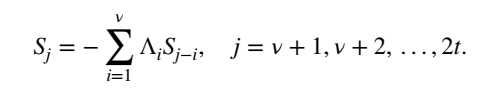
LFSR for BM Algorithm

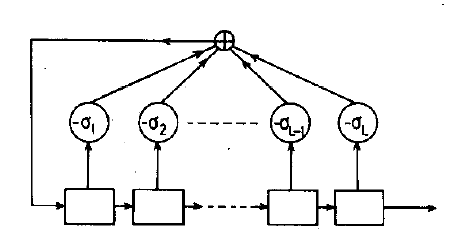
1. LFSR properties
2. Purpose of LFSR based Algorithm

The LFSR is based on the recursive equation below:



The purpose of this BM algorithm is to find shortest Λ sequence (construct a LFSR) to satisfied this equation (constrained by S).

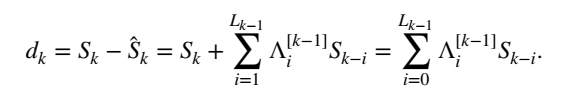
The Λ sequence can be found by recursive calculation (updating Λ).



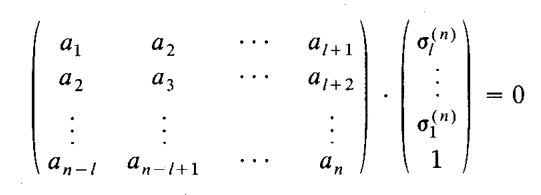
Find out the number and value of these coefficients

1. Recursive Calculation
2. Hankel Matrix

The recursive equation of LFSR based BMA can be transformed to the equation below:



The length should be as small as possible and such equation can be written as some kind of matrix multiplication:



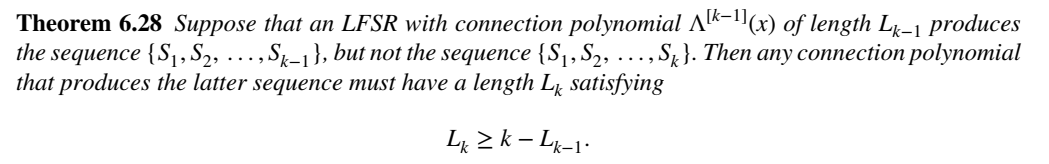
That is called Hankel matrix. Some methods can be applied to analyze this kind of matrix.

1. the Length of LFSR

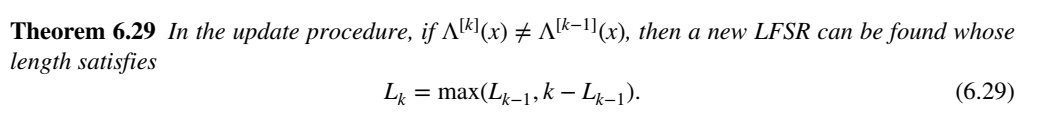
The Length of LFSR, denoted by L, can be updated under different conditions compared with k/2.

The updating logic of k (or n, r in some papers) is trying satisfying the constrain of every S.

There are 2 important theorems about updating the length of LFSR, see Todd K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*, section 6.43:



That is to say if the constrain of S sequence cannot be satisfied, there must be some relationship between L(k) and L(k-1). The proof of this theorem is a little complicated. It is proved with reduction to absurdity.



That is to say if the constrain of S sequence cannot be satisfied, how the length of LFSR can be updated. The proof is also complicated and is done with mathematical induction.

The change of L happens at k = L(k) + L(k+1) (where L(k) ≠ L(k+1)). Actually, L(k+1) is updated by Theorem 6.29 and it is unknown at this time. Then this change happens at L(k) < k/2. The above two conditions is equal in fact (See Section II of *A Simple Derivation of the Berlekatip-Massey Algorithm and some application*s, Kyoki Imamura and Wataru Yoshida, IEEE TIT, 1987. Those theorems are complicated and proved from the perspective of Hankel matrix). You can check below sequences to observe this conclusion:

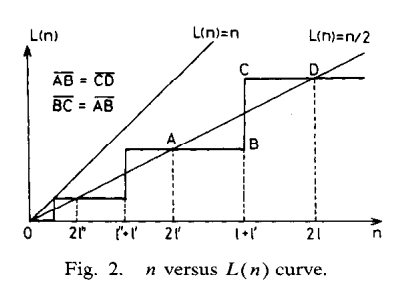
|  |  |  |
| --- | --- | --- |
| L | k/2 | k |
| 1 | 0.5 | 1 |
| 1 | 1 | 2 |
| 1 | 1.5 | 3 |
| 2 | 2 | 4 |
| 2 | 2.5 | 5 |
| 3 | 3 | 6 |
| 3 | 3.5 | 7 |
| 4 | 4 | 8 |

Sequence L(k) + L(k+1) are: 3, 5, 7.

Sequence k/2 are: 4, 6, 8.

The changes of the length of LFSR happen at those points.

This fact can be observed in pictures below (it is not easy to understand actually):



1. Λ Updating Method

The updating method for Λ can be seen in Section III of *A Simple Derivation of the Berlekatip-Massey Algorithm and some application*s, Kyoki Imamura and Wataru Yoshida, IEEE TIT, 1987.

We want to get Λ to satisfy:



We have, where the discrepancy  exists:



And we also have, when the recent change of the length of LFSR happens:



These equations above can be transformed to (padding zeros, notice these 2 equations is truly established, you can try some simple examples):



and



Then we have:



Therefore:



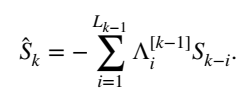
 also satisfies the constrain of S sequence ( here) just like . That is the we want. It can be calculated with ,, which is just produced, and , which is produced when the recent change of the length of LFSR happens.

Whether or not the length of LFSR changes, above calculation can be applied.

More details can be seen in Section III of *A Simple Derivation of the Berlekatip-Massey Algorithm and some application*s, Kyoki Imamura and Wataru Yoshida, IEEE TIT, 1987.

1. Implementation
2. The essence of LFSR based BMA

The equation below is just the LFSR form!



1. Something about Implementation

The storage required is very small because of the iteration architect. However, this architect limits the scale of parallelization.