June 12, 2003
Submitted to IEEE Trans. on Information Theory

Low-Complexity LFSR Synchronization by Forward-Only Message Passing

Benjamin Vigoda, Justin Dauwels, Neil Gershenfeld, and Hans-Andrea Loeliger

Abstract—Gershenfeld and Grinstein showed that a modulated linear-feedback shift-register (LFSR) sequence can be synchronized by feeding the modulated sequence into an analog version of the LFSR. In this paper, a similar algorithm for synchronizing a LFSR sequence disturbed by additive white Gaussian noise is derived as forward-only message passing through the corresponding factor graph.

Keywords—LFSR synchronization, factor graphs, message passing.

1 Introduction

We consider the following problem (see Fig. 1). Let "$\oplus$" denote addition modulo 2. For fixed integers ℓ and m satisfying $1 \leq \ell < m$, let

$$X \triangleq X_{-m+1}, \ldots, X_{-1}, X_0, X_1, X_2, \ldots \tag{1}$$

be a binary sequence satisfying the recursion

$$X_k = X_{k-\ell} \oplus X_{k-m} \tag{2}$$

for all $k > 0$. Any such sequence will be called a LFSR (linear-feedback shift register) sequence. For $k \geq 0$, the m-tuple $[X]_k \triangleq (X_{k-m+1}, \ldots, X_{k-1}, X_k)$ will be called the *state* of X at time k. The sequence $X_1, X_2, \ldots$ is observed via a memoryless channel with transition probabilities $p(y_k|x_k)$. From the received sequence $Y_1, Y_2, \ldots, Y_n$, we wish to estimate the state $[X]_n$ of the transmitted sequence.

The situation is illustrated in Fig. 1 for $\ell = 1$ and $m = 3$; the boxes labelled "D" are unit-delay cells.

The computation of the maximum-likelihood (ML) estimate is straightforward and well known [1]; however, the complexity of this computation is proportional to $n2^m$,

B. Vigoda and N. Gershenfeld are with the MIT Center for Bits and Atoms, MIT, Cambridge, MA, 02139. Email: vigoda@media.mit.edu, neilg@cba.mit.edu.

J. Dauwels and H.-A. Loeliger are with the Dept. of Information Technology and Electrical Engineering, ETH, CH-8092 Zürich, Switzerland. Email: {dauwels, loeliger}@isi.ee.ethz.ch.

B. Vigoda and N. Gershenfeld wish to acknowledge the support by NSF grant CCR-0122419.

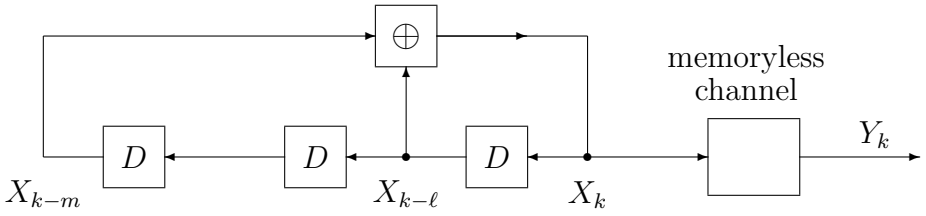

Figure 1: Linear feedback shift register (LFSR) sequence observed via a noisy channel.

which makes it impractical unless m is very small. In this paper, we propose a suboptimal algorithm, the complexity of which is independent of m. This algorithm, which is obtained in a straightforward way from the factor graph of the system, may be viewed as filtering the received sequence Y by a "soft" or "analog" (AFSR) version of the LFSR that generates X.

The algorithm may be of interest not only due to its potential practical value, but also

- as an example of a new detection algorithm derived from a graphical model
- as an example of nonlinear filtering as *forward-only* message passing
- due to its connections to another research topic, the synchronization (or "entrainment") of dynamical systems. In fact, the proposed algorithm is similar in structure to an algorithm proposed by Gershenfeld and Grinstein [4] for a different, but related problem.

In the examples, we will assume that the channel is defined by

$$Y_k = \tilde{X}_k + Z_k \tag{3}$$

with

$$\tilde{X}_k \triangleq \begin{cases} 1, & \text{if } X_k = 0 \\ -1, & \text{if } X_k = 1 \end{cases} \tag{4}$$

(i.e., binary antipodal signaling) and where $Z = Z_1, Z_2, \ldots$ is white Gaussian noise (i.e., independent zero-mean Gaussian random variables) with variance σ^2.

This paper is structured as follows. In Section 2, the maximum-likelihood solution is reviewed and interpreted as forward-only message passing through a cycle-free factor graph. In Section 3, the proposed new algorithm is derived as forward-only message passing through another factor graph. Some simulation results are given in Section 4, and some conclusions are offered in Section 5.

The description of the new algorithm in Section 3 can be read and implemented without worrying about its derivation from a factor graph.

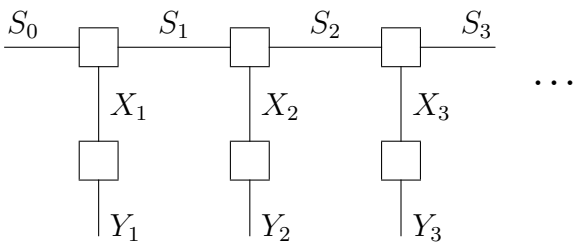

Figure 2: Forney-style factor graph (FFG) corresponding to the trellis of the system in Fig. 1.

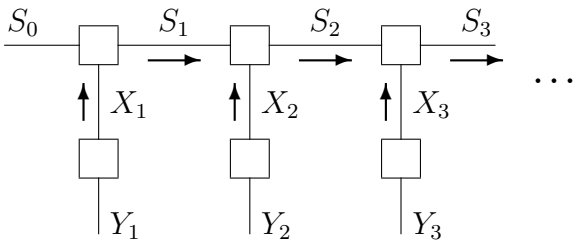

Figure 3: Forward-only message passing through the FFG of Fig. 2.

2 ML Estimation, Trellis, and Factor Graphs

We begin by recalling some obvious and well known facts. First, we note that the mapping $x \mapsto [x]_k$ (from sequences to states) is invertible for any $k \geq 0$: from the forward recursion (2) and the backward recursion $X_{k-m} = X_k + X_{k-\ell}$, the complete sequence x is determined by its state at any time k.

Second, we consider the maximum-likelihood (ML) estimate of $[X]_n$. Using the notation $y^n \stackrel{\triangle}{=} (y_1, \ldots, y_n)$ and $x^n \stackrel{\triangle}{=} (x_{-m+1}, \ldots, x_n)$, the ML estimate of $[X]_n$ is the maximum (over all possible states $[x]_n$) of the likelihood function

$$p(y^n \mid [x]_n) = p(y^n|x^n) \tag{5}$$

$$= \prod_{k=1}^{n} p(y_k|x_k). \tag{6}$$

For the channel (3), maximizing (6) amounts to maximizing the correlation between $\tilde{x}^n$ and y^n.

Due to the one-to-one correspondence between sequences and states (and assuming a uniform prior over initial states $[x]_0$), the ML estimate and the MAP estimate of $[x]_n$ coincide.

Third, we note that the computation of (6) may be viewed as the forward recursion of the BCJR algorithm [2] through the trellis of the system or—equivalently—as forward-only message passing through the corresponding factor graph. Let us consider this more

closely. Instead of factor graphs as in [5], we will use *Forney-style* factor graphs as in [3] and, e.g., [6]. A Forney-style factor graph (FFG) of our system is shown in Fig. 2. (Add a circle on each edge to obtain a factor graph as in [5]). The nodes in the top row of Fig. 2 represent $\{0,1\}$-valued functions $J(s_{k-1}, x_k, s_k)$ that indicate the allowed combinations of old state $s_{k-1} = [x]_{k-1}$, output symbol x_k, and new state $s_k = [x]_k$. The nodes in the bottom row of Fig. 2 represent the channel transition probabilities $p(y_k|x_k)$. As a whole, the FFG of Fig. 2 represents the function

$$p(y^n|x^n)J(x^n, s^n) = \prod_{k=1}^{n} J(s_{k-1}, x_k, s_k)\, p(y_k|x_k) \tag{7}$$

(defined for arbitrary binary sequences x^n), where $J(x^n, s^n) \triangleq \prod_{k=1}^{n} J(s_{k-1}, x_k, s_k)$ is the indicator function of valid LFSR sequences, which may also be viewed as a uniform prior over all valid x^n.

It then follows from basic factor graph theory [5] [3] that the MAP estimate of $S_n = [X]_n$ is obtained from forward-only message passing as illustrated in Fig. 3. Since the trellis has no merging paths, the sum-product rule [5] [3] for the computation of messages reduces to a product-only rule. By taking logarithms, the product-only rule becomes a sum-only rule, which amounts to a recursive computation of the correlation between $\tilde{x}^n$ and y^n.

3 The New Low-Complexity Algorithm

Another FFG for our system is shown (for $\ell = 1$ and $m = 3$) in Fig. 4. This FFG represents the function

$$p(y^n|x^n)J(x^n) = \prod_{k=1}^{n} \delta[x_k \oplus x_{k-\ell} \oplus x_{k-m}]\, p(y_k|x_k), \tag{8}$$

where $\delta[.]$ is the Kronecker delta and where $J(x^n) = \prod_{k=1}^{n} \delta[x_k \oplus x_{k-\ell} \oplus x_{k-m}]$ is the indicator function for valid LFSR sequences according to (2).

As this FFG has cycles, the standard sum-product algorithm becomes an iterative algorithm. However, the optimality of forward-only message passing in Fig. 3 suggests that forward-only message passing might do well also in Fig. 4. The resulting algorithm is a simple (non-iterative) recursion, which may be interpreted as running the received sequence Y through the "soft LFSR" circuit of Fig. 5. The quantities $\mu_{A,k}$, $\mu_{B,k}$, and μ_k in Fig. 5 are the messages indicated in Fig. 4. Note that the same message μ_k is sent along two edges out of the equality check node corresponding to X_k.

The computation of these messages (as indicated in Fig. 5) is a standard application of the sum-product rule [5] [3]. Each message represents "pseudo-probabilities" $\tilde{p}(0)$ and $\tilde{p}(1)$. For the sake of definiteness, we give the explicit computations for two (of several possible) representations of the messages as listed in [5, p. 512]; the two version are equivalent although the computations look different.

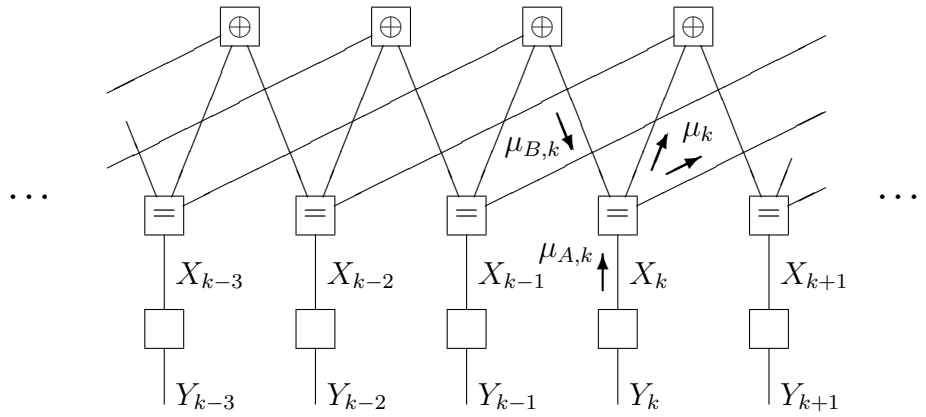

Figure 4: FFG corresponding directly to Fig. 1.

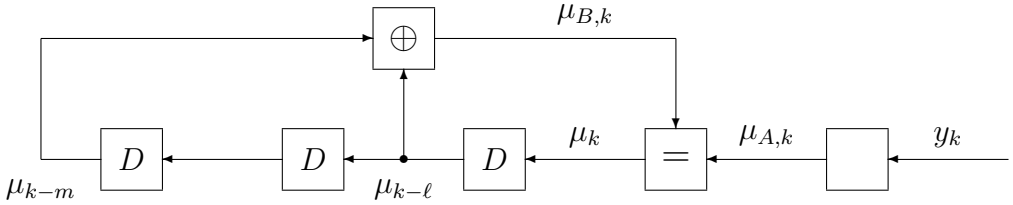

Figure 5: Computation of messages in Fig. 4 by a "soft LFSR".

Computations for likelihood ratio representation:

In this case, the messages represent the *ratio* $\tilde{p}(0)/\tilde{p}(1)$ of the pseudo-probabilities and are computed as follows.

Initialization: $\mu_k = 1$ for $k = -m+1, -m+2, \ldots, 0$.

Recursion (for $k = 1, 2, 3, \ldots$):

$$\mu_{A,k} = \frac{p(y_k|x_k = 0)}{p(y_k|x_k = 1)} \tag{9}$$

$$\overset{\text{for AWGN}}{=} \exp(2y_k/\sigma^2) \tag{10}$$

$$\mu_{B,k} = \frac{1 + \mu_{k-\ell} \cdot \mu_{k-m}}{\mu_{k-\ell} + \mu_{k-m}} \tag{11}$$

$$\mu_k = \mu_{A,k} \cdot \mu_{B,k} \tag{12}$$

Equation (9) holds for a general memoryless channel while (10) is the specialization to the channel specified at the end of Section 1. At any given time k, an estimate of X_k is

obtained as

$$\hat{X}_k \triangleq \begin{cases} 0, & \text{if } \mu_k \geq 1 \\ 1, & \text{if } \mu_k < 1 \end{cases} \tag{13}$$

and $[\hat{X}_k] = (\hat{X}_{k-m+1}, \ldots, \hat{X}_{k-1}, \hat{X}_k)$ is an estimate of the state $[X_k]$.

Computations for likelihood difference:

In this case, the messages represent the *difference* $\tilde{p}(0) - \tilde{p}(1)$ of the pseudo-probabilities and are computed as follows.

Initialization: $\mu_k = 0$ for $k = -m+1, -m+2, \ldots, 0$.

Recursion (for $k = 1, 2, 3, \ldots$):

$$\mu_{A,k} = \frac{p(y_k|x_k=0) - p(y_k|x_k=1)}{p(y_k|x_k=0) + p(y_k|x_k=1)} \tag{14}$$

$$\overset{\text{for AWGN}}{=} \frac{\exp(2y_k/\sigma^2) - 1}{\exp(2y_k/\sigma^2) + 1} \tag{15}$$

$$\mu_{B,k} = \mu_{k-\ell} \cdot \mu_{k-m} \tag{16}$$

$$\mu_k = \frac{\mu_{A,k} + \mu_{B,k}}{1 + \mu_{A,k} \cdot \mu_{B,k}} \tag{17}$$

At any given time k, an estimate of X_k is obtained as

$$\hat{X}_k \triangleq \begin{cases} 0, & \text{if } \mu_k \geq 0 \\ 1, & \text{if } \mu_k < 0 \end{cases} \tag{18}$$

and $[\hat{X}_k] = (\hat{X}_{k-m+1}, \ldots, \hat{X}_{k-1}, \hat{X}_k)$ is an estimate of the state $[X_k]$.

4 Simulation Results

The performance of the proposed algorithm was assessed by simulations, which are summarized in Figures 6–8. All these Figures show plots of the probability of synchronization

$$P_{\text{synch}}(k) \triangleq P\Big([\hat{X}_k] = [X_k]\Big) \tag{19}$$

vs. the time index k. Recall that σ^2 is the variance of the added Gaussian noise. All three Figures show plots for $\sigma = 0.6$, $\sigma = 1$, and $\sigma = 1.4$. All considered LFSR sequences are maximum-length sequences, i.e., their period is $2^m - 1$.

Fig. 6 shows simulation results for $\ell = 1$ and $m = 7$, and Fig. 7 shows simulation results for $\ell = 1$ and $m = 15$; in both cases, the performance of the maximum-likelihood state estimator is also shown. For each value of σ, the performance of the low-complexity

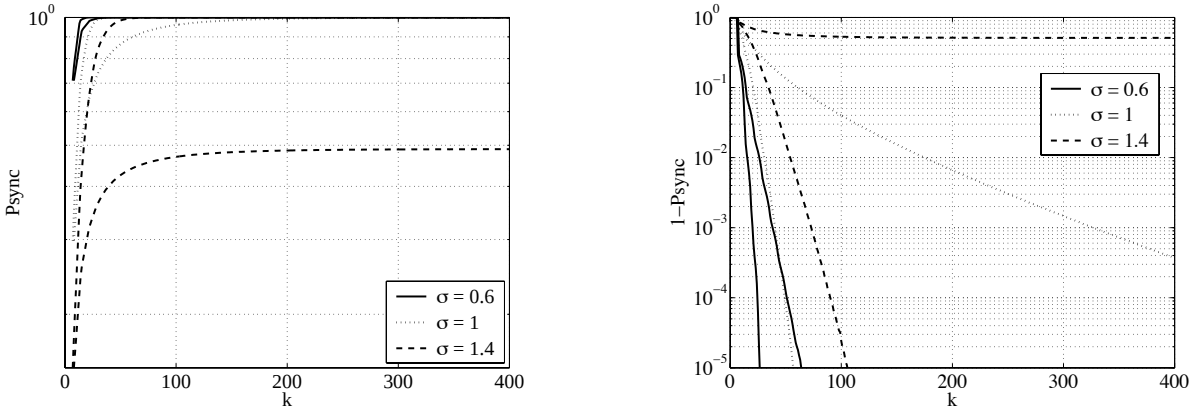

Figure 6: $P_{\text{sync}}(k)$ (left) and $1 - P_{\text{sync}}(k)$ (right) for $l = 1$ and $m = 7$, both ML and proposed algorithm.

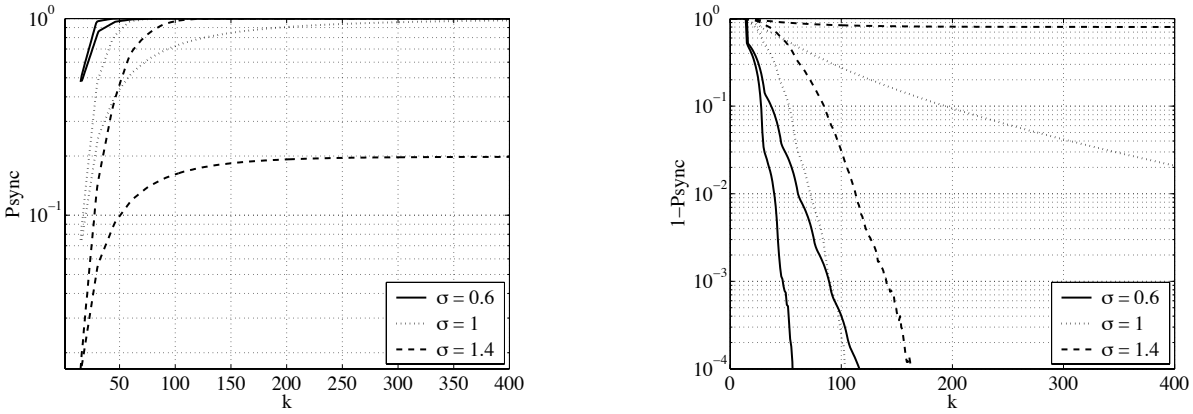

Figure 7: $P_{\text{sync}}(k)$ (left) and $1 - P_{\text{sync}}(k)$ (right) for $l = 1$ and $m = 15$, both ML and proposed algorithm.

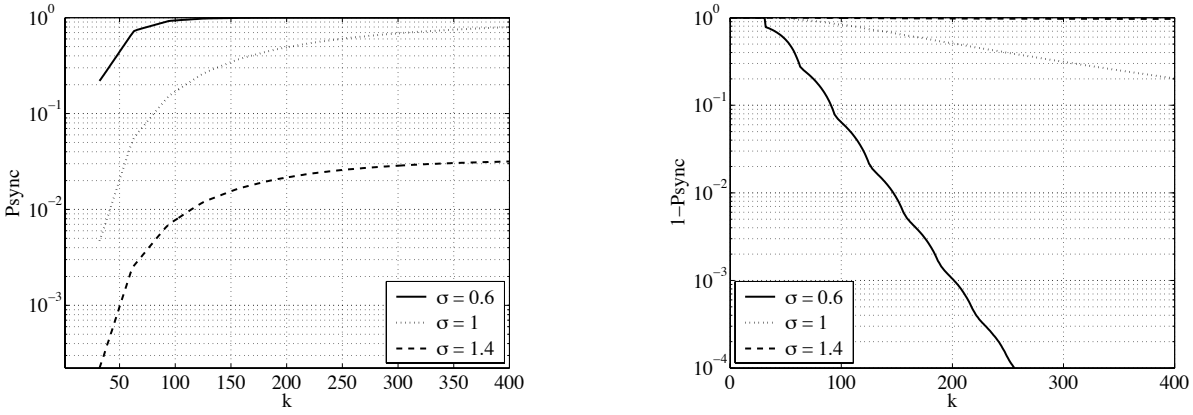

Figure 8: $P_{\text{sync}}(k)$ (left) and $1 - P_{\text{sync}}(k)$ (right) for $l = 3$ and $m = 31$.

algorithm and of the ML algorithm are plotted with the same line type. Fig. 8 shows simulation results for $\ell = 3$ and $m = 31$.

It can be seen from these figures that the proposed low-complexity algorithm works fine for sufficiently low noise power (up to about 0 dB), but fails to achieve synchronization for high noise power.

5 Concluding Remarks

The proposed algorithm is extremely simple and yet achieves synchronization if there is not too much noise.

The algorithm was derived in a straightforward manner from the factor graph of the system, which makes it easy to generalize the algorithm in various ways. In particular, generalizations in the following directions are easily accomplished: (i) linear LFSRs with more than two "taps"; (ii) linear LFSRs over other fields (or rings); (iii) nonlinear LFSRs; (iv) channels with memory.

It is interesting that the proposed algorithm may be viewed as running the noisy sequence through a "soft" or "analog" FSR, which connects our statistical approach to the phenomenon of "entrainment" of dynamical systems and, in particular, to prior work by Gershenfeld and Grinstein.

References

[1] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications,* vol. 3. Rockville, Maryland, USA: Computer Science Press, 1985.

[2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Information Theory,* vol. 20, pp. 284–287, March 1974.

[3] G. D. Forney, Jr., "Codes on graphs: normal realizations," *IEEE Trans. Information Theory,* vol. 47, no. 2, pp. 520–548, 2001.

[4] N. Gershenfeld and G. Grinstein, "Entrainment and communication with dissipative pseudorandom dynamics," *Physical Review Letters,* vol. 74, pp. 5024–5027, 1995.

[5] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Information Theory,* vol. 47, pp. 498–519, Feb. 2001.

[6] H.-A. Loeliger, "Least squares and Kalman filtering on Forney graphs," in *Codes, Graphs, and Systems,* (festschrift in honour of David Forney on the occasion of his 60^{th} birthday), R. E. Blahut and R. Koetter, eds., Kluwer, 2002, pp. 113–135.