ANR, Appel à projets Générique (AAPG 2019)

QCSP Project (ANR-19-CE25-0013-02)

Deliverable D2.1

Specification of the spreading sequence.

Editor:	Emmanuel Boutillon
Deliverable Nature:	Internal (scope: Consortium and ANR)
Due Date:	September the 30th
Delivery Date:	September the 30th
Version:	1.0
Total number of page:	19 pages
Keywords:	CCSK sequences, autocorrelation.

Abstract: This document gives the PN sequences found by a matlab program that tries to find better solution than LFSR based PN sequences construction.

List of authors

Partner:	Authors
UBS	Emmanuel Boutillon

History

V1.0: January the 24, 2020, Proposition of PN sequences.
V2.0: September the 9, 2020, Final deliverable.

Outline

1 Quality of a PN sequence 5
2 Comments on the LFRS sequence given in D1.1 5
3 Genetic algorithm to generate a PN sequence 6
4 PN sequences 9
PN sequence for q = 64 9
PN sequence for q = 128 10
PN sequence for q = 256 11
PN sequence for q = 512 12
PN sequence for q = 1024 13
PN sequence for q = 2048 15
PN sequence for q = 4086 17
5 Annex 20
PARETO SOLUTION FOR Q = 64 ..... **Erreur ! Signet non défini.**
PARETO SOLUTIONS OF SIZE Q = 128 ..... **Erreur ! Signet non défini.**
PARETO SOLUTIONS OF SIZE Q = 256 ..... **Erreur ! Signet non défini.**
PARETO SOLUTIONS OF SIZE Q = 512 ..... **Erreur ! Signet non défini.**
Pareto solutions of size q = 512 ..... **Erreur ! Signet non défini.**
Pareto solutions of size q = 1024 ..... **Erreur ! Signet non défini.**
Pareto solution of size q=2048 ..... **Erreur ! Signet non défini.**
Pareto solution of size q = 4048 ..... **Erreur ! Signet non défini.**

1 EXECUTIVE SUMMARY

The initial aim of this deliverable is, quoting the proposal:

"This task will be linked to the definition of spreading sequence that maximize the probability of good detection and synchronization. The impact of different type of spreading sequence will be study on the overall performance system. In particular, two questions will be addressed: should we add a cyclic prefix to the spreading sequence to mitigate the effect of channel echoes and help the synchronization? Should we add special side information that helps the synchronization algorithm? Should we use "classical" BPSK spreading sequence or something different? In fact, we have the freedom to generate any continuous based sequence $\mathbf{P}_0$. This property can be exploited to adapt the spreading sequence to the existing waveform used in 3GPP."

The scope of the document is a little narrower, since so far, no need for CRC or for additional information has been identified. Thus document explains how to generate better PN sequence than the ones constructed using LFSR. We give the PN sequences of size $q \in \{64, 128, 256, 512, 1024, 2048, 4196\}$ optimized thanks to a genetic algorithm. **Those sequences will be used as reference sequences in the rest of the project** to simulate QCSP frames.

2 QUALITY OF A PN SEQUENCE

A PN sequence of size q is denoted $P_0 = (P_0(0), P_0(1), \ldots, P_0(q\text{-}1))$, with $P_0(i) \in \{-1, 1\}$ for $i = 0, 1, \ldots, q\text{-}1$.

The autocorrelation function $\theta(k)$ of a sequence P_0 is given as:

$$\theta(\mathrm{k}) = \sum_{i=1}^{q-1} P_0(i)P_0(i+k),\ k = 0, 1, \ldots, q\text{-}1. \qquad (1)$$

where addition $(i + k)$ is done modulo q.

An ideal autocorrelation function should verify $\theta(0) = q$ (this is always true, see (1)) and $\theta(\mathrm{k}) = 0$, $k = 1, 2, \ldots, q\text{-}1$. In practice, except for particular cases, no known solution exists and the quality of a PN sequence can be measured on how the partial autocorrelation vector $(\theta(1), \theta(2), \ldots, \theta(q\text{-}1))$ is close to the zero vector. Three types of norms can be used to measure this distance: L_1, L_2 and $L_{+\infty}$ respectively defined as

$$L_1(\theta) = \frac{1}{q-1}\sum_{k=1}^{q-1}|\theta(k)|$$

$$L_2(\theta) = \frac{1}{q-1}\sum_{k=1}^{q-1}\theta(k)^2$$

and

$$L_{+\infty}(\theta) = \max\{|\theta(k)|, k = 1, \ldots, q-1\}.$$

3 COMMENTS ON THE LFRS SEQUENCE GIVEN IN D1.1

The following matlab code reproduces the method used to generate the LFSR sequence of length q = 64 given in D1.1.

```
clear L
L(1,:) = zeros(1,6);
L(1,1) = 1;

for i = 2:64
    L(i,:) = [mod(L(i-1,1)+L(i-1,2)+L(i-1,5)+L(i-1,6),2) L(i-1,1:5)];
end
L = 2*L -1; %to go from {0,1} values to {-1, 1} values.
```

From the 64 × 6 table L thus generated, it is possible to extract 6 PN sequences as Pi = L(:,i), i = 1,…,6.

P1 gives an unbalanced sequence (sum(p) not equal to 0) while the others sequence P2, P3, …, P6 are balanced. The resulting metrics are also different:

Results with sequence P1:
L2 = 16.750, L1 = 2.438, Linf = 12.000, sum = 2

Effective rate at -7 dB: 8.39937×10^{-2} (computed be averaging the entropy of 10^7 CCSK symbol transmissions).

Results with sequence P6:
L2 = 20.000, L1 = 3.250, Linf = 12.000, sum = 0
Effective rate at -7 dB: 8.39602×10^{-2} (computed be averaging the entropy of 10^7 CCSK symbol transmissions).

In conclusion, the unbalanced sequence has a slightly higher effective rate that the balanced on.

In the sequel, we use the unbalanced sequences proposed by V. Savin for the comparison with the optimized sequence.

4 GENETIC ALGORITHM TO GENERATE A PN SEQUENCE

The sequences are found by a MATLAB program that uses a genetic algorithm to optimize the L_2 norm. The explanation of the genetic algorithm is explicit in the MATLAB code and its comments given here after. Note that many others parameters can be used and that several attempts can be done to improve a result.

The matlab code used to generate the PN sequence is given here after.

```
%=========================================================================
% E. Boutillon
% 16/12/2019
% Search of good PN sequence of length q using L2 norm.
% Input: q:      size of the PN sequence.
%        best_l2: store the result in file “PN_sequence_q.txt” only
%    if a sequence with l2 distance bellow best_l2 is found.
%
% Output: PN : best found PN sequence of the round.
%         L2 : norm 2 value associated with PN.
%
% Moreover, if a sequence with smaller L2 norm than best_L2 is found, the PN
% sequence and its associated L2 norms is stored in the file
% “PN_sequence_q.txt” in append mode.
%
% It allows an easy implementation of several attempts:
%
%        best_l2 = 10^10;
%        for k = 1:50
%            [PN, L2(k)] = find_sequence(q, best_L2);
%            if L2(k) < best_L2
%               best_L2 = L2(k);
%            end
%        end;
%=========================================================================
function Generation_PN_final(q, best_L2)

%=========================================================================
% Genetic algorithm parameters:
```

```
%=====================================================================
Nb_of_gen  = 4000; % Number of generation (q>1024, 10^6 is a good choice).
Pop_size   = 50;   % Size of the population at the end of each
                   % generation.

Pop_growth = 50;   % size of new generation made by random association
             % of the old generation. A new element is created as
             % pn_new = [pn_old(a, 1:u) pn_old(b, u+1:end)]
             % with “a” et “b” taken randomly as well as crossing
             % point “u”.

% ============================================
% Random selection of the initial population.
% ============================================
pn = sign(randn(Pop_size,q));

for i = 1 : Pop_size
  pn(i, :) = zero_sum(pn(i, :)); % the function zero_sum modifies the first
                         % elements of pn so that there is an equal
                         % number of -1 and 1 (and thus, a zero sum).
end

% ============================================
% Loop on number of generations.
% ============================================
for gen = 1:Nb_of_gen

  %==================================
  % Creation of next generation.
  %==================================
  u = 2 + floor(rand(Pop_growth, 1)*(q-2));
  a = 1 + floor( rand(1, Pop_growth)* Pop_size);
  b = 1 + mod( a + 1 + floor( rand(1,Pop_growth)*( Pop_size -1)), Pop_size);
  % Note: by construction, a and b are different and between 1 and Pop_size.

  for j = 1:Pop_growth
    k = j + Pop_size;
    pn(k,:) = [pn(a(j), 1:u(j)) pn(b(j), u(j)+1:end)];
    pn(k, :) = zero_sum(pn(k, :));
  end

  %==========================================
  % Selection of the best “Pop_size” sequence.
  %==========================================
  for j = 1: (Pop_size + Pop_growth)

    % Autocorrelation computation
    theta = ifft( fft(pn(j,:)).*conj(fft(pn(j,:))));
    L2(j) = sum(theta(2:end).^2)/(q-1);
  end

   % The best are selected.
   [v I] = sort(L2);
   pn = pn(I(1:Pop_size),:);
```

```
    %=====================================================
    % Random rotations to give diversity of the crossing of the next
    % generation
    %=====================================================
    R = 1 + floor( rand(1,Pop_size)*(q-1));
    for i = 1:Pop_size
      pn(i,:) = [pn(i, R(i):end) pn(i, 1:R(i)-1)];
    end

    %=====================================================
    % Mutation of the current population
    %=====================================================
    pn( Pop_size/2+1:end, q/2+1:q) = sign(randn( Pop_size/2, q/2));

    %=====================================================
    % Test of the best candidate.
    %=====================================================
    if v(1) < best_L2
      best_L2 = v(1);
      P = pn(1,:);
      U = ifft( fft(P).*conj(fft(P)));
      L2 = sum(abs(U(2:end)).^2)/q;
      L1 = sum(abs(U(2:end)))/q;
      Linf = max(abs(U(2:end)));
      fprintf('Generation : %d, L2 = %2.3f, L1 = %2.3f, Linf = %2.3f\n', gen, L2, L1, Linf);
      file_name = strcat('resultq', string(q),'.txt');
      fid = fopen(file_name,'a');
      fprintf(fid,'L1 norm : %2.3f, L2 norm : %2.3f, Linf %2.3f\n',L1, L2, Linf);
      for i = 1:q
        fprintf(fid,'%1d',(pn(1,i)+1)/2);
      end
      fprintf(fid,'\n=======================\n');
      fclose(fid);
    end
end
end
%=======================================================
% Function P_o = zero_sum(P_i)
%=======================================================
function P_o = zero_sum(P_i)

   P_o = P_i;
   S = sum(P_i)/2;
   if S < 0
    I = find( P_o == -1);
    P_o(I(1:S)) = 1;
   elseif S > 0
    I = find( P_o == 1);
    P_o(I(1:S)) = -1;
   end
end
```

Figure 1 shows example of evolution of the L2 norm as a function of the iteration number. The different colors correspond to different set of population size (10, 20, 40, 80). The size of the PN sequence is $q = 256$.

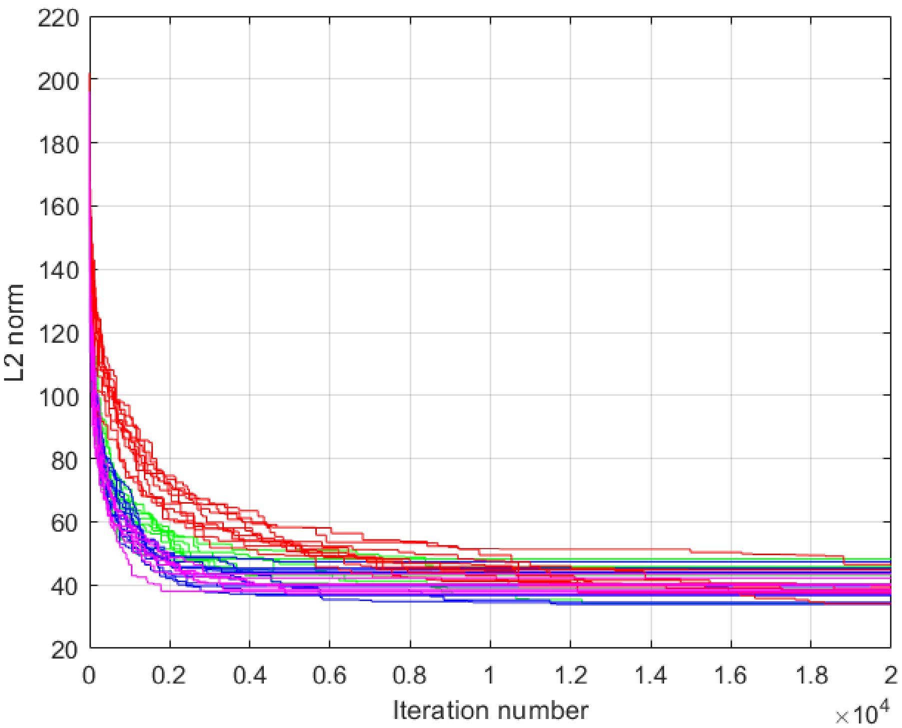


Figure 1: Evolution of the L_2 value according to the iteration number for several parameters for $q = 256$.

The following chapter gives the found sequences by the genetic algorithm and compare then with the LFSR sequences proposed in D1.1.

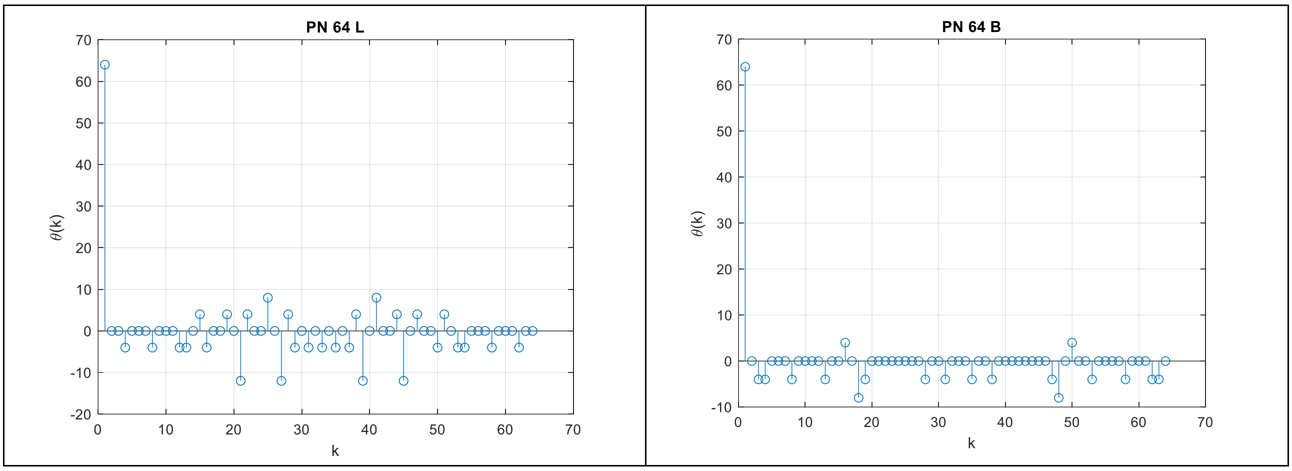
5 PN SEQUENCES

In this section, the result found for the PN sequence are given for $q = 64$ up to $q = 2048$. The table associated at each section gives the metric of 3 types of sequence: the one obtained by LFSR (PN_q_L), the one obtained by Genetic Algorithm (GA) with balanced number of -1 and +1 (PN_q_B) and, for information, the one obtained with GA algorithm without the balanced constraint.

The proposition is to use the sequence PN_q_B for all CCSK simulation. The 3 obtained sequences are given in the following order: PN_q_B (the one to be used) then, for information, PN_q_L and PN_q_U are also given in this order. The sequences are described are matlab string. From this string, it is easy to retrieve the PN sequence. The following code shows how to do that:
s = ‘1101’; (string)
p = s – ‘0’; (the operation are done with ASCII code => it gives a table [1 1 0 1];
p = 2*p – 1; (BPSK modulation to obtain an array [1 1 -1 1];

PN SEQUENCE FOR Q = 64

	L1	L2	Linf	sum	R_{eff} @ -7 dB
PN_64_L	2.44	16.75	12	2	0.08399
PN_64_B	1.25	6.00	8	0	0.08421
PN_64_U	1.50	6.00	4	-8	0.08375

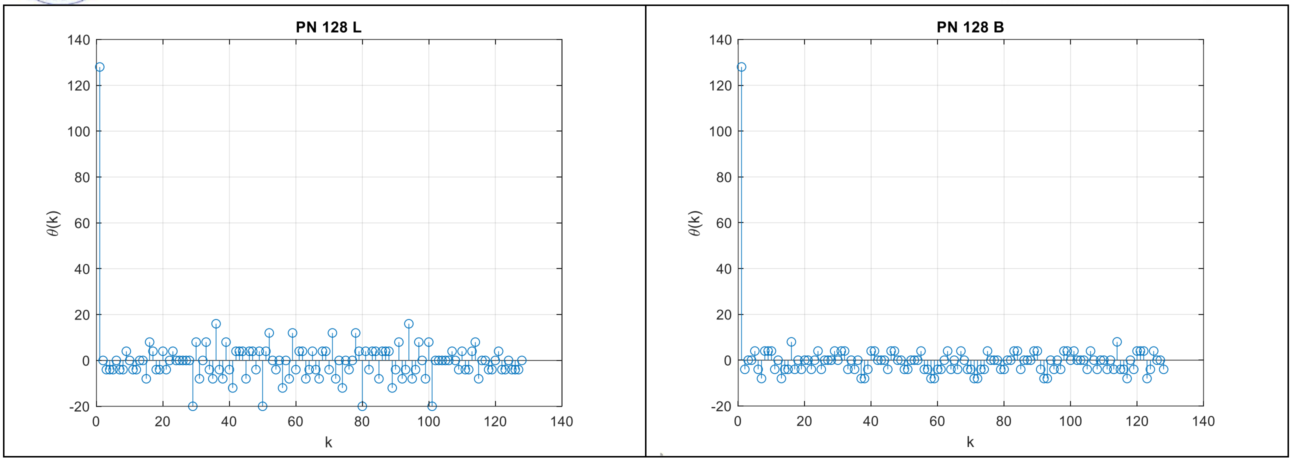


Sequence to be used (PN_64_B):
'0111011001011101011001110000010000101110000111011100100001101011';

For information:
PN_64_L
'1101110011000111010111111011010001000010110010101001001111000001';
PN_64_U
'1000101100010000001100111110100010111000010010110101100111000100';

PN SEQUENCE FOR Q = 128

	L1	L2	Linf	sum
PN_128_L	4.91	44.12	20	2
PN_128_B	3.00	15.50	8	0
PN_128_U	2.62	15.75	8	-8



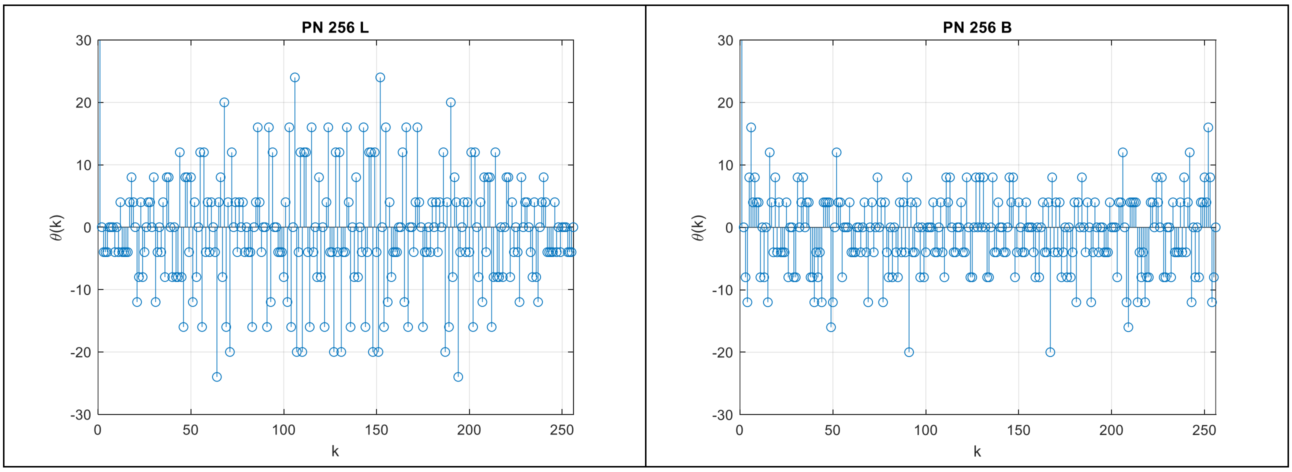
Sequence to be used (PN_128_B):
'10001010101000001101100110101101000001100011011001101100000101000011011110011100101010000111101001001110111111111000110111010100';

For information:

PN_128_L
'10000001001001101001111011100001111111000111011000101001011111010101000010110111110011100101011001100000110110101110100011001000 1'
PN_128_U
'10001000000110100001010001100111101110010011010000000110001101111010111000010100101110101100001011111001101001101100101001110100 0';

PN SEQUENCE FOR Q = 256

	L1	L2	Linf	sum
PN_256_L	6.92	82.93	24	2
PN_256_B	4.81	38.87	20	0
PN_256_U	4.33	32.19	12	-18

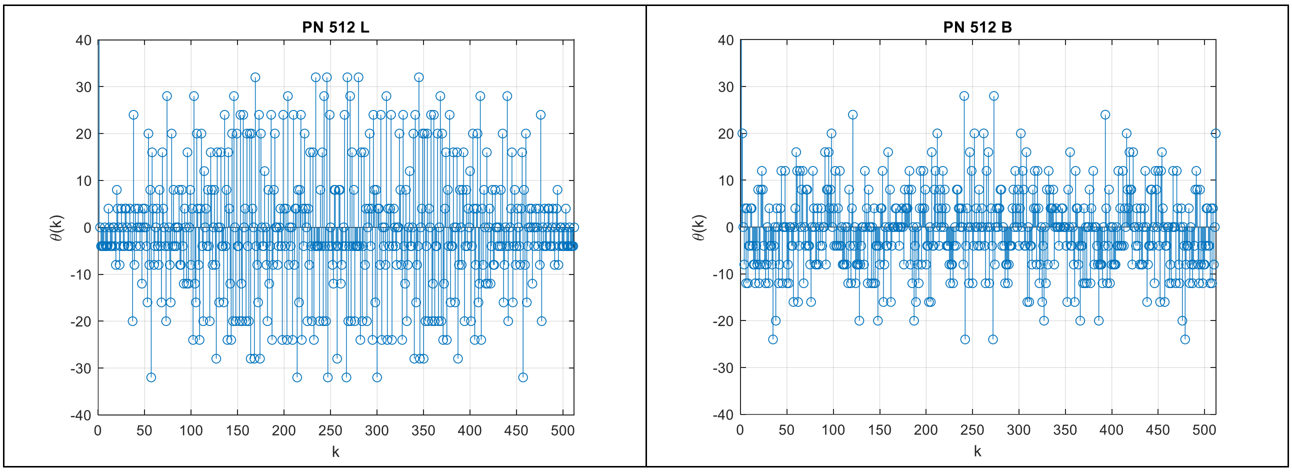


Sequence to be used (PN_256_B):
'0111010101010010010011011010111100100110001000010100100100011010101000010101011101101110000000110011001110000011001001011000111000100111100111001010111101001110011000101100001011000000100100111011110000110000111010100010111111101111111011110111100000001101110';

For information:
PN_256_L
'1000000010110001111010000111111110010000101001111101010101110000011000101011001100101111110111100110111011100101010010100010010110100011001110011110001101100001000101110101111011011111000011010011010110110101000001001110110010010011000000111010010001110001'
PN_256_U
'00100000011000001110010011100010000000110000000011101101101000111001010110100110101101110100111111100000001001011100100101011010101110101101100001010000000000111001001011101110011011100110110111110100001000011001110000101111100010000000100101110100101000110001111';

PN SEQUENCE FOR Q = 512

	L1	L2	Linf	sum
PN_512_L	10.05	181.16	32	2
PN_512_B	6.84	78.56	28	0
PN_512_U	6.37	67.50	20	-24



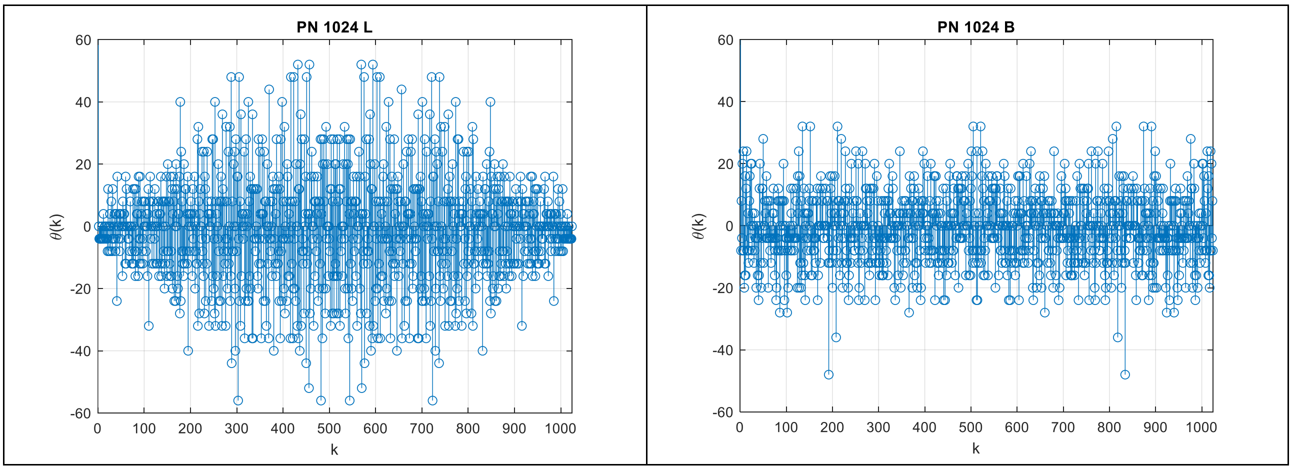
Sequence to be used (PN_512_B):
'10100000011001000000110001111111000110101001101010110100001011111100000011111010101101011000100010100000010101100000010110110110110001011101000110011010000000001011110111011001000000110111110111111100011101010011000100000101110110001110100101100100001111001001010111101000110100011100001111000111100110010101000101011110011000111011011001000001100010000100000011010010001011100111111111010101000111000111100111100110011100100101111011100100100

```
10011110010111001100011011110110011111101111011000000011111011001010100100
00';

For information:
PN_512_L
'1000000001000100011001000111010101101100011100010010101000110110011111100
1111000101101110010100100000100110011101000111110111100000111111111000011
1101110000101100110110111101000011100110000100100010101110101111001001011
1001110000000111011101001111010100101000000101010101111101011010000011011
0110110101100000101110111110001111001101001101011100011010001011111110100
1011000101001100011000000011001100101011001001111110110100100100110111111
0010110101000010100010011101100101111011000011010101001110010000110001000
01'
PN_512_U
'1110001111110010000000000010010000100100010000010101011000000100100000111
0111110011100110100101011001000011110100111110000111100010010011011011010
1100111010000000010101001100111100110000010101001001111110001100010010011
1101010011111110001110011010100110010010011110010100001100111011100100100
1110011111010001100110000000110001110100111101010100011011110101000000000
1001111000010110101011001110110010010111010001110000011100110111001001001
1011111000100101010101100011100110001100001010110110000100101010101001111
11';
```

PN SEQUENCE FOR Q = 1024

	L1	L2	Linf	Sum
PN_1024_L	13.31	312.58	56	2
PN_1024_B	9.37	142.66	48	0
PN_1024_U	9.73	148.97	32	-36



Sequence to be used (PN_1024_B):

```
'1000010010011010001111001000010010110101110010101101000101011111010100110
0000000111111011100011000110101110111110101100000000011100010110100100
```

```
1000101011000001000101101001000100100110011000111101000101011001100011101
0101101101111110001101110110011101100101111100110101000110101000010101110
0011111011100011111111101110000100101010100000110010010001110001110010101
1000111000011000110010000100110000000000001010100110111101111010010100111
0111010100101100011001110000011111001110000001100000100010110101011010001
1010000001011111111101100110110010110011101000101000111001011001010000101
1011011010010010010001110001110011010111001110100001010011010011100000110
0010101100001000101011111010110111000110001011001011111111000010111110100
0000100110111000010100101010001110101001111111010011111101001001110110000
0000101111111110100111101010111100010011011011111010010011001101110010101
1001101000101011001011100001111010011101101101100101000110010111000011011
1001001111111100011011001011001000100010110111011100111010100000111010000
001'
```

For information:

PN_1024_L

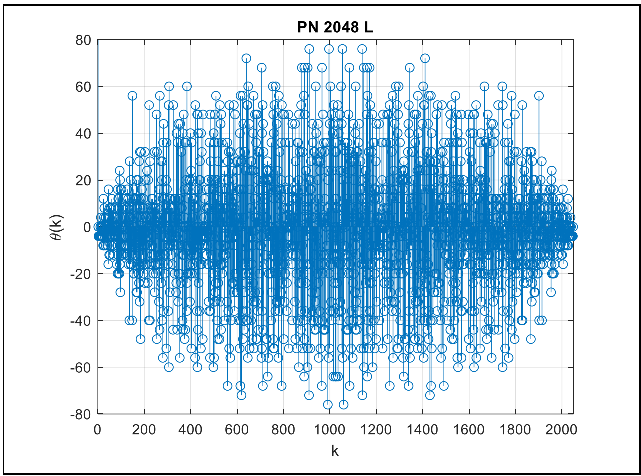
```
'1000000000100100100110100110101111100110001111100100011101111111000011100
0000011111111110001110001001110110010101110111101010001111010010101000001
0111111110101010101111010000111010010001100101101011001111010110001100111
1110010101010011001100101001111101001110000100011011001000101001101111011
1010101110011001110111011100111010100111010000011110110111000011000100101
0010110011010001000101101001011101001100010110000000101001001011111011100
0110001101110110000111100100111001011000100001101111111001110001101010010
1000010000100101101111101011100010111001000011111011010101000101111011001
1100111110000011100100101011001011110010111000001010110110011000011010110
1110100010101111110100011100110111001010001101000000110010010001000001001
1011010011110011010101100001011101101000110000100111111101110001111000000
1110110110001010001001100100000110100100111101111100010101011010000101000
0000101101101111001111000100011111101100011101011010100001100110110000011
0000000011011011010111010111100001010100100001011001001100000100010010000
001'
```

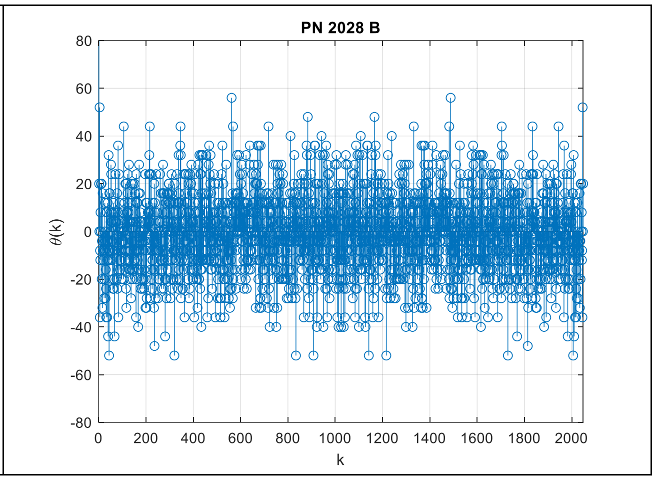
PN_1024_U

```
'0101011100110100101010001000111110000101101000010100000000100110000011011
0011101011011101000010010000010110110010010011111111011101100111111010110
0000011001100011001100101100110100011110111011101000111110000010001001111
1110001010000001010000101001011000000100110100110001100110010101111101001101
0101010101010101010010011111001111010010011011111101001010100000000010101100
0001010110000101000111111110000101101000011111110101110001100010110111010
0010001110001110100111010011011110001000010111011000100100110100100101101
0001101100011000111000000110101111001010001011011100110111111010100110001
0100001101001100111001000101000110001011010001111110101000000111111111010
0101110001100010011100001100001101010000100010101100010100111010011110000
1111111000011011010110011111010011000010010011001001101011111101110010110
1001010110011010011011111010000110000010001000100110111111100100001000100
0110101100001111100000001101011001110010010100000110111011001101000011000
1100101000000000100111110000110100000110001001110000011000100101111011100
011';
```

PN SEQUENCE FOR Q = 2048

	L1	L2	Linf	sum
PN_2048_L	19.84	701.26	76	2
PN_2048_B	13.60	295.45	56	0
PN_2048_U	13.43	287.56	68	-8





Sequence to be used (PN_2048_B):

'0001010100000110110110101100011011111010110111110101111111111010001011001
0110000101000111101100000110010001111010010100110010010011010111100000101
1110111001001111011000111000100001000111011111011001010110111110110000101
00000011100111000110111010001000110100000010011101000111100100110111000100
0100100110100100011111111101101111001011001011000000011000110110011011010
01110001011000000100000001011000100101001101111001011100101101110000010110
1001110100011011110110000000111001010111011001100001011001010001011011011
0011011101011111011001111011000100011010100110100110011011000101011110111
11110000001111000110100100000000100011001101011111110101100011011101011110
11111111110110010000011111111101000110010010010111011111110011100100110001
00001010001011101010110111111001001001101001010101101000000001111000011111
10110001100010100111110101101000100000011001111000000000110011110100010011
110101100011101011111100100110110000100000010101000000000111100011001001011
0101010111010000010110011001010001100101110101001100111011001011010010101
0001110100011011010100110010000010001000010001011100100110111001110010111
11011001111110000010000101011000000011010101101110101111001000110000010010
11111101100001100110011101000010110111111110100001010001011111011100111101
00101100110001101000000000110001111100010001101010101110101001010110100100
1011000000100101110010100010111100001101101010011100100110000011000101011
00101011010011100011110100000001110101001101111010000110111001111101101011
1101100001101010001111011011011000101010111101111101110101000011100101000
101101101010010110000000100001010011001000000010101000010000011010000101111
1100011100100101110011110110101101100001100000111011100001000100001111111
0000110100111001101101111101100010110101001000100101100011111000011011000
00100010110011001011000111011000001111101111110011110101111111001010110111
0001000101101101110011110000100000111110111000111111010001001111100101010

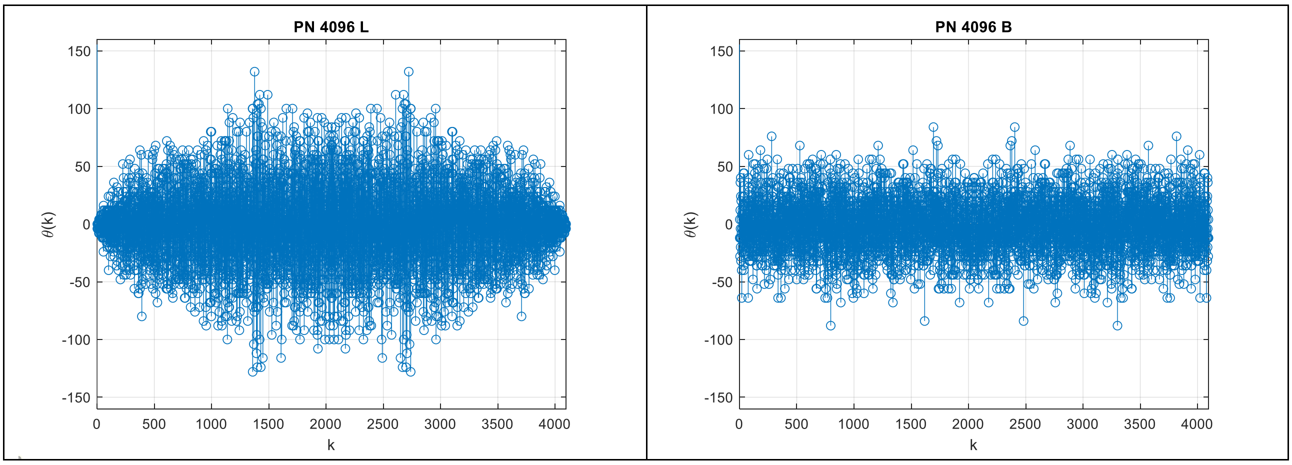
000100000011100110111001011000010000110001000111001111000111101000101101000
0010010101110011101000101100101000001011000001111000010110110111101000110
01011'

For information:
PN_2048_L
'10000000000101010101110111011100100111001011110010111001111011000011000
0111111000001100111111100001100110101001011011101000111001000000110100000
0111010101011000100010010100000101110101011100100010011110101111001110110
0101100011110110101100100100011110100000110001010100101000100001000001010
1111111011100110010011110000101100101011100001000110101000000111011111100
1001100111101001011001000010010111111010001100111010110100111011011110010
0100110100001011011111101101100110110110100100100100000101111010100011000
0000001111101010110011101110000110110000000111000000001101010101001000100
0111111010111001101110010110110000100100101011110100010011011111010010011
0001011110000001001101010111100010001100000101001010100010111011111011000
0010010100101111011110110010011011010000111000101011000000100011111110101
0011000100101101011110110111001110001101001111100010110011111011010011000
1110100101001110100010110001010001111101111100110110011110001111001111100
1111001101001101001000011101000000110111111110001100110000001111000000011
0000000011111111110011001100101101001011100010111001010001101000101001000
0001000001010100000001000101010111110111011001110010010110000101110000000
1111111101001100110111100001110011111100101100110100011110001010011010111
0100100011011110101101100100011100001010011111101111001100011000011111000
0001100101010110100010001110101111100100011001011111000010011000000101101
0101000111011101011000110111000001110010101001111011101001101100010110110
1011100011101100000110110101011011011101101101100011100011111001010011000
0100000111111010100110010001011010000010010000000101111111110110011001110
0011110010101100101110110100011011011111000111001101011000011101101010011
1000100000110101111110001001100101000011110111111001110011000011010010101
0111101110001100011010110101101110001001001111101000011001000000011110101
1001101110111100011011001010010010111010000100111010100001101111010100100
1101111010011100111010011110010000110000101011010100010010001010000101000
1010111010111011000100111000000101100000000100101010100001000100000001010000
00001'
PN_2048_U
'111100100000010100111001000000101001000000101000000000010111010011010011110
0111000010011111001101110000101101011001101101100101000011111010000100011
0110000100011000111011110111000100000100110101001000001001111010111110001
1000111001000101110111001011111101100000111000011011001000111011101101100
1011011100010000010010000110100110100111111100111001001100100101100011101
0011111011111110100111011010110010000110100011010010001111101001011000101
1100100000100111101100011010100010011001111000011010111010010010011001000
0100000010011000010011101110010101100101100110010011101010000100000001010
1000001110010110111111101110011001010001000101001110010001010000100000000
0100110111110000000001011100110110110100010000000110001101100111011110101
1010001010100100000011011011001011010101001011111110000111100000001001110
1110101100000010100101110111111100110000111111110011100010111011000010100
1000101000000110110010011101111110101011111111000011100110110100101010100
0101111101001100110101110011010001010110011000100010100101101010111000101

```
1100100101011001101111101110111101110100011011001000110001101000001001100
0000111110111101010011111100101010010001010000110111001111101101000000100
1111001100110001011110100100000001011110101110100000000011000010110100110
0111001011111111001110000011101110010101010001010110101001011011010011111
0011010001101011111000011110010010101100011011001111100111010100110101001
0110001110000101111110001010110010000101111001000110000010010100001001111
0010101110000100100100111010101010010000010001010111100001010111010010010
1011010011111101111010110011011111101010111101111100101111010000001110001
1011010001100001001010010011110011111000100011110111011110000000111100101
1000110010010000001001110100101011011101101001110000011110010011110111010
0110011010011100010011111000001000000110000101000000110101001000111011101
0010010001100001111011111000001000111000000101110110000011010101111011111
0001100100011010011110111100111011100011000001110001011101001011110110101
0001100010111110011010111110100111110000111101001001000010111001101001110
10101';
```

PN SEQUENCE FOR Q = 4096

	L1	L2	Linf	Sum
PN_4096_L	28.15	1337.36	132	2
PN_4096_B	18.92	565.89	88	0



Sequence to be used (PN_4096_B):

```
'0111110100001100100001101101110000111001010100111000001010111110010100010
0110010001010100101000111101110101100000111100111010110110100000110100001
0110100100110110111111001101001010000010011010100111010111110011110000010
1010000000101000000111101001011011000010101100001001000111110101110111011
1101010101011010010100101101010001010010101001100101000010100110010111111
1111001000001011011100001100010011011110101011010110011010110110011110010
0000110001100100011101001010010101100011001110110100010111101111001011101
0010111010100111000110111110110101000111011100010111110000010011110001111
0110011111111011100110010111001010110100111111010101111010101010001010101
0011000101000101000000001011110110010010111101010100100111101011111011001
0011100001010000011101001011000001010010011101100111111110000110000101011
```

```
0110101010110001110110011110101111001101100110100110011111101001101001111
1011101010001000111010001000110011110001011011001011011000101101111101100
0110100011100011101010101001011110110000000001101011000111100111111010010
1100010001001111010001010101100100011010110100110001001010001100100011000
0110001100101001000111110001000100010000011000100110011001001100110010110
0110111010001010010001101101111110011110100111111010111001001101111010001
0111000000111000000110000100100111101001001001110110000111011011010001010
0100100000111000100000110010100111111010001010111011101111110111010110111
1110000001111101001010111011110011111101111010000000000100001000011000100
0000001110110001100110100110000110000110001000101011111000101001110011101
0011011001001100011000000111001000010110101111100110011101110011110000110
0100110010100100101101110111100001011010011101111110111101000011111011011
1000100100000011111000011001100111011110000101001100000010100000111110100
0110101001110001011000000010100010110111011100100001010011000000001111011
0101100000001011111000100111000011101011101101010000010000011010011100001
0111011111000100101101111110011001011000011000000100000010100101101000011
1111000101000011001110000111011001000000001101010110011101010010000001011
1011000111110001011111011000101111110000001011011100011010010010011110000
1011100001000111100111101001100100111000111001010010100110111010101111011
1100000010100011001001000011011011100001110110110001111110001011100111000
1011110100100001000000100100111111000010011110010111111111111001010110111
1100000100101001001011000110000100111110000001000000100100010110001001011
1010101000000000101101101100110000000110110110101101110001001001000010101
0000000100111111010100101000001001000000111111101001110001100101101110110
1000100000010010100111110101010110001100000010111111101110111111000000100
1101101110000110111100101000100110000000000100110111011010111000111111001
1011111010101110001000111010010100110101101110000011011000010110010101100
0110100011111011001010000101101011010011110001101011100100000010010101110
1101011110100001001000111010101001100010010001011101011111000000011110001
1101110110000101100110110101000001011011110000000001110010001010101100110
0001111001110001100100100011000000101111111001111001110111010000111111110
0110110100000000010011011100110111001110001001101000101100001100010010100
1001111011010011111111001110011001011011100010101010110111010011010101011
0101010111111111111010100100110111000000101111011000110001110010100100101
1011001111100111101010101101110101010111111011001111100011000110010001100
1011111001010101000011010101001100110001010001010111101001000011101011011
1011011011101110101110010110011001101010101111001011000110000011110011111
0111001010011100010001101100010110101011010001110001010101000000100110111
1000011110011101001001010010101000000111000010010110011001100001100000011
1000010000001111001100010010000110001110101100111000010100000011011001011
0101100011100110101100000011000000111111011111110010111001111001111011011
0101101000100100111111011000000001001101100001010010101011101000110000111
1110000001111101100011111100110011001111000000110101001100111100100011000
0000001110101001111010110011101111010110000001000100110011101101000001101
1101011100101011011001111100000010111011010101110010110111011110000001000
110011010';
```

For information:
PN_4096_L

'1000000000001000001101000010000110010111101000110110000011000100111101000
0000110101101001000101000110001000111011101001111011100000110111001010100
0011011111101000111110001110000110111011010111011101011111000110000010110
1011110110111000100101001011100010000100110100001100011111111111100000010
0111110000010001101011000010010000000100001110101100000000100110110001111000
1111011110000101101001110101101000000100101110011000001001010100111101010
0001011111011010010011010010110010101111011111100100110101101101000011100
1110010111100000111011000001000010101010101000000011101010000000111011001
0000011100000000111110100110111111111000100100001010001010010100000011011000
1011001001111111000110100010000000111001100010100110000001101010010011100001
0011100100011001010010101000111011001001001111101000101110010100000010110
1111110000011001100110000000001011001111111010110111000000101111111101010100
1110110101010111100011111001111001110011111101101111001000111010101000010
0111100000000101110111100111011111110110011100010111100010111000010001100001
1001111010010001110001010110000110100011000000011011010101111111011011101
1111111001000101010110010010111111100101010100110001011011001100101000001
0101110101110100010101001001010001111010011001111100010100000000110100101
0001011010101101110100001101011110010111110000100001000101001110000101111
0011011111011000010000000001001001100101010010110100101010101110000110000110
1110001101010100011001100010000110110111011100111110011010011010011110011
0001110011110101100010001010111100111111111011000110000110110011011010011
1110110010100011010100000111110000110001011111001010001001010110111110000
0000011100010001100111001001110011001101000000100010010111101100110000001000
1000011111011011010010111010100010011101100010100010000101110101100100001
1110011010111010101010010000100000100100010010011101010010001100001001100
0000101011111011110010101110011110001100100101011111111111010000111101101
0001110100010001001111110000111001010010001000000111100000101011011011000001
1010011000011101001000011000011111001011001000111110101110110010110011001
1100000111111000111100011101001100011110111110001000000001100101011010111100
0110100010100000100011100110101010101100000010100101100001011000011110001001
0100000101110011100001111110101111110011011011011110110001000001011000111
1101111101010001101110000101011001011011100100010001011111101010111101011
1101001110111101011010100001111111101110011011001011101100001100000111101
0001001100110010000001100011011111100100000010100001100111111001011101000000
1010100110101011101101101101011011110000001101111010100101011001111011010
0000110010001101101000111100001101010110011010110010100111010010100110110
0111110101010110100010010001101001000001000000010100111100010001000111111
0111101101010001111110011110111010101110010110000000110011101100110100100
1001001101101011111110110101100111001101110101101100000001000111101101011000
0101000000100110010001001101110011101001110011101101110101001100110110000001
1100001001110000001111111001110101111000011110101010010100100110000101010
0010101101001100101110010000000010001011011101100011100011101101100101010101
1011001000101110110100110110111111100001010010010000001100000011101110000001
1101101000101100010111010010010110011101110110111110100000111010000000101
0101110100110101111101011001100011000111011111001100001100100111011100101
1100000000111100100101101111010000101101101110011001001000111100101101100
0100100101010111110011100011111101111111101001011111010010011110010000110101
0011100011011101001011011010100111111010011111111100110010110001001101010100
1101110111101111010010101110111110111000101101000110100000000010100011100
1001010011111001001001011101110001111010111001001000011100011001101010001

```
0111101010110101010011100101011000111010111000001001110100001001011000110
1101100111010101100010101010001000110111100010011000100101101011010110001
1001011010100101111001001111011000000000011000010111000110001010111000111
0010110100001010011010001110000010111101110110011110010100110010011001110
1000110010000101100101111110001011000001110010000100100100010110011011110
0110011110000100101000011101111000110110010011011110111001001100011010111
000101001'
```

6 ANNEX

The annexe gives some useful matlab codes.

MATLAB CODE TO ANALYSE A SEQUENCE

```
%==============================================================
% E. Boutillon, 1/10/2020
% Fonction qui affiche les résultats d'une sequence PN
candidate.
%==============================================================
function [L1 L2 Linf p] = analyse_seq(a)

p = a - '0';
p = 2*p - 1;
q = length(p);

U = ifft( fft( p).* conj(fft(p)));
Linf = max( abs(U(2:end)));
L2 =  sum( U(2:end).^2)/q;
L1 =  sum( abs(U(2:end)))/q;

fprintf('L2 = %2.3f, L1 = %2.3f, Linf = %2.3f, sum = %d\n',
L2, L1, Linf, sum(p));
stem(U);
grid on;
xlabel('k');
ylabel('\theta(k)');
```

MATLAB CODE FOR THE GENERATION OF LFSR SEQUENCES.

```
%==============================================================
% E. Boutillon, 1/10/2020
% Generation of Extended LFSR sequence proposed in D1.1
```

```
%==============================================================
clear L
Pol = [1 0 0 1 1 1 0 0 0 0 0 0
       0 0 1 0 0 0 1 0 0 0 0 0
       0 1 1 1 0 0 0 1 0 0 0 0
       0 0 0 1 0 0 0 0 1 0 0 0
       0 0 1 0 0 0 0 0 0 1 0 0
       0 1 0 0 0 0 0 0 0 0 1 0
       0 0 0 0 0 1 1 0 1 0 0 1];

for p = 6:12
    q = 2^p;
    clear L;
    L(1,:) = zeros(1,12);
    L(1,p) = 1;
    for i = 2:q
        L(i,:) = [mod( L(i-1,:)*Pol(p-5,:)', 2) L(i-1,1:11)];
    end

    s = [];
    for i = 1:q
        s = sprintf('%s%d',s,L(i,p));
    end
    s
    figure(p)
    [L1 L2 Linf p] = analyse_seq(s);
    Title_string = sprintf('PN %d L',q);
    title(Title_string);
end
```