Pseudorandom Position Encoder and Code Conversion Problems

Dragan B. Denić[1], Ivana S. Ranđelović[2] and Milica P. Rančić[3]

Abstract **- Basic methods related to functioning of currently known pseudorandom position encoders with serial code reading are considered. In particular, problems of pseudorandom to natural code conversion are discussed. It is indicated to the possibility of code conversion time reducing in regard to the standard method of serial pseudorandom to natural code conversion.**

Keywords **- position measurement, pseudorandom code, optical encoders**

I. INTRODUCTION

In order to avoid use of large number of code tracks and still obtain high-resolution measurements, a new type of absolute encoders, named pseudorandom absolute encoders, was developed. It considers use of cyclic or serial codes, which possess a feature that two *n*-bit code words that correspond to two successive positions contain an identical sequence of (*n-1*) bits. In other words, the last (*n-1*) bits of the current code word are equivalent to the first (*n-1*) bits of the subsequent code word. A possibility of overlapping of all 2^n code word records on one code track is evident, [2]. To begin with, such encoder has an enormous advantage not only because it solves the problem of increasing the number of code tracks for the purpose of high-resolution system, yet it always has only one code track regardless to the resolution. Due to well-known and simple rules for its later conversion into the natural code, the pseudorandom code is applied. However, we would still need *n* sensors for the instant reading of *n* output code bits, which, in case of high resolutions, becomes a technical problem of allocating *n* sensor heads within a small physical area.

Fortunately, cyclic code properties provide a new way for code bits reading using only one detector, [3]. This method considers code bits collecting into a code-forming shift register. Only one bit is being read for each new position of the movable system and entered into the mentioned shift register. After the initial movement that corresponds to space width of *n* bits, forming of the code word, which corresponds to the current position of the movable system, will be done. For each of the following positions a new bit is being read, and along with (*n-1*) bits of the previous code word, an output code word of the new position is obtained. Evidently, there is a necessity of initial moving after the first plugging in, meaning that the movable system (MS) crosses a distance equivalent to space width of *n* code bits, so that the first valid output *n*-bit code could be formed. This is the reason that these absolute encoders are called virtual absolute encoders. In the case of high-resolution encoders, mentioned distance of initial movement is very small. However, this is still a virtual absolute encoder's disadvantage. This disadvantage becomes almost negligible regarding the new quality that is provided by the virtual absolute encoder. After pseudorandom code reading is done, it is necessary to convert it into the natural code. In the case of high-resolution encoders, the method of parallel conversion using memory elements is not acceptable. Since the pseudorandom code is so specific [1], it is possible to apply serial code conversion method [2]. A disadvantage that comes from the significant code conversion time could be lightened using the code conversion algorithm proposed in the paper.

II. BASIC FUNCTIONING PRINCIPLE OF THE VIRTUAL ABSOLUTE ENCODER

Virtual absolute pseudorandom encoder is nowadays a big hit that, day by day, straightens its position on the market in regard to the classic absolute encoders. So that main problems would be pointed out, a concrete simple example of the virtual absolute encoder will be discussed.

A rotary disc consists of two tracks, Fig. 1. Let us consider that these two tracks consist of transparent and non-transparent segments. Also, let us consider that appropriate optical methods for detection are applied. Interior track is identical to the one of incremental encoder, [4], and it is used in this example for generating two bits in the output code word with the smallest weight. Its main role is providing synchronized code reading and it is often called synchronization or clock or time track, [5].

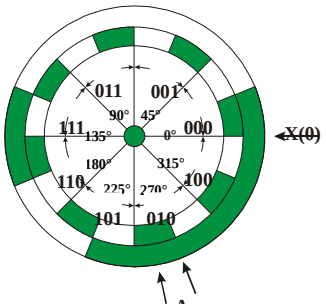

Fig. 1. Virtual absolute encoder disc

In this simple example, which considers 5-bit binary encoder, it is adopted that the space-time width of one incremental cycle is equivalent to the space width of one code track bit. Otherwise, that ratio can change. External code track is coded in a way to provide residual important bits needed for forming

[1] Dragan B. Denić is with the Faculty of Electronic Engineering, University of Niš, Aleksandar Medvedev 14, 18000 Niš, Serbia and Montenegro, e-mail: ddenic@elfak.ni.ac.yu

[2] Ivana S. Ran|elović is with the Faculty of Electronic Engineering, University of Niš, Aleksandar Medvedev 14, 18000 Niš, Serbia and Montenegro, e-mail: rivana@elfak.ni.ac.yu

[3] Milica P. Rančić is with the Faculty of Electronic Engineering, University of Niš, Aleksandar Medvedev 14, 18000 Niš, Serbia and Montenegro, e-mail: milica@elfak.ni.ac.yu

of the complete absolute output code word. Applied cyclic code, named shift register code [1], provides a unique code word for each new position of the encoder, which alludes reading of a new bit from the code track.

For obtaining the output position code, three detectors are being used. Serial bit reading from the code track is done by detector X(0). Obtaining a signal from the synchronized track is realized using two detectors, as in case of conventional incremental encoder, [4].

In this example, classic quadrature signals are required (two sine signals dislocated by 90°), because two additional bits are planned that would magnify the position measurement resolution four times. These two signals are also used for determining the rotation direction of the encoder disc. These signals are then shaped into rectangular signals, and whenever a transition of signal A (with signal B on logical "0") is detected, reading of a new code bit is being performed. In order to entirely explain a principle of serial code reading, an example of realization of electronic block of this virtual absolute encoder is shown in Fig. 2.

A LED diode used as a light source for track synchronization is always actuated and it illuminates two detectors forming quadrature signals A and B at the comparator outputs C_1 and C_2. As said before, code reading is done whenever a transition of the signal A (with signal B on logical "0") takes place. Because of this, signal A goes to the input of a signal edge detection circuit, and then, the output signal of this circuit along with the signal B complement are led to the input of the AND circuit I_1. Whenever an impulse at the logical AND circuit output appears, a new bit reading is done. A simple realization of the signal edge detector is presented here. Rectangular signal A from the comparator output C_1 is brought to both inputs of the same EXOR circuit E_1, but with small delay at one of the inputs. In this case, the delay is generated using integrator in the form of RC circuit.

Whenever an impulse appears at the transistor T base, it leads, whereby the LED diode, which illuminates the code track, is excited. Considering that at that moment the code track detector is located at the middle of the sector that defines current code track bit, reliable reading of that bit can be done. A logical value of the read bit is located at the comparator output C3. That bit is brought to the appropriate shift register input depending on the disc rotation direction. Considering that impulses at the signal edge detector output always appear at the moment immediately after the detected transition on the synchronization track, then, based on the logical value of the quadrature signal A, encoder rotation direction is being determined, Fig. 1. If A equals "1" when the impulse appears, then the rotation direction is clockwise (CW). Then, an impulse appears at the output of the logical AND circuit I_4, shift register shifts to the left and newly read bit is accepted at the appropriate shift register input. In the case of reverse encoder disc rotation, an impulse occurs at the output of the logical AND circuit I_5. After the initial movement of (*n-2*) bits in the same direction, a correct code word is formed and valid position information is at the output. It is obvious, that it is necessary to preconvert a cyclic code at the shift register output into a desired output code, usually into the natural binary code. There are few known methods for code conversion, [6, 7], one of which named parallel conversion method that uses a table memory located in PROM, is applied here. At the end, two bits of the smallest weight are obtained at the quadrature detector output, which consists of one EXOR circuit and one logical NOT circuit.

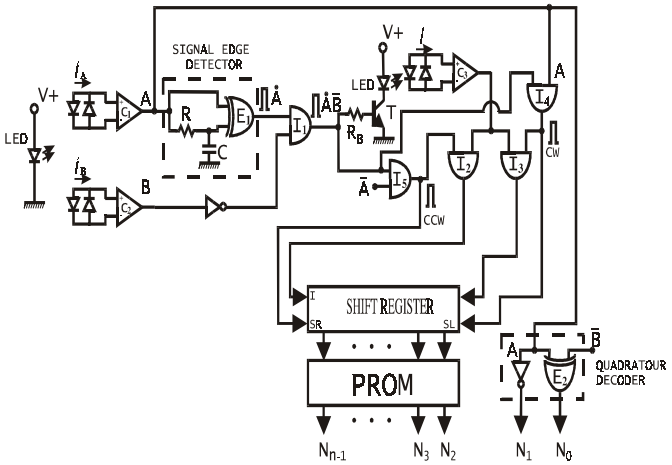

Fig. 2. An example of the virtual absolute encoder electronic block realization

Basic reason for using of the impulse stimulus of the LED diode is that there is one gap for bit reading, in contrast to conventional incremental optical encoders where a number of gaps is used for fine tracks observing (multiple-line slits). Impulse stimulus allows greater pick values for current, whereby greater momentary illumination is achieved and thus, a probability of amplitude loss due to usage of only one gap (single-line slit) is reduced. Only in the case of low-resolution measurements when gap is width enough to provide enough signal amplitude from photo detector, DC activating of the LED diode is possible.

Illustrated example in a simple way presents the manner in which the virtual absolute encoder functions. A possibility of applying the parallel code conversion method is indicated, although, generally, it is not the best solution, especially in the case of encoders for high-resolution position measuring. Therefore, an application of the serial code conversion method is suggested, where the PROM memory, shown in Fig. 2., is simply replaced by the appropriate serial code converter.

III. A NEW SERIAL CODE CONVERSION METHOD

A pseudorandom binary sequence of length 2^n-1 is considered, that is:

$$\{S(p) / \ p = 0,1,.....2^n - 2\} \qquad (1)$$

Since it is written with one bit per sector, a code track is divided in 2^n sectors. Term $S(p)$ represents content of the n^{th} element of the shift register after p shifts to the left. Considering that pseudorandom sequences are periodical, it is of no importance which *n*-bit word is adopted as the initial one.

$$\{S(n-k) / k = n,.....,1\} \qquad (2)$$

Further, it can be considered that the pseudorandom binary sequence is generated using a shift register with a feedback, according to the following algorithm.

$$X(0) = X(n) \oplus c(n-1)X(n-1) \oplus \ldots\ldots\ldots\ldots \otimes c(1)X(1),$$
$$X(i) = X(i-1), \text{ for } i = n,\ldots\ldots\ldots 1,$$

where feedback coefficients $c(n-1)$ are foreclose defined, as shown in Table I.

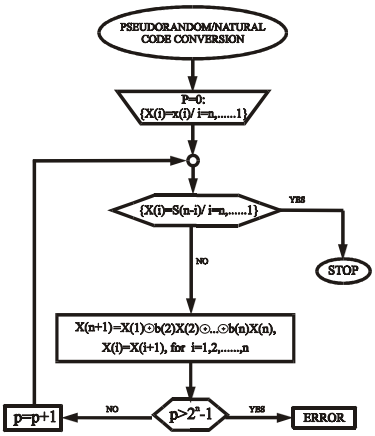

Fig. 3. The code conversion algorithm

$$X(0) = X(n) \oplus c(n-1)X(n-1) \oplus \ldots\ldots\ldots\ldots \otimes c(1)X(1)$$
$$X(i) = X(i-1), \text{ for } i = n,\ldots\ldots\ldots 1,$$

Conversion of these n bits into the natural code, following the serial code conversion method, is performed based on the property of the PRBS generating algorithm that it is reversible. The code conversion algorithm is shown in Fig. 3. The content of the register for code word generating, for the given position p, will be

$$\{S(p+n-k)/k = n,\ldots\ldots,1\}, \qquad (3)$$

for each $p = 0,1,..,2^n - 2$.

It is based on the idea that it is possible to find the actual value of the position p, simply by counting the steps that the shift register with the inverse feedback needs until it reaches the initial state by successively shifting from the read pseudorandom n-bit word (2). Therefore, the code conversion algorithm is started (Fig. 3.) setting the current value with the n-bit quantity "X", which is gained using reading heads. Afterwards, "X" is being cyclically modified in accordance to the inverse generation low given in Table I. The algorithm will be in the cycle until the "X" reaches the defined initial state (2). Finally, when this state is reached, the algorithm stops, and current value of the register represents n-bit natural code of the real position. A new pseudorandom to natural code conversion algorithm is shown in Fig. 4.

In the case of high resolution, the code conversion time becomes a limiting factor for linear moving or rotation speed. The new algorithm reduces the maximum conversion time approximately two times. It is based on the idea that, thanks to PRBS cycling property, the initial state (2) could be reached using feedback relations that are used for either "direct" or "inverse" PRBS generating. Depending on the preceding position of the transporting system it is decided which PRBS generating low would be used for the current code conversion. When the relation for the "direct" PRBS is used, the real position will be 2^n-1-p. Associating to the information about the preceding position is not of big importance, because eventual absence of the information (e.g. when starting the system) does not affect the code conversion accuracy, and it could only exceed the code conversion time.

TABLE I

Shift register length n	Feedback in case of direct PRBS $X(0) = X(n) \oplus c(n-1)X(n-1) \oplus \ldots \otimes c(1)X(1)$
3	$X(0) = X(3) \oplus X(1)$;
4	$X(0) = X(4) \oplus X(1)$;
5	$X(0) = X(5) \oplus X(2)$;
6	$X(0) = X(6) \oplus X(1)$;
7	$X(0) = X(7) \oplus X(3)$;
8	$X(0) = X(8) \oplus X(4) \oplus X(3) \otimes X(2)$;
9	$X(0) = X(9) \oplus X(6)$;
10	$X(0) = X(10) \oplus X(3)$;
11	$X(0) = X(11) \oplus X(2)$;
12	$X(0) = X(12) \oplus X(6) \oplus X(4) \otimes X(1)$
13	$X(0) = X(13) \oplus X(10) \oplus X(6) \otimes X(4)$;
14	$X(0) = X(14) \oplus X(13) \oplus X(8) \otimes X(4)$;
Shift register length n	Feedback in case of inverse PRBS $X(n+1) = X(1) \oplus b(2)X(2) \oplus \ldots \otimes b(n)X(n)$
3	$X(4) = X(1) \oplus X(2)$;
4	$X(5) = X(1) \oplus X(2)$;
5	$X(6) = X(1) \oplus X(3)$;
6	$X(7) = X(1) \oplus X(2)$;
7	$X(8) = X(1) \oplus X(4)$;
8	$X(9) = X(1) \oplus X(3) \oplus X(4) \otimes X(5)$;
9	$X(10) = X(1) \oplus X(3)$;
10	$X(11) = X(1) \oplus X(4)$;
11	$X(12) = X(1) \oplus X(3)$;
12	$X(13) = X(1) \oplus X(2) \oplus X(5) \otimes X(7)$;
13	$X(14) = X(1) \oplus X(5) \oplus X(7) \otimes X(11)$;
14	$X(15) = X(1) \oplus X(5) \oplus X(9) \otimes X(14)$;

Both software and hardware realizations of the pseudorandom to natural code conversion algorithms, shown in Fig. 3. and Fig. 4., could be done. Comparing of software realizations of two algorithms is done using a micro computing development system based on a microprocessor Intel 8051, which operates at 12 MHz. Maximal gained code conversion time for the solution described in [2] is 5.6 ms. Maximal conversion time, for the solution proposed in this paper, is 2.8 ms. This means that applying the hardware realization of the new algorithm, shown in Fig. 5., an improvement would be achieved in regard to the hardware realization of the solution in Fig. 3 [2].

Another bidirectional shift register, depending on the MSB bit of the output binary information of the transporting system preceding position, performs shifting to the left or to the right from the initial state (2). A counter counts those shifts and, in the case of shifting to the right, its output is the actual value of the position. When shifting to the left, the value that corresponds to the actual position is obtained by

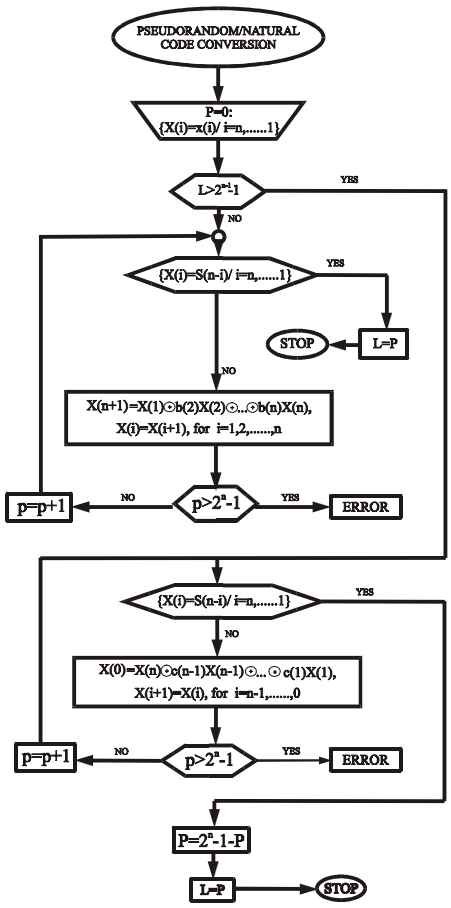

Fig. 4. The new code conversion algorithm

complementing the counter output, which is equal to 2^n-1-p in the algorithm. The problem that occurs because of whether output should be complemented or not, depending on a shifting direction, is simply solved using EXOR circuits, Fig. 5.

At the end, it should be mentioned that the system must involve a functional part, which would, after the system is turned on, signalize that the read code is wrong for the first n quanting steps. There are both software and hardware solutions to this problem. The software solution considers a subroutine that is automatically called for after n information on position. Appropriate hardware solution that uses an additional counter is described in detail in [2].

IV. Conclusion

The virtual absolute encoder is currently the greatest hit, as something new with an entirely new quality. They are especially interesting because they own great number of possibilities for further upgrading of their performance. Their price is less than the one of conventional absolute encoders, in return of great new quality. The system reliability is increased as well as the possibility of providing additional information to the user about the output measuring information validity. An important function that significantly affects pseudorandom encoder performances is pseudorandom to natural code conversion. The pseudorandom to natural code conversion is

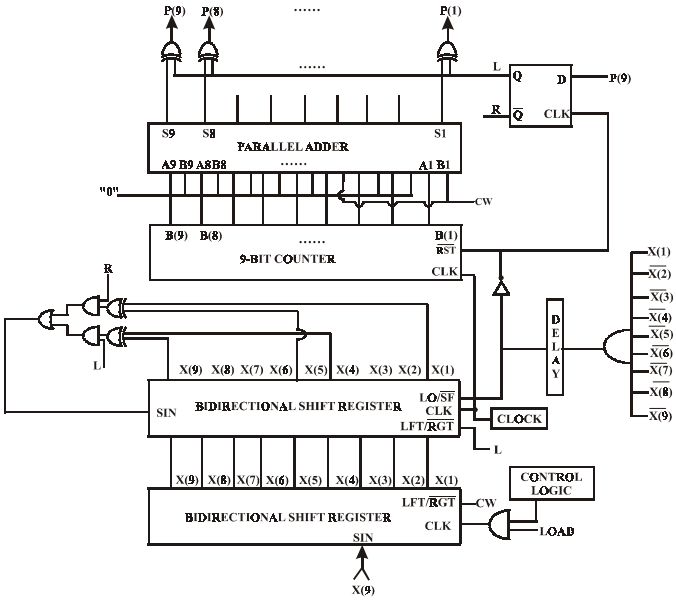

Fig. 5. Hardware realization of the pseudorandom/natural code conversion

one-way and can be performed using a memory with a translation table. However, this method is not practical for pseudorandom binary sequences of relatively long length, so it is of great importance to develop a code conversion method that, even for very long sequences, provides simple conversion into the natural code. A new approach to code conversion suggested in this paper, provides, in a simple way, an acceleration of the code conversion algorithm approximately two times.

References

[1] MacWilliams, F.J., Slone, N.J.A.: "Pseudo-random sequences and arrays", *Proceeding of IEEE*, Vol. 64, No. 12, pp. 1715-1728, December 1976.

[2] Petriu, E.M.: "Absolute-type position transducers using a pseudorandom encoding", *IEEE Trans. Instrum. and Meas.*, Vol. IM-36, No. 4, pp. 950-955, December 1987.

[3] Petriu, E.M., Basran, J.S.: "On the position measurement of automated guided vehicles using pseudorandom encoding", *IEEE Trans. Instrum. and Meas.*, Vol. 38, No. 3, pp. 799-803, June 1989.

[4] Wigmore, T.: "Optical shaft encoder from SHARP", *Elektor Electronics*, pp. 60-62, July/August, 1989.

[5] Petriu, E.M., Basran, J.S., Groen, F.C.A.: "Automated guided vehicle position recovery", *IEEE Trans. Instrum. and Meas.*, Vol. 39, No. 1, pp. 254-258, February 1990.

[6] Petriu, E.M.: "New pseudorandom/natural code conversion method", *Electronics Letters*, Vol. 24, No. 22, pp. 1358-1359, 1988.

[7] John G. Webster: "The measurement, instrumentation and sensors handbook", CRC Press and IEEE Press, 1999.