Proceedings of the 2009 IEEE International Conference on Mechatronics.
Málaga, Spain, April 2009.

An algorithm to design prescribed length codes for single-tracked shaft encoders

Borja Balle*, Enric Ventura† and Josep M. Fuertes‡
*Departament de Llenguatges i Sistemes Informàtics
†Departament de Matemàtica Aplicada 3
‡Departament d'Enginyeria de Sistemes,
Automàtica i Informàtica Industrial
Universitat Politècnica de Catalunya

***Abstract*—Maximal-length binary shift register sequences have been known for a long time. They have many interesting properties, one of them is that when taken in blocks of n consecutive positions they form $2^n - 1$ different codes in a closed circular sequence. This property can be used for measuring absolute angular positions as the circle can be divided in as many parts as different codes can be retrieved. This paper describes how a closed binary sequence with arbitrary length can be effectively designed with the minimal possible block-length, using linear feedback shift registers (LFSR). Such sequences can be used for measuring a specified exact number of angular positions, using the minimal possible number of detectors allowed by linear methods.**

I. Introduction

In many application areas, including angular position control systems, careful selection of sensing components is key to providing the best application performance. Traditional angular position sensing devices are incremental or absolute shaft encoders. Such angular encoders are used for measuring the angular position of an object by detecting marks on a scale affixed to the object axis. Incremental encoders are lightweight as they only need a circular marked corona on a disk and a sensing element, usually a light detector, which detects and recognizes the switching light beam as the axis rotates. Their major drawback is that incremental encoders require some means for synchronizing to obtain the axis position. Absolute encoders provide the ability to remember the object position following any power interruption. This is one of the reasons why absolute encoders are used where a high safety standard is required, typical applications are for speed or position control circuits in aerospace and aviation, positioning mechanisms, computer-aided machinery, semiconductor manufacturing, inspection equipment, machine tools and robotics, medical imaging, telescopes and other instruments.

Absolute angular position measurement is usually carried out by transducers that expand a different n-bit code word for each of a finite number of angular positions. One of the common components of such transducers is a marked disk with as many tracks as bits the angular positions to be sensed have. Traditional disks use a radial bit sensing method that consists in an arrangement of blacks and whites ("1" and "0") distributed in concentric coronas. Most commercial transducers use the Gray coding bit distribution to reduce the different scanning errors. But such coding has two drawbacks: as the resolution (and so the number of bits) increases, the disk diameter must also increase; and secondly, the number of sectors has to be exactly a power of 2. For the first drawback, there is a method that uses only one bit code track, based on the window property of pseudorandom binary sequences. Such property states that in a pseudo-random cyclic code expansion, all the n-bit elements that can be successively taken are different to each other. The result is that once the pseudo-random binary sequence is expanded in the circular corona, there are as many different measurements as the length of the cyclic code expansion. In this case, the sensing elements are not radially but tangentially distributed. There are several papers stating such configuration, see [1], [2], [3], [4], [5], [6]. Next question is about the number of sectors. We need to produce a pseudo-random cyclic code expansion, all of whose n-bit subwords are different to each other, and having a prescribed length $e > 2$.

The rest of this paper is organized as follows. In section II the preliminaries about shaft encoders and LFSRs are recalled and some notation is introduced. The problem to be solved is also stated there. The relation between this problem and the orders of polynomials over finite fields are the subject of section III. Using these results, in section IV an algorithm to solve the problem is presented and its complexity analyzed. The conclusions of this work are summarized in section V. An appendix can be found at the end of the paper containing some basic results about polynomials over finite fields which are used extensively throughout the paper.

II. Absolute shaft encoders and LFSRs

A. Absolute shaft encoders

A shaft encoder is an electro-mechanical device used to convert the angular position of a rotating rod into an electrical signal. The nature of the signal can be either analog or digital, and the signaling policy can be either incremental or absolute. Here we will only consider digital absolute shaft encoders and we will refer to them simply as shaft encoders.

The usual setup to build a shaft encoder is to attach to the rotating rod a ring tangentially divided into e equal sectors, where $360/e$ is the angular resolution of the encoder. Each of these sectors is then uniquely labeled and a reader capable of

978-1-4244-4195-2/09/$25.00 (c) 2009 IEEE

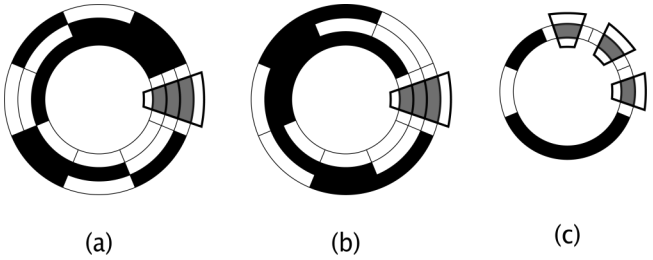

Fig. 1. Shaft encoders using different codes

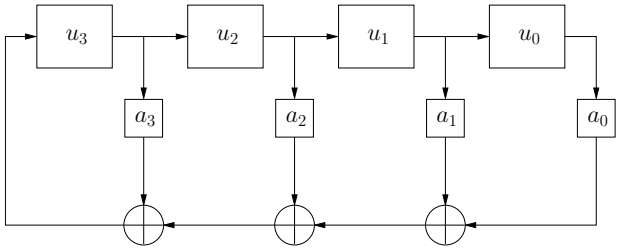

Fig. 2. LFSR of length 4 with Fibonacci architecture

reconizing the labels and giving a different output signal for each of them is attached to the rod in a fixed position. As the ring rotates with the rod, the label under the reader changes and the output signal changes accordingly. Fig. 1 (a) and (b) show two common implementations of this setup for $e = 8$ using natural and gray codings respectively. Note the reader is composed of $\lceil \log_2 e \rceil$ binary detectors, one for each track. If q-ary detectors were available, equivalent encoders could be built with $\lceil \log_q e \rceil$ tracks.

Another important but less known implementation of a shaft encoder is one which uses a single track. Here the detectors are distributed tangentially over the track. If q-ary detectors are used, $\lceil \log_q e \rceil$ detectors are needed and the track is labelled using a circular sequence of length e such that every word composed of $\lceil \log_q e \rceil$ consecutive symbols from the sequence is different from the rest. An example of such a track with $q = 2$ and $e = 8$ is shown in Fig. 1 (c).

The major advantage of single-tracked versus multi-tracked shaft encoders resides in the space saving achieved by reducing the number of tracks and the resulting reduction of moving mass, specially useful in critical mass applications (robotic arms, telescopes [7], [8], space applications).

To build a single-tracked encoder one needs first to construct the sequence with which to label the track. In general, this problem is not easy. Although Lempel proved in [9] that these sequences always exist for any combination of the parameters q and e, no efficient algorithm is known to solve the problem of finding one of these sequences for any given q and e. However, when $e = q^n - 1$ for some positive integer n and q is a power of some prime p, maximal LFSRs are knwon to generate such sequences.

The goal of this paper is to present an efficient method to solve this problem for q being a prime power and with a relaxation on the size of the words. More specifically, given a positive integer e and $q = p^m$, with p a prime number and m a positive integer, our method constructs a circular sequence of length e over an alphabet of q symbols such that, for some positive integer $n \geq \lceil \log_q e \rceil$, the subwords composed by n consecutive symbols from the sequence are all different from each other. The approach pursued here consists in extending the well known case of sequences generated by maximal LFSRs to the general case. Although this method will not, in general, yield a sequence with optimal n, it will be proven that it is indeed optimal in a relaxed sense: n will be the smallest possible among those generated by linear methods.

For later reference, let us state formally the problem that needs to be solved in order to build a single-tracked shaft encoder of length e using q-ary detectors.

Problem 1: Given two positive integers e and q, find a circular sequence of length e over an alphabet of q symbols such that all words composed of n consecutive symbols from the sequence are different from each other (for some $n \geq \lceil \log_q e \rceil$ as small as possible).

B. Fibonacci LFSRs

A Fibonacci Linear Feedback Shift Register (FLFSR) is a shift register whose input bit is computed as a linear combination of its state bits (see Fig. 2). These circuits can be used to generate sequences by concatenating the value of the output bit u_0 at each successive clock cycle. Here we will consider LFSRs as abstract machines working over $\mathbb{F}_q$. Any result can be applied to digital LFSRs by declaring $q = 2$.

The generated sequence only depends on the feedback logic and the initial state of the shift register (also known as the *seed*):

$$(u_0, \ldots, u_{n-1}), (a_0, \ldots, a_{n-1}) \in \mathbb{F}_q^n,$$

with $a_0 \neq 0$, for a FLFSR of size n. These sequences are periodic.

In a sequence generated by a FLFSR of size n, each word composed by n consecutive bits from the sequence represent the state of the shift register at the time cycle at which the word's first bit was outputed. If the sequence has period e, this implies that taking e consecutive words of length n from the sequece all of them are different. This is the fundamental property that makes sequences generated by FLFSRs useful to label single-tracked shaft encoders.

Given a FLFSR with a fixed feedback logic and changing the seed, one can generate different sequences with different periods. The set of all these different periods is called the *cyclic structure* of that FLFSR. Note that, since any sequence of period e generated by a FLFSR corresponds to a sequence of states of perdiod e, the cyclic structure of the sequences generated by a FLFSR coincides with the cyclic structure of the state sequences of the same FLFSR. We say that two seeds generate the same sequences if both sequences are equal as circular sequences. A *maximal FLFSR* of size n is a FLFSR whose cyclic structure is $\mathcal{CS} = \{1, q^n - 1\}$. The existence of

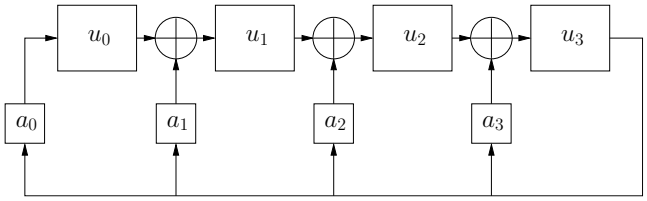

Fig. 3. LFSR of size 4 with Galois architecture

such FLFSRs for any positive integer n is a well known fact [10].

C. Galois LFSRs

A Galois Linear Feedback Shift Register (GLFSR) is a shift register where the value of each of the state bits is computed at each clock cycle as a linear combination of the register's output bit and the immediately preceding state bit (see Fig. 3). A FLFSR and a GLFSR with the same feedback logic coefficients (note the reversed order in the figures) share some properties and have the same *connection polynomial* defined as $a(x) = x^n - (a_{n-1}x^{n-1} + \cdots + a_1 x + a_0)$. However, it is worth to note that there exists one remarkable difference between both classes of LFSRs: given a GLFSR with fixed feedback logic the cyclic structures of its generated sequences and its state sequences do not coincide in general. We will define the cyclic structure of a GLFSR as the cyclic structure of its state sequences and the following holds.

Proposition 1: A FLSFR and a GLFSR with the same connection polynomial $a(x)$ have the same cyclic structure.

We will use the symbol $\mathcal{CS}(a(x))$ to denote the cyclic structure of any LFSR with connection polynomial $a(x)$. Using this notation we can state the following problem whose solution yields a solution to Problem 1 with the minimal n that can be achieved by the FLFSR approach.

Problem 2: Given e and $q = p^m$, find a connection polynomial $a(x) \in \mathbb{F}_q[X]$ with minimum degree n such that $e \in \mathcal{CS}(a(x))$ and a seed $u \in \mathbb{F}_q^n$ generating a sequence of length e.

A widely known observation states that $(0, \ldots, 0, 1)$ and $(1, 0, \ldots, 0)$ as respective seeds for a FLFSR and GLFSR with the same connection polynomial have the same period. These seeds will play a special role in the following section.

III. LFSR SEQUENCES WITH PRESCRIBED LENGTH

The state sequence of a GLFSR with connection polynomial $a(x)$ can also be represented in polynomial notation. Let the polynomial $u(x) = u_{n-1}x^{n-1} + \cdots + u_1 x + u_0 \in \mathbb{F}_q[X]$ represent the state of a GLFSR with connection polynomial $a(x)$ at a certain clock cycle, then the state at the next clock cycle is $u(x)x \mod a(x)$. The period of the state sequence associated with a seed $u(x)$ is the smallest positive integer such that

$$u(x) \equiv u(x)x^e \mod a(x).$$

That is, the smallest integer such that $a(x)$ divides $u(x)(x^e - 1)$. When $u(x) = 1$ this number is known as the *order* of $a(x)$. Note that this corresponds to the special seed mentioned at the end of section II.

In [6] the cyclic structure of $a(x)$ was studied and explicitely computed in terms of the orders of its irreducible divisors. Here we recall that result without proof. Some of the notation used here is defined in the Appendix.

Proposition 2: Let $a(x) \in \mathbb{F}_q[X]$ be a connection polynomial, and consider its decomposition into different irreducible factors

$$a(x) = a_1(x)^{s_1} a_2(x)^{s_2} \cdots a_r(x)^{s_r}. \tag{1}$$

Let $e_i = \mathrm{ord}(a_i(x))$ for $i \in I = \{1, \ldots, r\}$. Then, the cyclic structure of $a(x)$ is

$$\mathcal{CS}(a(x)) = \{1\} \cup \left\{ p^t \operatorname*{lcm}_{i \in J} \{e_i\} \;\middle|\; \varnothing \neq J \subseteq I, 0 \leq t \leq \left\lceil \max_{j \in J} \{s_j\} \right\rceil \right\}.$$

Through a careful analysis of the formula for $\mathcal{CS}(a(x))$, a characterization of the possible solutions to Problem 2 can be obtained. This characterization will give us with a finite set of feasible solutions which can be explored algorithmically as will be shown in section IV.

In [6] it was proved that $\max \mathcal{CS}(a(x)) = \mathrm{ord}(a(x))$ and that if one has $e \in \mathcal{CS}(a(x))$ with $e \neq \mathrm{ord}(a(x))$, then there exists a divisor $b(x)$ of $a(x)$ such that $e = \mathrm{ord}(b(x))$ and $\deg(b(x)) < \deg(a(x))$. Therefore, in order to solve Problem 2, it suffices to look for polynomials of order e with minimum degree. In this case it is already known that the seed $u(x) = 1$ will generate a sequence of length e. Thus, the problem to be solved can be restated as follows.

Problem 3: Given e and $q = p^m$, find a polynomial $a(x) \in \mathbb{F}_q[X]$ with order e and minimum degree.

Note that, by (i) in Proposition 5, the order of any irreducible factor of $a(x)$ is coprime with p. Therefore, according to (iii) and (iv) in Proposition 5, if the desired order e is divisible by p, some of the irreducible factors of $a(x)$ must be raised to a power greater than one. From now on, in any expression of type (1) we will suppose that the polynomials are given in decreasing order of degree: $d_1 \geq d_2 \geq \cdots \geq d_r$, where $d_i = \deg(a_i(x))$.

Let $a(x)$ be a polynomial given in the form (1) with order $e = p^{\alpha_0} e_*$, where e_* is coprime with p. Now let $s' = \max_{i \in I}\{s_i\}$ and $s = p^{\lceil s' \rceil - 1} + 1 = p^{\alpha_0 - 1} + 1$ if $\alpha_0 > 0$ and $s = 1$ otherwise. It can be shown (see [6]) that $a'(x)$ and $a''(x)$, where

$$a'(x) = a_1(x) \cdots a_{r-1}(x) a_r(x)^s, \tag{2}$$

$$a''(x) = a_1(x) \cdots a_r(x)(x-1)^s, \tag{3}$$

are polynomials of order e and at least one of them is a polynomial of minimal degree with the same irreducible factors as $a(x)$. Furthermore, $\deg(a''(x)) < \deg(a'(x))$ if and only if $d_r > 1$ and $s > 1$. Note that if we let $E = \{e_1, \ldots, e_r\}$, then $\operatorname{lcm} E = e_*$.

This describes how, given a fixed order $e = p^{\alpha_0} e_*$ and a set of irreducible polynomials such that $\operatorname{lcm} E = e_*$, a polynomial of minimal degree and order e having all these

polynomials as irreducible factors must look like. The next step is to characterize which irreducible factors can achieve the minimal degree among those yielding order e, i.e. having $\operatorname{lcm} E = e_*$.

Recall that, by (ii) in Proposition 5, the degree of an irreducible polynomial of order e over $\mathbb{F}_q[X]$ is $\operatorname{ord}_e(q)$. Therefore, we can forget about polynomials and consider only sets E of integers satisfying $\operatorname{lcm} E = e_*$. We will call any such set $E = \{e_1, \ldots, e_r\}$ a *set of orders* for e, and define its degree $\deg E$ as the minimum degree of any polynomial of order e having $a_1(x), \ldots, a_r(x)$ as its irreducible factors, where $\operatorname{ord}(a_i(x)) = e_i$.

By the previous discussion we have the following expression for the degree of E:

$$\deg E = \begin{cases} \sum_{i=1}^{r} \operatorname{ord}_{e_i}(q) & \text{if } s = 1, \\ \sum_{i=1}^{r} \operatorname{ord}_{e_i}(q) + s - 1 + \varepsilon & \text{if } s > 1, \end{cases} \quad (4)$$

where $\varepsilon = 0$ if $\min\{\operatorname{ord}_{e_i}(q)\} = 1$ and $\varepsilon = 1$ otherwise.

A set of orders $E = \{e_1, \ldots, e_r\}$ for e will be called *irredundant* if it satisfies

(i) $e_i \neq 1$ for all i, and
(ii) $\gcd(e_i, e_j) = 1$ for all $i \neq j$.

If E is not irredundant we will call it *redundant*.

Proposition 3: If $E = \{e_1, \ldots, e_r\}$ is a redundant set of orders for e, then there exists a set E' of orders for e such that $\deg E' \leq \deg E$. Furthermore, E' can be chosen to be irredundant.

Proof: Suppose first that E violates (i) and let $E' = E \setminus \{1\}$. If $s = 1$ we have $\deg E - \deg E' = 1$, and if $s > 1$ we have $\deg E - \deg E' = 1 - \varepsilon' \geq 0$, where ε' appears in the expression for $\deg E'$. Now suppose that E satisfies (i) but violates (ii) and let $e_i, e_j \in E$ be such that $\gcd(e_i, e_j) = d > 1$. Let us consider first the case where d is equal to one of the orders, say $d = e_i$, and take $E' = E \setminus \{e_i\}$. If $s = 1$ we have $\deg E - \deg E' = \operatorname{ord}_{e_i}(q)$. Otherwise, $\deg E - \deg E' = \operatorname{ord}_{e_i}(q) + \varepsilon - \varepsilon' \geq 1$. Suppose now that $d \neq e_i, e_j$, let p_k be a prime in the factorization of e_* dividing d, and let β the maximum positive integer such that p_k^β divides d. Then one of the orders, say e_i, is divisible by p_k^β but not by $p_k^{\beta+1}$. Let $e_i' = e_i / p_k^\beta$ and take E' equal to E with e_i replaced by e_i'. Since e_i' divides e_i, by Lemma 4, we have that $\operatorname{ord}_{e_i'}(q)$ divides $\operatorname{ord}_{e_i}(q)$ and therefore $\operatorname{ord}_{e_i'}(q) \leq \operatorname{ord}_{e_i}(q)$. Hence, if $s = 1$ we have $\deg E - \deg E' = \operatorname{ord}_{e_i}(q) - \operatorname{ord}_{e_i'}(q) \geq 0$. Otherwise, if $s > 1$ we have $\deg E - \deg E' = \operatorname{ord}_{e_i}(q) - \operatorname{ord}_{e_i'}(q) + \varepsilon - \varepsilon' \geq 0$ because $\varepsilon = 0$ implies $\varepsilon' = 0$ (i.e. $\varepsilon \geq \varepsilon'$).

It is easy to check that for any of the sets E' we have considered $\operatorname{lcm} E' = \operatorname{lcm} E$. Therefore, we have constructed a set of orders for e with $\deg E' \leq \deg E$. If E' is still redundant, the whole process can be iterated until an irredundant one is found (at each stage, the quantity $r + e_1 + \cdots + e_r$ strictly decreases). ∎

IV. Sequence construction algorithm

Proposition 3 says that, in order to find a set of orders for e giving a polynomial of minimum degree, it is enough to search for this set among those being irredundant. If $e = p^{\alpha_0} p_1^{\alpha_1} \cdots p_t^{\alpha_t}$ and $E = \{e_1, \ldots, e_r\}$ is an irredundant set of orders for e, then each prime p_j in the factorization of e must divide one and only one order $e_i \in E$. Furthermore, if p_j divides e_i then $p_j^{\alpha_j}$ must divide e_i as well. Finally, since $\operatorname{lcm} E = p_1^{\alpha_1} \cdots p_t^{\alpha_t}$, no prime p' different from $p_1, \ldots, p_t$ can appear in the factorization of any order $e_i \in E$. Thus, we can conclude that every irredundant set of orders for e corresponds to a partition of the set $D = \{p_1^{\alpha_1}, \ldots, p_t^{\alpha_t}\}$. Using this correspondence, an algorithm exploring all possible irredundant sets of orders for a given e can be built. This algorithm is given in Fig. 4.

The algorithm receives as its input two factorized integers, $q = p^m$ and $e = p^{\alpha_0} p_1^{\alpha_1} \cdots p_t^{\alpha_t}$ with $m > 0$, $\alpha_0 \geq 0$ and $\alpha_i > 0$ for $i = 1, \ldots, t$, and returns a polynomial $a(x) \in \mathbb{F}_q[X]$ of order e and minimum degree. Using $a(x)$ as the connection polynomial of a FLFSR, the sequence generated using $(0, \ldots, 0, 1)$ as a seed will be a sequence of length e over an alphabet of q symbols such that all subwords of length $n = \deg(a(x))$ are different from each other. This is a solution to Problem 1.

The procedure $\mathsf{Irr}(e_i, q)$ appearing in line 35 of the algorithm returns an irreducible polynomial of order e_i in $\mathbb{F}_q[X]$. These polynomials can be tabulated or computed using the method given in Proposition 5 (ii). The rest of procedures (Ord, Lcm and Deg) used in the algorithm are self-explanatory.

A. Algorithm complexity analysis

Although a complete analysis of the algorithm's complexity is outside the scope of the present paper, a few things can be said to justify its efficiency, at least when compared to the brute-force approach of exploring all possible q^e sequences.

First of all, note that once the polynomial $a(x)$ is given, $\Omega(e)$ operations over $\mathbb{F}_q$ are needed to generate the desired sequence. Therefore, the whole complexity of Problem 1 is, at least, linear with the length of the sequence. Thus, from the point of view of complexity, it would be irrelevant to try to solve Problem 3 with cost lower than $\Theta(e)$.

Second, all operations and methods that appear in the algorithm have polynomial complexity on its input parameters, which in every case can be bounded by q and e. See [11] for details about the complexity of all the procedures involved. Therefore, the whole complexity of the algorithm in Fig. 4 will be a polynomial on q and e multiplied by the number of iterations of the main loop. The factorization of e and q are supposed to take place outside the algorithm for simplicity. Although integer factorization is in general a difficult problem, it can be considered fast for the typical application values considered here (since these values are far below the 512 bit numbers considered in cryptography applications, see [11]).

The delicate point is then the number of iterations of the main loop. This number will be equal to B_t, the t-th Bell number, which counts the number of different partitions of a set of t elements. The number B_t grows very fast with t and satisfies $B_t \leq 2^{t^2}$. However, we have $t = \nu(e_*)$, the number of different prime factors appearing in the factorization

if $\alpha_0 > 0$ **then** ▷ Compute s
  $s \leftarrow p^{\alpha_0 - 1} + 1$
else
  $s \leftarrow 1$
end if
if $t = 0$ **then** ▷ Special case where $e = p^{\alpha_0}$
  $a(x) \leftarrow (x-1)^s$
  return $a(x)$
end if
$D \leftarrow \{p_1^{\alpha_1}, \ldots, p_t^{\alpha_t}\}, n_{min} \leftarrow \infty$
for all $P \in \mathcal{P}(D)$ **do** ▷ Main loop
  if $s > 1$ **then**
    $n \leftarrow s, E \leftarrow \{1\}$ ▷ Suppose $\varepsilon = 1$
  else
    $n \leftarrow 0, E \leftarrow \varnothing$
  end if
  for all $D_i \in P$ **do** ▷ $D_i \subseteq D$
    $n_i \leftarrow 1, e_i \leftarrow 1$
    for all $d \in D_i$ **do**
      $o_d \leftarrow \mathsf{Ord}(d, q)$
      $n_i \leftarrow \mathsf{Lcm}(n_i, o_d)$ ▷ $n_i = \mathrm{lcm}_{d \in D_i}\{\mathrm{ord}_d(q)\}$
      $e_i \leftarrow e_i \cdot d$ ▷ $e_i = \prod_{d \in D_i} d$
    end for
    $n \leftarrow n + n_i, E \leftarrow E \cup \{e_i\}$
    if $n_i = 1 \wedge s > 1$ **then** ▷ Check if $\varepsilon = 0$
      $n \leftarrow n - 1, E \leftarrow E \setminus \{1\}$
    end if
  end for
  if $n < n_{min}$ **then**
    $n_{min} \leftarrow n, E_{min} \leftarrow E$
  end if
end for ▷ End of main loop
$a(x) \leftarrow 1, n_{min} \leftarrow \infty$
for all $e_i \in E_{min}$ **do** ▷ Compute $a(x)$
  $a_i(x) \leftarrow \mathsf{Irr}(e_i, q)$
  $n_i \leftarrow \mathsf{Deg}(a_i(x))$
  $a(x) \leftarrow a(x) \cdot a_i(x)$
  if $n_i < n_{min}$ **then**
    $a_{min}(x) \leftarrow a_i(x), n_{min} \leftarrow n_i$
  end if
end for
$a(x) \leftarrow a(x) \cdot a_{min}(x)^{s-1}$
return $a(x)$

Fig. 4. Algorithm to solve Problem 3

of e_*, and this number is known to behave, in average, like $\log\log e_*$ [12]. Although a rigorous proof have not been pursued at this time, some heuristic arguments point to the fact that, on average, the number $B_{\nu(e_*)}$ behaves mildly enough to make the algorithm efficient for typical application values.

Finally, the claims for theoretical average efficiency are supported by the results obtained experimentally. The main loop of the algorithm on Fig. 4 was implemented in Maple$^{\mathrm{TM}}$ 9.5 [13] and several experiments were conducted using a Pentium IV at 1.6 Ghz. For each pair of values q and

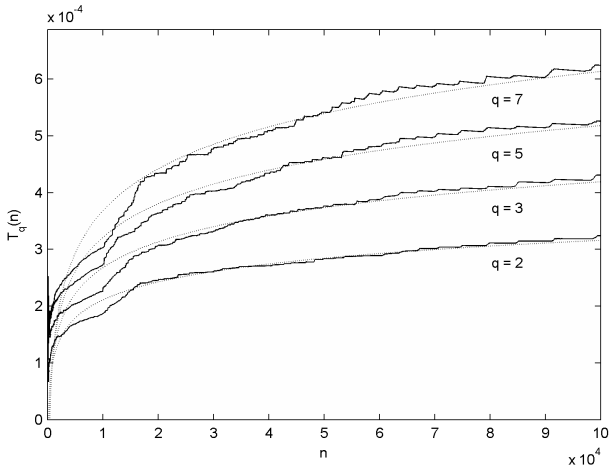

Fig. 5. Experimental results

e with $q \in \{2, 3, 5, 7\}$ and $100 \leq e \leq 100000$, the main loop was executed 20 times and the average $t_{q,e}$ of these times was computed. Then, for every integer $100 \leq n \leq 100000$, the average time taken to evaluate the main loop for every e such that $100 \leq e \leq n$, $T_q(n) = \mathrm{mean}\{t_{q,e} | 100 \leq e \leq n\}$, was plotted against n for each q. These correspond to the "noisy" curves in Fig. 5. Since these curves suggest a logarithmic average behaviour, each of the curves was fitted to a model of the form $A \log n + B$ with the method of least square errors. These are the smooth curves appearing in Fig. 5.

B. Application examples

Two examples coming from application problems are presented here.

First, consider the construction of a single-tracked shaft encoder with one degree of resolution using detectors capable of distinguishing between three different symbols. To build this encoder one needs to solve Problem 1 with $q = 3$ and $e = 360 = 3^2 2^3 5$. Executing the algorithm with these parameters we get $s = 4$ and $E_{min} = \{40\}$ which corresponds to the partition $\{\{8, 5\}\}$. The connection polynomial obtained for the FLFSR generating the desired sequence is

$$a(x) = x^8 + 2x^5 + x^4 + x^3 + x^2 + x + 1 \in \mathbb{F}_3[X].$$

Since this polynomial has degree 8 and $\lceil \log_3 360 \rceil = 6$, the encoder constructed this way will in principle use two extra detectors over the theoretical minimim. However, it could be the case that the property of subwords in the generated sequence being different from each other remains true for some $n_0 < 8$. In the present example this happens for $n_0 = 7$ (and not for $n_0 = 6$) and so the sequence generated can be used with only 7 detectors. In general, this possibility of extra reduction in the number of detectors depends non-linearly on the inputs of the problem and its study is out of the scope of this paper.

Another interesting problem would be the construction of a single-tracked shaft encoder of length $e = 12960$ using binary detectors. This length appears in the construction of

some system which has a resolution of one arcsecond (divide a circle in $360 \cdot 60 \cdot 60$ parts) and a set of gears with a gear ratio equal to 100 is used in the rotating device (for example [14]). Executing the algorithm with $e = 2^5 3^4 5$ gives $s = 17$ and $E_{min} = \{5, 81\}$. The connection polynomial obtained is

$$a(x) = (x^4+x^3+x^2+x+1)(x^{54}+x^{27}+1)(x-1)^{17} \in \mathbb{F}_2[X].$$

Note that $\deg(a(x)) = 75$ and $\lceil \log_2 12960 \rceil = 14$. In this case one can see that the best that can be done with linear methods does not always give degrees close to the theoretical minimum.

V. Conclusion

This paper addresses a combinatorial problem that appears in the design of single-tracked shaft encoders. It starts, in section II, by describing the origins of the problem and stating it formally. Then, by generalizing a well-known particular solution based on maximal LFSR sequences, the first combinatorial problem is reduced to an algebraic problem about polynomials over finite fields in section III. Finally, a finite set containing the solution to the later problem is identified and an algorithm conducting an exhaustive search in that set is presented in section IV. This gives a method for designing single-tracked shaft encoders with *any* desired resolution. The *efficiency* of the method for typical application values is theoretically justified and supported with experimental evidence.

The work presented here represents an extension to the results of the authors in [6]. The algorithm described here is a generalization of the one that appeared there. Furthermore, here a complete pseudo-code implementation is provided and a rough complexity analysis is given. The discussion starting from Proposition 2 (whose proof appeared in [6]) and culminating in the proof of Proposition 3 represents an improvement and clarification of some results from [6]. New application examples are also provided here.

Future works on this topic include a detailed account of the algorithm's complexity, including an estimation of the average behaviour of the quantity $B_{\nu(e_*)}$, and the exploration of nonlinear techniques to reduce the number of required detectors as pointed out in section IV. The actual construction of some prototypes and their experimental evaluation is another field for future work.

Appendix
Results about polynomials over finite fields

For every prime power q there exists a finite field with q elements denoted by $\mathbb{F}_q$. The ring of polynomials over this field is denoted by $\mathbb{F}_q[X]$. If $a(x) \in \mathbb{F}_q[X]$ is a monic polynomial not divisible by x, then its order, denoted by $\mathrm{ord}(a(x))$, is the smallest positive integer $e \geq 1$ such that $a(x)$ divides $x^e - 1$. Given two coprime positive integers a and b, the order of a modulo b, denoted by $\mathrm{ord}_b(a)$, is the smallest positive integer $i \geq 1$ such that b divides $a^i - 1$. For any prime p we define the notation $\lceil s \rceil_p = \lceil \log_p s \rceil$. When the prime p is obvious from the context we will simply use $\lceil s \rceil$. Using all this notation the following important results about polynomials over finite fields can be stated. For complete proofs see [15] and [6].

Lemma 4: If a, b and q are three integers such that q is coprime with a and b, then

$$\mathrm{ord}_{\mathrm{lcm}\{a,b\}}(q) = \mathrm{lcm}\{\mathrm{ord}_a(q), \mathrm{ord}_b(q)\}.$$

Proposition 5: Let $a(x), a_1(x), \ldots, a_r(x) \in \mathbb{F}_q[X]$ be monic polynomials not divisible by x. Then the following holds:

(i) If $a(x)$ has degree n then $\mathrm{ord}(a(x))$ divides $q^n - 1$ and therefore is coprime with p.
(ii) If $a(x)$ is irreducible and has order e then it is a divisor of $(x^e - 1)/\mathrm{lcm}_{d|e, d \neq e}\{x^d - 1\}$ and has degree $\deg(a(x)) = \mathrm{ord}_e(q)$.
(iii) For any integer $s \geq 1$ one has $\mathrm{ord}(a(x)^s) = p^{\lceil s \rceil} \mathrm{ord}(a(x))$.
(iv) If $a_1(x), \ldots, a_r(x)$ are pairwise coprime then $\mathrm{ord}(a_1(x) \cdots a_r(x)) = \mathrm{lcm}_{1 \leq i \leq r}\{\mathrm{ord}(a_i(x))\}$.

Acknowledgment

This work has received partial support by the Spanish Science and Technology Council CICYT project ref. DPI2007-61527 and by MEC (Spain) and the EFRD (EC) through project number MTM2006-13544.

References

[1] B. Arazi, "Position recovery using binary sequences," *Electronics Letters*, vol. 20, pp. 61–62, 1984.
[2] I. J. Good, "Normal recurring decimals," *J. London Math.*, vol. 21, pp. 167–172, 1946.
[3] E. M. Petriu, "Abolute-type pseudorandom shaft encoder with any desired resolution," *Electronics Letters*, vol. 21, pp. 215–216, 1985.
[4] ——, "Absolute position measurement using pseudo-random binary encoding," *IEEE Instrumentation and Measurement Magazine*, vol. 1, no. 3, pp. 19–23, 1998.
[5] G. H. Tomlinson, "Abolute-type shaft encoder using shift register sequences," *Electronics Letters*, vol. 23, pp. 398–400, 1987.
[6] J. M. Fuertes, B. Balle, and E. Ventura, "Absolute-type shaft encoding using LFSR sequences with a prescribed length," *IEEE Trans. Instrumentation and Measurement Magazine*, vol. 57, no. 5, 2008.
[7] J. M. Fuertes, S. de Miguel, R. Villà, J. Gonzalez, and J. Castro, "Dynamic analysis of a segmented telescope test bed," in *European Control Conference ECC'97 Proceedings*, 1997.
[8] J. Llacer, R. Jared, and J. Fuertes, "Analysis of the w.m. keck telescope primary mirror control loop," in *SPIE Symp. on Astronomical Telescopes & Instrumentation for the 21st. Century Proceedings*, 1990.
[9] A. Lempel, "m-ary closed sequences," *Journal of Combinatorial Theory*, vol. 10, pp. 253–258, 1971.
[10] S. W. Golomb, *Shift Register Sequences*. Aegean Park Press, 1981.
[11] J. von zur Gathen and J. Gerhard, *Modern computer algebra*. Cambridge University Press, 1999.
[12] G. Tenenbaum, *Introduction to analytic and probabilistic number theory*. Cambridge University Press, 1995.
[13] MapleSoft. Maple 9.5.
[14] R. Jared, A. Arthur, S. Andreae, A. Biocca, R. Cohen, J. Franck, J. Fuertes, O. Gabor, J. Llacer, T. Mast, J. Meng, T. Merrick, R. Minor, J. Nelson, M. Orayani, P. Salz, B. Shaefer, and C. Witebsky, "The w.m. keck telescope segmented primary mirror active control system," in *SPIE Symp. on Astronomical Telescopes & Instrumentation for the 21st. Century Proceedings*, 1990.
[15] R. Lidl and H. Niederreiter, *Finite Fields*. Cambridge University Press, 1997.