Innovative technique for easy high-resolution position acquisition with sinusoidal incremental encoders

B. Höscheler, L. Számel, University of Erlangen, Institute for Electrical Drives,
Cauerstraße 9, D-91058 Erlangen, phone.: +49/ 9131/ 85 7657, fax: +49/ 9131/ 85 7658
email: bernd@eas.e-technik.uni-erlangen.de

Abstract

Currently known methods for high-resolution position acquisition with sinusoidal incremental encoders (i.e. VeCon) always require additional hardware components for counting whole cycles, two A/D-converters for quantifying the position within the current cycle and a routine for synchronising both information. The presented method archieves this first time only using the information coming from the two A/D-converters with a relatively simple algorithm. The most important fact is that the position may be calculated even at a rotation speed, where the fundamental frequency of the encoder signals exceeds the sampling frequency by a multiple value, so that the sampling theorem may be violated.
The paper discusses the development of this technique, the implementation of an additional error detection method, practical application as well as simulations and experimental results.

1. Introduction

Modern servo drives demand steadily increasing accuracy for position and speed measurement. Therefore it is useful to reduce the effort in sensor technology hardware (such as tacho generator, position sensor and hall-sensors) as far as possible by efficient digital signal processing.
Digital (optical) encoders seem to be a good solution, because the position is here available in digital form. Increasing dynamic behaviour and higher sampling frequencies however lead to an increasing problem, that may be clear with a simple example:
If for instance a speed resolution of less then 0.1 rpm at a sampling frequency of 2 kHz is required, a sensor with a minimum number of 300 000 cycles per revolution (c/r) will be necessary (using a quad edge counting method). Apart from the high cost of production, such an encoder will produce a fundamental frequency of already 15 MHz at a speed of 3000 rpm!
Hence digital encoders are more and more replaced by encoders with analog sine/cosine outputs, where position information is continuously available. With these encoders it is possible to calculate the position within one cycle, so that a smaller number of cycles per revolution (as 1000..5000) will be sufficient. Unfortunately, the effort in hard- and software increases: recent methods (as e.g. VeCon [3], [5]) are proceeding in a parallel structure of two components as seen in fig. 1.

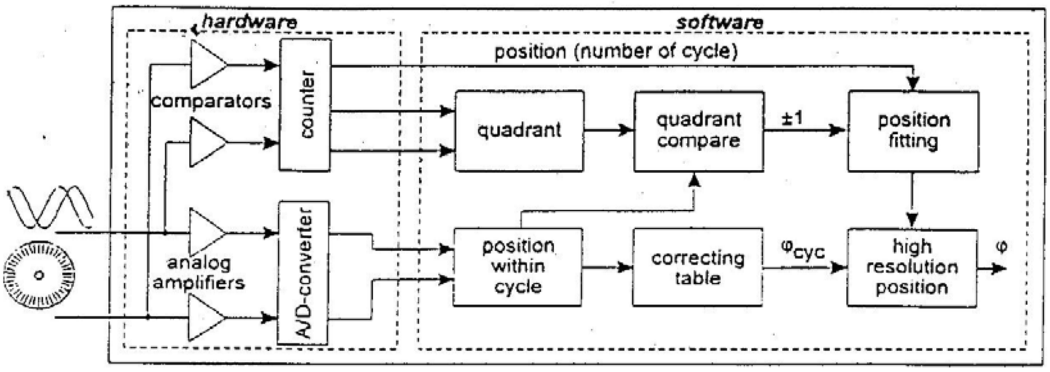

fig. 1. Schematic diagram of high-resolution-position-acquisition with sinusoidal incremental encoders.

Calculating the position is done here by combining different information:
First, the zero crossings of the encoder outputs are detected by comparators, and a hardware counter calculates the number of the current cycle in a 'classical' way. Apart form that, the number of quadrant within the cycle is detected. As a second point, the analog encoder tracks are sampled, using two A/D-converters. A software routine then computes the position within the current cycle and the quadrant within the cycle from it.
Now there is the problem to combine these data. Even a very small offset in time between counter and A/D-converter may lead to a situation, where both information belong to different cycles. In this case, calculating the absolute position will lead to a wrong value. Instead of that an additional block is required where the synchronism is proofed by comparing the two different quadrant values. If they are different, the counter has to be fitted by ±1, before counter value and position within the current cycle are formed to the value of the absolute position.
The presented method manages this in a much more easier way so that all components in fig. 1 marked in grey may be omitted. This is done by using a very effective algorithm in the block indicated as 'high resolution position'. But first lets see, how to calculate the position within one cycle.

2. Calculation of the position within a cycle

In fig. 2 (left part) the two analog encoder tracks are shown within one whole encoder cycle. The tracks are named as s(φ) and c(φ) to make the affinity to sine- and cosine like signals clear. φ stands for the absolute position, which is scaled to the number of cycles per revolution.

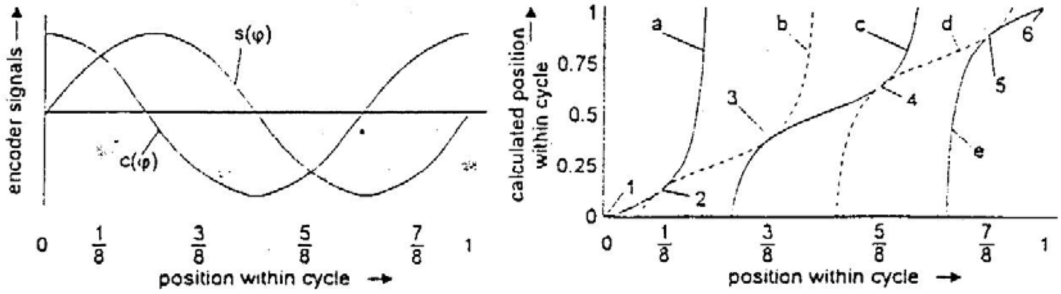

fig. 2. Analog encoder signals (left) and calculating the position within one cycle (right).

The object is, to calculate the position within one cycle as close as possible to the real value. One possibility is to use an arc sine- or arc cosine function for one of the encoder signals. Unfortunately the encoder tracks show aberrations from an ideal sinus, especially offset and amplitude errors are often observed.

Therefore a method is used which eliminates common amplitude errors. This is done by dividing the (absolute) smaller one of the encoder signal through the other and adding an offset. In fig. 2 (right part), 5 different functions named with 'a..e' are shown. These are the division rules which are very simple as the following table makes clear:

position within cycle	$0..\frac{1}{8}$	$\frac{1}{8}..\frac{3}{8}$	$\frac{3}{8}..\frac{5}{8}$	$\frac{5}{8}..\frac{7}{8}$	$\frac{7}{8}..1$
rule for calculating position within cycle	$\frac{1}{8}\cdot\left(\frac{s(\varphi)}{c(\varphi)}\right)$	$\frac{1}{8}\cdot\left(2-\frac{c(\varphi)}{s(\varphi)}\right)$	$\frac{1}{8}\cdot\left(4+\frac{s(\varphi)}{c(\varphi)}\right)$	$\frac{1}{8}\cdot\left(6-\frac{c(\varphi)}{s(\varphi)}\right)$	$\frac{1}{8}\cdot\left(8+\frac{s(\varphi)}{c(\varphi)}\right)$
curve, shown in fig. 2	a	b	c	d	e

tab. 1. Calculating the position within one cycle.

By choosing the right method depending to the position between one cycle, a monotonous and continuous function between the real and calculated position is reached, as it is pictured in fig. 2 by the curve going through the points 1 to 6. The decision which rule to take may easily be calculated with the sign of sum and difference of the two encoder tracks. But, the calculated position is just an approximation, ideal sinusoidal encoder tracks will lead to errors of about 1.2% of one cycle and can slightly be seen in fig. 2 in an aberration from a straight line. This error may be reduced by processing an arc-tangent/arc-cotangent function, as it is also done in the VeCon Chip with a lookup table. However having encoder track errors (e.g. offset) this correction will fail. So an individual correcting table for each encoder is used here. The problem yet is how to gain such a lookup table without using additional sensor technology. In this paper a method will be presented in section 5 where the reference measuring may easily be done without any other equipment.

3. Getting the absolute position

The next task is to gain the absolute position only from the calculated values of the position within one cycle. This would be very easy if there were always many samples per cycle. Because of the high number of cycles per revolution, encoders however reach high fundamental frequencies even at lower speed, so that the sampling theorem may be violated and a reconstruction of the position seems to be impossible.

This problem is explained with fig. 3 where an encoder with 2500 counts per revolution is analysed in a system with a sampling frequency of 10 kHz. Assuming a constant (high) acceleration of 15000 rad/sec^2 beginning at time zero and standstill will lead to a speed progression as shown in fig. 3a.

In fig. 3b the resulting encoder tracks are shown. On the left side a cutting of the first 10 samples can be seen where the speed is so low that the encoder cycles are sampled several times per cycle. On the right side of fig. 3b a cutting of the tracks is shown at a time, where the encoder reaches a speed of about 750 rpm and the tracks are sampled just only every third cycle (the fundamental frequency of the encoder is here 31 kHz!). Finally, the third row in fig. 3 shows the position within the current cycle and the calculated samples as demonstrated in the section above.

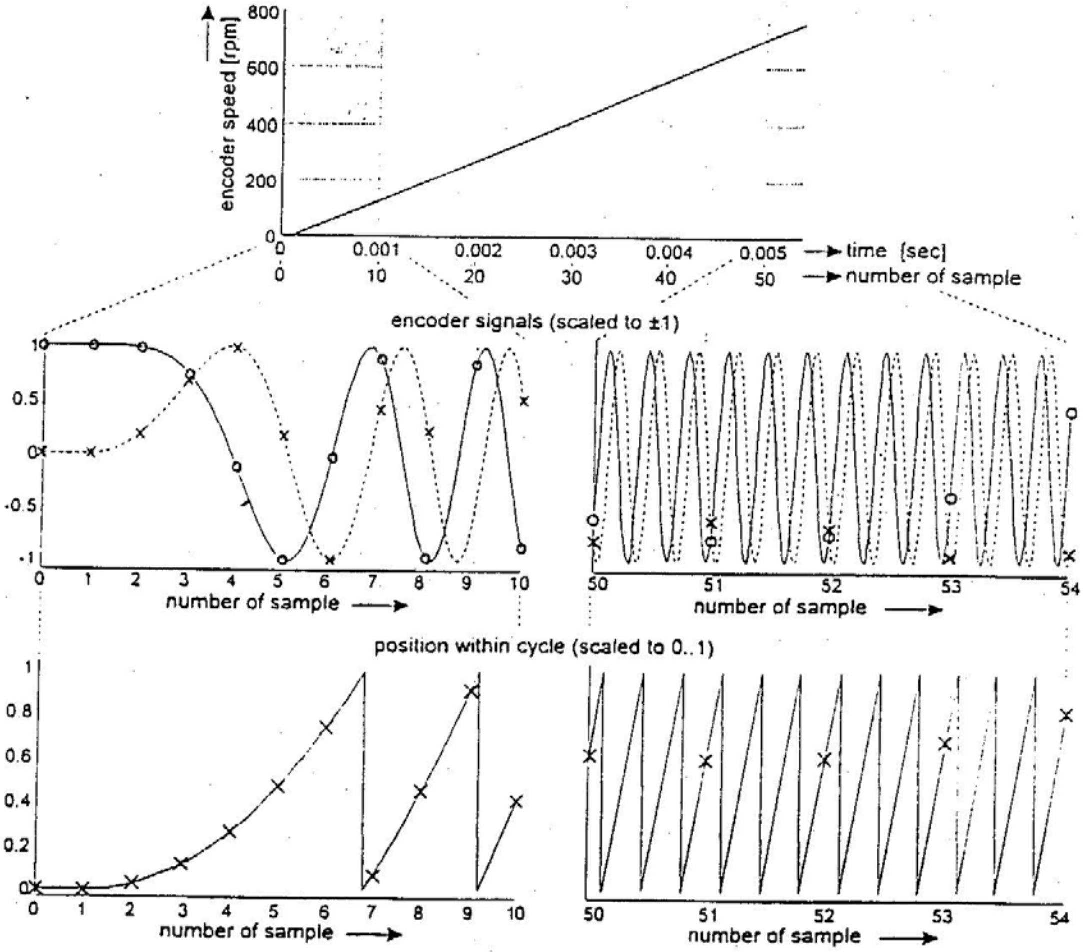

fig. 3. Encoder signals at a constant acceleration of 15000 rad/sec² (sampling frequency 10 kHz)
a. (above) encoder speed in rpm during the first 54 samples.
b. (middle) encoder signals (left: sample 0..10, right: sample 50..54)
c. (bottom) position within current cycle (left: sample 0..10, right: sample 50..54)

It is obvious that the samples on the right side of fig. 3 are at all insufficient to recognize the crossing of one (or even more) cycles and so reconstructing the absolute position will be nearly impossible. But we can use one more information:

A mechanical system can change its speed only at a limited acceleration rate. So in a sampled data system, every value of absolute position has to be in an finite interval compared to the last positions. Considering that the algorithm for the position within the cycle produces a value between 0 and 1, it will be clear that a reconstruction of the absolute position will be possible, if acceleration is so low that the current absolute position will be in an interval with the width of less than one. This will be shown in the following figure:

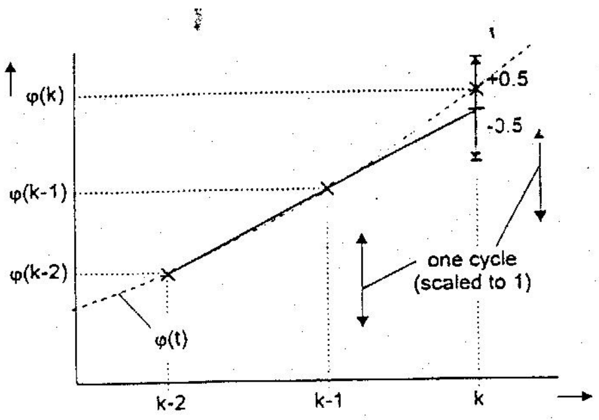

fig. 4: Possible interval (±0.5) of the current absolute position φ(k) postulating a certain maximum acceleration compared to the past two positions φ(k-1) and φ(k-2) (dashed: continuous progression of position).

If the speed would be constant (acceleration = 0) the current position φ(k) may be extrapolated by a line going through φ(k-1) and φ(k-2). Postulating that the current position may be in-between an interval around ±0.5 of this extrapolated value, a reconstruction is able with the following algorithm:

Denominating the difference of two successive position samples with u(k)

$$u(k) = \varphi(k) - \varphi(k-1) \tag{1}$$

we can forecast the absolute position φ(k) in the following interval:

$$\varphi(k) \in \left[\varphi(k-1) + u(k-1) - \frac{1}{2} \ldots \varphi(k-1) + u(k-1) + \frac{1}{2}\right[\tag{2}$$

Substituting (1), we get:

$$\varphi(k) \in \left[2 \cdot \varphi(k-1) - \varphi(k-2) - \frac{1}{2} \ldots 2 \cdot \varphi(k-1) - \varphi(k-2) + \frac{1}{2}\right[\tag{3}$$

Scaling the position φ(k) to cycles, it may be simply described by a sum of number of cycle (i) and position within cycle (φ_{cyc}):

$$\varphi(k) = i(k) + \varphi_{cyc}(k) \tag{4}$$

The possible number of the current cycle i(k) is then:

$$i(k) \in \left[2\varphi(k-1) - \varphi(k-2) - \varphi_{cyc}(k) - \frac{1}{2} \ \ldots \ 2\varphi(k-1) - \varphi(k-2) - \varphi_{cyc}(k) + \frac{1}{2}\right[\tag{5}$$

With the abbreviation

$$c(k) = 2\,\varphi(k-1) - \varphi(k-2) - \varphi_{cyc}(k) \tag{6}$$

we can say

$$i(k) \in \left[c(k) - \frac{1}{2} \ldots c(k) + \frac{1}{2}\right[\tag{7}$$

Because i(k) has to be Integer there is only one definite value that may be estimated simply by rounding:

$$i'(k) = \mathrm{Round}(c(k)) \tag{8}$$

The absolute position may so be calculated in the following way:

$$\varphi(k) = \mathrm{Round}\left(2\,\varphi(k-1) - \varphi(k-2) - \varphi_{cyc}(k)\right) + \varphi_{cyc}(k) \qquad (9)$$

The formula shows that the three values $\varphi(k-1)$, $\varphi(k-2)$ and $\varphi_{cyc}(k)$ are sufficient to gain $\varphi(k)$ in an recursive algorithm independent of the rotational speed.

This algorithm was implemented on a signal processor DSP32C (50 MHz) from AT&T where calculating the absolute position requires only 780 nsec. The whole program including sampling the encoder signals, calculating and correcting the position between the cycles, calculating the absolute position and counting revolutions is done in less than 8 μsec, so that a sampling frequency of 25 kHz was chosen and excellent dynamic behaviour is archived. For testing the algorithm at an extreme point where only little encoder information is available, the sampling frequency was reduced from 25 kHz to 1 kHz. Then, the encoder was driven with a speed of ± 2000 rpm. Even at the maximum speed, when the algorithm gets only one sample every 83th cycle, the algorithm worked without any errors.

But there are two points we have to respect: the start-up and the acceleration limit.

Concerning the first problem, the algorithm has to be started at standstill or such a low speed where the position changes less than half a cycle during a sampling interval. This will in practice lead to no problem when starting the algorithm at the time the system is put into operation.

Another point is, the restriction of maximum acceleration. Calculating this leads to a rule in the following form:

$$\frac{d\omega}{dt} \le \frac{\pi}{\text{cycles per revolution}} \cdot f_s^2 \qquad (10)$$

where f_s is the sampling frequency of the encoder tracks. The following figure shows that fact:

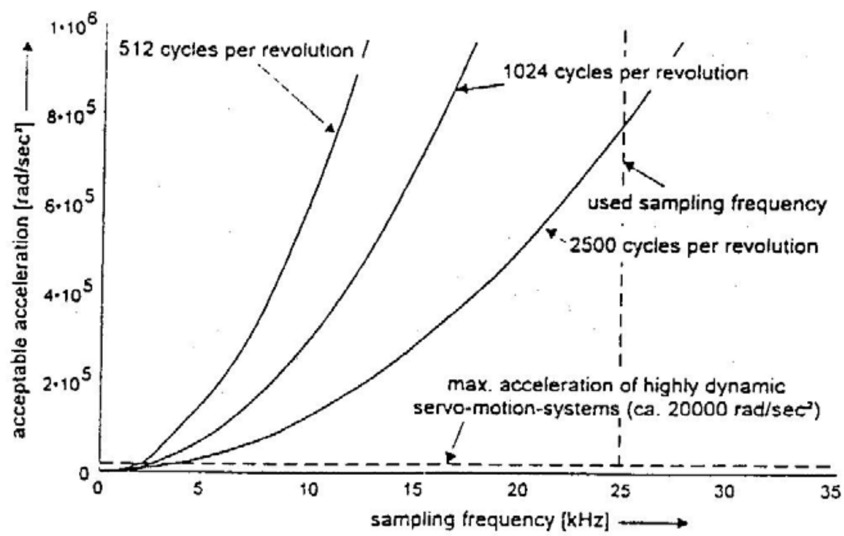

fig. 5. Maximum acceptable acceleration for different types of encoders and sampling frequencies.

In the implemented case (encoder with 2500 c/r and a sampling frequency of 25 kHz) there can be seen, that a maximum angular acceleration of $\dot{\omega}_{max} \approx 785000\ \mathrm{rad/sec^2}$ is still allowed. Compared to accelerations of highly dynamic servo machines ($\approx 20000\ \mathrm{rad/sec^2}$), the restriction get a rather constant speed? Speed control in a closed loop is impossible because the encoder itself yet has not been calibrated. But DC-motors can be supplied with a constant voltage and AC-motors with a constantly rotating voltage phasor. The following figure shows the typical speed of a brushless DC motor (sinusoidal e.m.f.) which is supplied in that way.

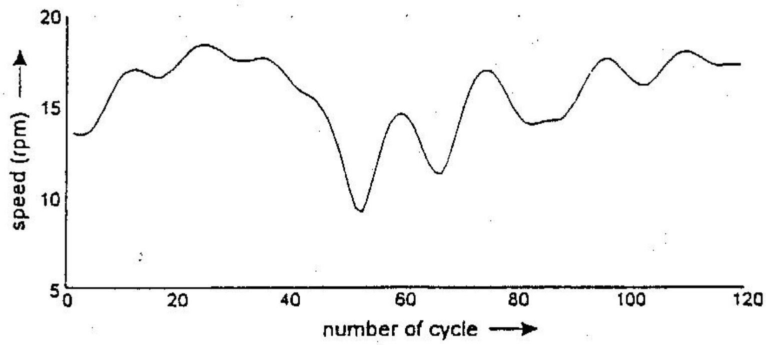

fig. 6. Speed measurement of a brushless DC motor, supplied with a constantly rotating voltage phasor, rotating at a mechanical speed of 15 rpm (encoder with 2048 c/r).

In the figure a obvious fluctuating speed can be seen, which is caused by the magnetic rest positions of the motor. It is clear, that the assumption of constant speed is no longer valid in this case and the encoder tracks will be influenced. The following picture shows a cutting from the encoder sine track resulting from such a speed:

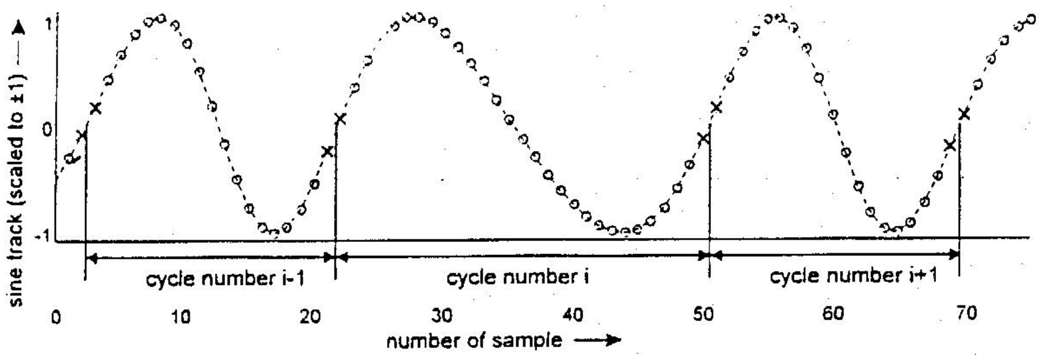

fig. 7. Samples of encoder sine track (sampling frequency 50 kHz) at a speed fluctuation similar to fig. 6 within 3 cycles (for better recognising graph is exaggerated).

If the distortion in time would only be slightly, φ_{cyc} would linear increase from 0 to 1 between the two positive zero crossings of the sine track (in fig. 7 sample 21/22 and sample 50/51). But, regarding the fluctuation, the position has to be calculated more accurately. This is done by estimating a polynomal of 3rd order, which is determined with the four positive zero crossings liming the two neighbouring cycles. The exact time for every crossing is calculated by an interpolation between the two sine track samples around the crossing (marked with 'x'). Having calculated the polynomal rule, the real position $\varphi_{cyc}(k)$ and the uncorrected position $\varphi'_{cyc}(k)$ is then calculated for every sample (k) within the cycle. As the wanted lookup table is of the shape $\varphi_{cyc}(\varphi'_{cyc})$, the samples have to be recalculated to equidistant values of φ'_{cyc}, which is done with cubic spline interpolation. This calculation is individually done for a large number of cycles. The following figures show some calculated lookup tables measured at the fluctuating speed shown in fig. 6. Additional to that, the lookup tables were calculated as if the speed was constant and a linear interpolation would fit the position:

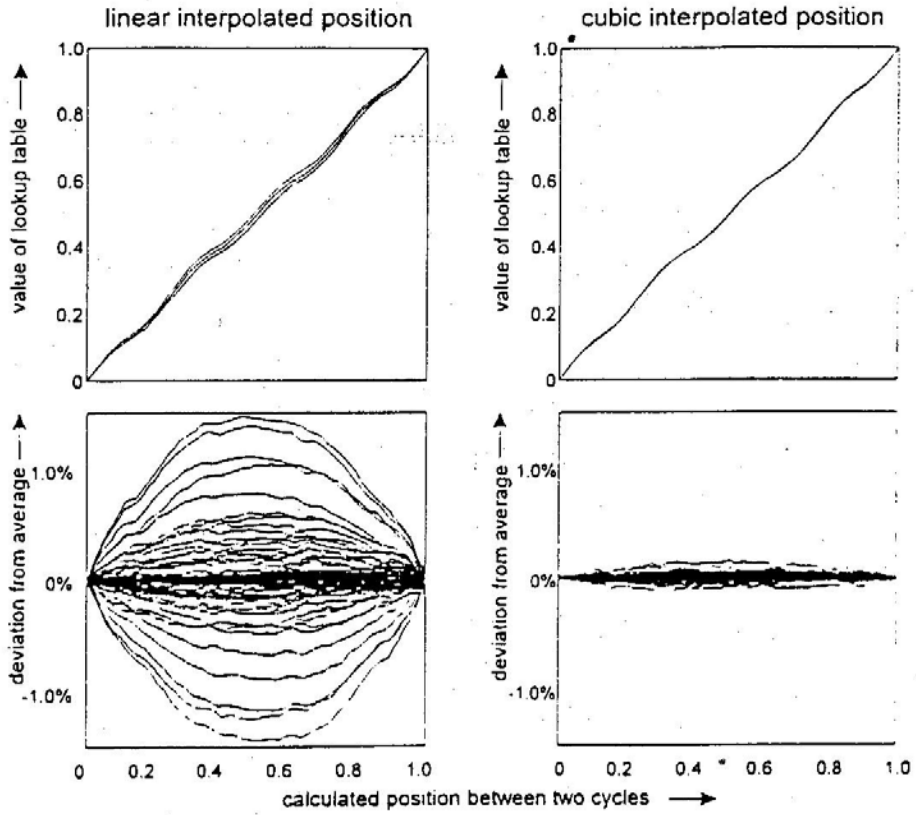

fig. 8. Calculated lookup tables generated during a measurement with speed of fig. 6.
above: maximum, average and minimum of the lookup tables.
bottom: deviation from average.
left: assuming constant speed, using linear interpolation for the position.
right: considering fluctuation speed by using a polynomal of 3rd order for the position.

The upper row shows lookup tables with maximum, average and minimum over the 120 cycles which can be seen in fig. 6. As the deviation between the curves is only very little, it is more meaningful to watch the deviation from average as shown in the bottom row. It can be seen, that the method using a polynomal of 3rd order produces deviations of less than one 10th compared to that method assuming constant speed and calculation with linear interpolation. The remaining error (maximum deviation 0.15%) is reduced furthermore by measuring every cycle number around one complete rotation and averaging.

The program on the DSP (see section 3) was enlarged with that algorithm so that a reference measurement is possible, whenever desired.

6. Conclusion and future work

The described method for position acquisition is a progress concerning the following aspects :

- Usually required hardware may be omitted by using an additional algorithm which is very easy to implement.
- The algorithm will have no limit concerning a maximum rotational speed. The allowed maximum acceleration is so high, that it will have little impact in practice.
- There is nearly no restriction for the shape of the encoder tracks. Harmonic distortions and offset of the encoder tracks are allowed without producing additional errors, if the proposed algorithm for gaining an individual correcting lookup table is used.
- Calculating the position by dividing the encoder tracks, the algorithm is non-sensitive regarding common amplitude errors of the encoder tracks.
- A very high resolution is available which is primary be influenced by the quantization of the used A/D-converter. Using 12 bit converters, a resolution of an 1/5800 cycle will be reached, if the quantizising error is ±1 LSB (last significant bit).
- An additional error detection is very easy to implement. With that an exceeding acceleration is detected as well as an error in the incoming cables.

The future investigations aim to get a further improvement of the achieved accuracy. As measurements have shown, encoder tracks also have typical fluctuations in offset and amplitude dependent on the position. The idea therefore is to combine two position calculations: the first one will compute an approximate value of the absolute position. This value will be used to compensate the track errors. The second calculation then will calculate the absolute position once more, but using the corrected encoder tracks.

Annotation:
This work was developed during a program promoted by the Deutsche Forschungsgemeinschaft (DFG).

References:

[1] Klug R. D. 'Echtzeitsimulation mechanischer Lastsysteme zur Prüfung elektrischer Servoantriebe', dissertation at the University Erlangen 1992.

[2] Höscheler B., "Erzeugung dynamischer Drehmoment- und Bewegungsabläufe nichtlinearer mechanischer Systeme zur Echtzeitsimulation der Beanspruchung von Komponenten", final report in the research program of the Deutsche Forschungsgemeinschaft (DFG) "System Integration of Electrical Drives", University Erlangen.

[3] Maiworm G. 'Einsatz neuester Spitzentechnologie in digitalisierten Servoverstaerkern' Antriebstechnik, nr. 12-1994.

[4] Pool K. 'High Resolution Optical Encoders' 1995, article from the WWW: 'http://opticalencoder.com/COPI_95_art.html'.

[5] Kiel E., W. Schumacher 'Der Servocontroller in einem Chip' Elektronik nr. 8-1994.