, which directly improves accuracy of *arctan* calculations. These two ways are not independent and the correction and angle calculation phases need to be improved simultaneously.

Table 1 summarizes overall memory resources for the algorithm. The required memory resources for scaling are four locations for placing roughly estimated mean values and amplitudes. Polynomial coefficients take locations according to theirs degree (n). Number of locations occupied by the CORDIC algorithm is equal to the number of iteration in the CORDIC.

TABLE 1: MEMORY RESOURCES OF THE ALGORITHM.

Resource	Number of memory locations
Scaling	4
Polynomial coeffs (deg = n)	8*(n+1)
CORDIC coeffs (k iterations)	k
TOTAL	**8*(n+1)+k+4**

V. RESULTS

A. Testbench description

The inputs into the Algorithm are mathematically described by Eq. (2).

The systematic error e_s is a periodical function of the angular position, and can be described as a Fourier series [5], with the angular position θ as an argument:

$$e_s(\theta) = C_0 + \sum_{k=1}^{\infty} \left[AF_k \cos(k\theta) + BF_k \sin(k\theta) \right] \qquad (7)$$

The random error e_r has Gaussian distribution, with standard deviation σ.

B. Correction example

In this example measurements are generated with the following values (referring to the Eqs. (2) and (7)):

The measurement amplitudes are $A_1=A_2=A_3=A_4=1$. These amplitudes are chosen equal without loss of generality, because the Algorithm always completely suppresses the amplitude differences.

The systematic error is represented as a Fourier series with five terms, i.e. in the Eq. (7) k takes natural number values from the set [1,5]. The randomly chosen values are given in the Table2.

TABLE 2: PARAMETERS OF THE SYSTEMATIC ERROR.

	AF_1 BF_1	AF_2 BF_2	AF_3 BF_3	AF_4 BF_4	AF_5 BF_5
e_{1s}	0.11	0.20	0.13	0.00	0.03
	0.10	0.11	0.13	0.06	0.03
e_{2s}	0.10	0.19	0.07	0.09	0.03
	0.10	0.19	0.13	0.04	0.04
e_{3s}	0.50	0.10	0.17	0.04	0.00
	0.13	0.17	0.00	0.07	0.01
e_{4s}	0.13	0.00	0.10	0.01	0.01
	0.00	0.13	0.10	0.03	0.01

The random error e_r has the following parameters: σ=0.05. Standard deviation is 5% of the measurement amplitudes.

The generated measurements are drawn in Fig. 3, where x axis represents the angular position of the system, while c1M..c4M represent the curves from Eq.(2), respectively. They are given in relative units. Even without performing any quantitative analysis, it is obvious the measurements are significantly deformed and useless for direct estimating the position.

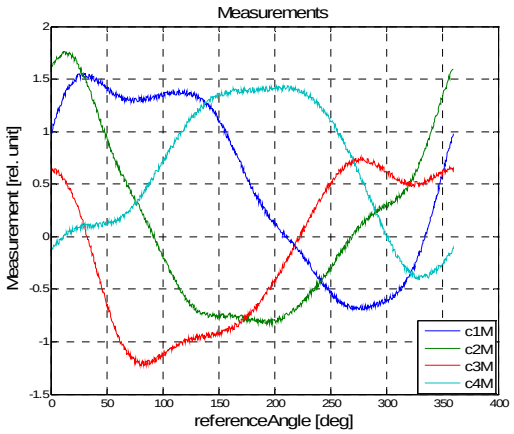

Fig. 3. Simulation of the measurements

In this case the correction is done with third degree polynomials.

The corrected curves are depicted in Fig. 4, where c1c..c4c represent corrected representation of the c1M..c4M, respectively.

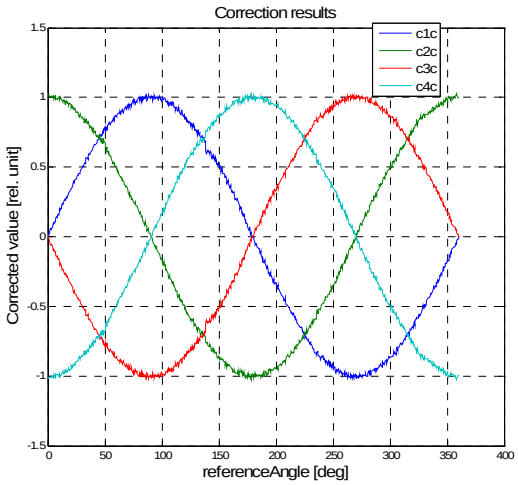

Fig. 4. Example of corrected curves, third degree correcting polynomial

After correction, the angle is calculated by using CORDIC algorithm. In this case, the number of iterations in CORDIC is 16. Comparison of the error in the case of not using the algorithm with the case when the algorithm is used is depicted in Fig. 5. The error is given in degrees.

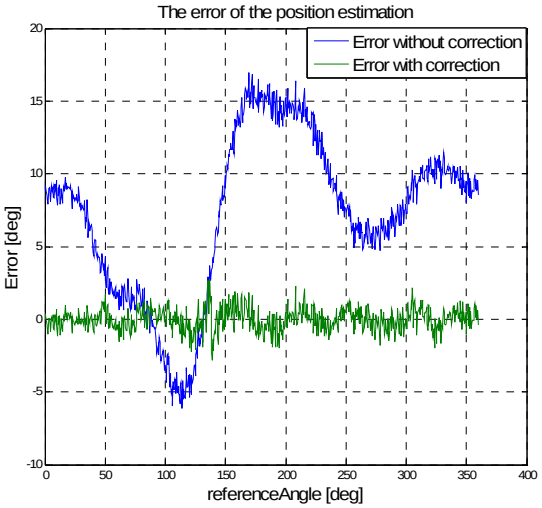

Fig. 5. Error of the position estimation, 16-iteration CORDIC

Table 3 gives comparison of some parameters that characterize the angular position estimation error. RMS(e) stands for the root mean square error on the one-period interval, Max (ABS(e)) is the absolute maximum of the error, Mean(e) is the mean value of the error. All values are rounded to two decimal places, thus the mean value in the case with correction is zero.

TABLE 3: CHARACTERISTIC OF THE ERROR IN A PARTICULAR IMPLEMENTATION.

	RMS(e)	**Max(ABS(e))**	**Mean(e)**
Error without correction [deg]	9.01	16.7	7.18
Error with correction [deg]	0.76	3.21	0.00

Table 4 gives the occupied memory resources for a particular implementation of the algorithm. It can be seen that only 52 locations are enough to support the algorithm. The simplest hardware platform on which this implementation can be executed consists of an accumulator, two temporal registers, adder, a shifter and 52 non-volatile register for storing parameters [6].

TABLE 4: MEMORY RESOURCES FOR A PARTICULAR IMPLEMENTATION.

Resource	**Number of memory locations**
Scaling	4
Polynomial coeff. (deg = 3)	32
CORDIC coeff. (16 iterations)	16
TOTAL	**52**

VI. NEXT STEPS

The current version of the algorithm is to be further characterized and accordingly improved. Moreover, the algorithm is planned to be implemented across various hardware platforms in order to verify its behaviour and find an optimum hardware for a particular desired performances. Also, the inputs from various types of sensors are going to be analyzed in order to adjust the correction procedure to some specific cases of the measurement error behaviour.

REFERENCES

[1] J.Webster (Editor-in-Chief), "The Measurement, Instrumentation and Sensors Handbook", CRC Press LLC & IEEE Press, 1999.
[2] S. Tumanski, "Principles of Electrical Measurements", Taylor & Fransis Group LLC, 2006.
[3] Ray Andraka, "A survey of CORDIC Algorithms for FPGA based computers", White paper, Andraka Consulting Group, 1998.
[4] Ken Turkowski, "Fixed-Point Trigonometry with CORDIC iterations", Apple Computers, 1990.
[5] Branimir Reljin, "Teorija Električnih kola II", Akademska misao, December 2003.
[6] John D. Carpinelli, "Computer Systems, Organization & Architecture", Addison Wesley Longman, Inc., 2001.