Synchronization of Boolean Dynamical Systems: a Spectral Characterization

Jeremy Parriaux, Philippe Guillot, Gilles Millérioux

Synchronization of Boolean Dynamical Systems: a Spectral Characterization

Jérémy Parriaux[1], Philippe Guillot[2], and Gilles Millérioux[1]

[1] Nancy University, CNRS,
Research Center for Automatic Control of Nancy (CRAN UMR 7039), France,
jeremy.parriaux@esstin.uhp-nancy.fr,
gilles.millerioux@esstin.uhp-nancy.fr,
[2] Université Paris 8,
Laboratoire Analyse, Géométrie et Applications (LAGA UMR 7539), France
philippe.guillot@univ-paris8.fr

Abstract. In this paper a spectral characterization of the synchronization property of Boolean dynamical systems is provided. Conditions on the spectrum of the next-state function are derived for two systems coupled in a unidirectional way - also called master-slave configuration - to guarantee self-synchronization. Two kinds of self-synchronization are discussed: the statistical one and the finite one. Next, some conditions are stated for a specific input sequence to allow the system to be self-synchronizing. Some of the results are based on the notion of influence of variables, a notion that is extended to vectorial Boolean functions for the purpose of the paper. A potential application to cryptography is finally given.

1 Introduction

Dynamical systems are commonly used to model natural or engineering based processes. We can distinguish two kinds of systems. The continuous ones, $\mathbb{R}$ valued and discrete ones, finite-set valued. The latter can be either an approximation of a continuous system or can be intrinsically discrete. Let us stress that the terminology *continuous* or *discrete* refers to the state variables of the system regardless the time which can be continuous or discrete. Among a wide variety of discrete dynamical systems, the class of Boolean Dynamical Systems (BDS for short) is of special interest. In this paper, we focus on non-autonomous BDS, that is with input. The specificity of BDS lies in that the internal state, the input and the output are Boolean variables and therefore the transition and output functions are Boolean functions.

In this paper we deal with the issue of synchronization of BDS which is the process through which two systems are brought to the same state. Although several structural conditions to guarantee synchronization have been provided in the open literature, few works deal with BDS. Moreover, these studies address the synchronization issue in the time. In this paper we propose a spectral point of view. The interest of the spectral approach lies in that the composition of

functions can be expressed in terms of product of matrices well suited for design purpose. Spectral characterization is also well appropriate in the perspective of ensuring special cryptographic properties when the dynamical systems under consideration are involved in a ciphering setup.

More precisely we investigate the problem of self-synchronization. By self-synchronization, it is intended a dynamical behavior which do no longer depend on the initial condition after a transient time. Besides the spectral characterization, the novelty of the study lies in that the problem is viewed through the notion of influence of variables. Roughly speaking, influence describes the ability of a subset of the input variables of a function to change its output. Here the set of variables under consideration is the initial condition of the dynamical system. For the purpose of the paper we also had to extend this notion to vectorial Boolean functions.

The layout is the following. In Section 2, we recall some background on Boolean functions and tools of spectral analysis in particular Walsh transform. Section 3 is devoted to the problem statement, namely the issue of self-synchronization between two dynamical systems coupled in a unidirectional way. Distinction between statistical and finite time self-synchronization is made. Section 4 deals with the Walsh transform of the iterated function of a dynamical system as a prerequisite for deriving the main result. In Section 5, the notion of self-synchronizing sequence is developed. The main result of the paper is stated in Section 6 wherein, based on the notion of influence, we derive conditions on the spectrum of the next-state function for a BDS to be self-synchronizing. Finally Section 7 is devoted to illustrative examples. An example potentially interesting for cryptographic applications involving the so called Self-Synchronizing Stream Ciphers (SSSC for short) is provided.

2 Preliminaries and Definitions

In this section, we recall the basics about spectral analysis of Boolean functions which is the main tool used in this paper. Let $\mathbb{F}_2$ denotes the two elements field. For any positive integer n, the n–dimensional vector space over $\mathbb{F}_2$ is denoted $\mathbb{F}_2^n$. A Boolean function f is a mapping $\mathbb{F}_2^n \longrightarrow \mathbb{F}_2$. If f is a Boolean function, we denote by $\widehat{f}$ its Fourier transform, which is by definition the real valued mapping $\mathbb{F}_2^n \longrightarrow \mathbb{R}$ defined, for any n–dimensional binary vector u, by

$$\widehat{f}(u) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{x \cdot u}, \tag{1}$$

where $x \cdot u = x_1 u_1 + \cdots + x_n u_n$. This transform is invertible and the inverse is given by:

$$\widehat{\widehat{f}} = 2^n f \tag{2}$$

Let us recall Parseval's theorem (see [1]):

Theorem 1 (Parseval's theorem). *For any Boolean function $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$, and any vector $u \in \mathbb{F}_2^n$, the following relation holds:*

$$\sum_{u \in \mathbb{F}_2^n} \widehat{f}^2(u) = 2^n \sum_{x \in \mathbb{F}_2^n} f^2(x). \tag{3}$$

When dealing with Boolean functions, we rather resort to the Walsh transform which gets nicer properties in most cases. The Walsh transform of a Boolean function f is simply the Fourier transform of its sign function f_χ where $f_\chi = (-1)^{f(x)} = 1 - 2f(x)$ that is,

$$\widehat{f_\chi}(u) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+x\cdot u} \tag{4}$$

As shown in [1], the correspondence between the Fourier and the Walsh transforms is given by

$$\forall u \in \mathbb{F}_2^n, \quad \widehat{f_\chi}(u) = 2^n \delta_0(u) - 2\widehat{f}(u), \tag{5}$$

where $\delta_0(u)$ is the Kronecker symbol, equals 1 if u is the n–dimensional zero vector, and equals 0 elsewhere.

An (n, m) vectorial Boolean function, or simply an (n, m)–function, is a function over the vector space $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$. Any of the output components defines a Boolean function. Therefore, an (n, m)–function f is nothing but a m–dimensional vector where each component is a n–variable Boolean function. The j^{th} coordinate is denoted by f_j. The Walsh matrix of any (n, m)–function is the $2^m \times 2^n$ dimensional matrix $W_f = \left(w^f_{u,v}\right)$ so that (see [2]):

$$\forall u \in \mathbb{F}_2^m,\ \forall v \in \mathbb{F}_2^n,\ w^f_{u,v} = \sum_{x \in \mathbb{F}_2^n} (-1)^{u\cdot f(x)+v\cdot x} \tag{6}$$

In other words, the rows indexed by $u \in \mathbb{F}_2^m$ of this matrix are the Walsh transforms of the linear combinations of the functions f_i defined by $x \longmapsto u \cdot f(x)$. The coefficients of the Walsh matrix of a function are called the spectrum of the function.

N.B. Matrices indexes may be without ambiguity either an integer or a binary vector representing the same integer in natural binary coding. Thus, if u and v are vectors of same dimension, we may write $u < v$. It means that the number represented by u is smaller than the one represented by v.

An interesting property relates the Walsh matrices of composed functions.

Proposition 1 (see [2]). *If f is a (n, m)–function and g is a (p, n)–function then*

$$W_{f\circ g} = \frac{1}{2^n} W_f \times W_g. \tag{7}$$

3 Problem Statement

Let us consider a compound system involving two BDS coupled in a unidirectional way, a setup called master-slave configuration. The system obeys the following equations

$$\begin{cases} x_{k+1} = F(x_k, u_k) \\ y_k \quad\;\; = G(x_k, u_k) \end{cases} \qquad \text{(Master equation)} \tag{8}$$

$$\begin{cases} \widehat{x}_{k+1} = f(\widehat{x}_k, y_k) \\ \widehat{u}_k \quad\;\; = g(\widehat{x}_k, y_k) \end{cases} \qquad \text{(Slave equation)} \tag{9}$$

where x_k and $\widehat{x}_k$ are n dimensional vectors. The subscript k stands for the discrete time. The $(n+1, n)$–functions F and f are called the next-state functions. The $(n+1, 1)$–functions G and g are called the output functions. The input and output of (8) (respectively (9)) are u_k and y_k (respectively y_k and $\widehat{u}_k$). The situation is depicted Figure 1. We are interested in self-synchronization. Before

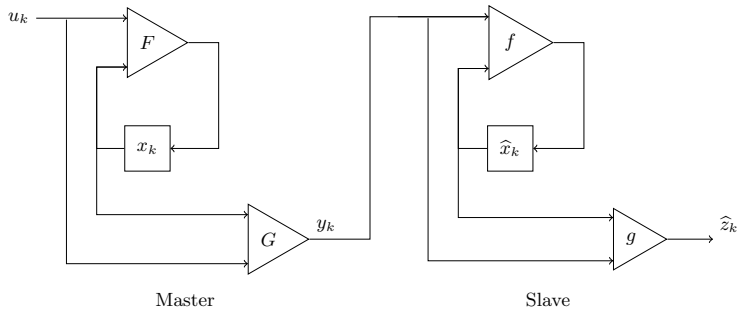

Fig. 1: Overall system

proceeding further, let us introduce some formal definitions.

Definition 1 (Synchronizing sequence). *A sequence* (u) *is synchronizing for* (8)–(9) *if there exists an integer* k_u *so that for all initial states* x_0 *and* $\widehat{x}_0$*:*

$$\forall k \geq k_u,\ x_k = \widehat{x}_k \tag{10}$$

Remark 1. This definition can be generalized by adding a constant delay r so that (10) turns into $\forall k \geq k_u,\ x_k = \widehat{x}_{k+r}$.

Definition 2 (Finite time synchronization). *The overall system* (8)–(9) *is finite time synchronizing if the minimum value* k_u *is upper bounded when* (u) *stands in the set of all input sequences. The upper bound is called the synchronization delay.*

Remark 2. If (u) is a random sequence then, (u) turns into (U) and k_u turns into K_U which is a random variable.

Definition 3 (Statistical synchronization). *A system is statistically synchronizing if* $\lim_{k\to+\infty} \text{Prob}(K_U \leq k) = 1$.

In the sequel, we will focus on the slave system. The synchronizing properties of this subsystem are entirely defined by those of the $(n+1, n)$–function f. Therefore, the previous definitions may be transposed as follow:

Definition 4 (Self-Synchronizing sequence). *A sequence* (y) *is self-synchronizing for* f *if there exists an integer* k_y *so that for all initial state* x_0 *and* $\widehat{x}_0$

$$\forall k \geq k_y,\ x_k = \widehat{x}_k \tag{11}$$

Definition 5 (Finite time self-synchronization). *The function* f *is finite time self-synchronizing if the minimum value* k_y *is upper bounded when* (y) *stands in the set of all input sequences. The upper bound is called the self-synchronization delay of* f.

Definition 6 (Statistical self-synchronization). *A function* f *is statistically self-synchronizing if* $\lim_{k\to+\infty} \text{Prob}(K_Y \leq k) = 1$, *where* K_Y *is the random synchronization delay for the random sequence* (Y).

For our purpose, we must define, for any positive integer i, the iterated function ϕ_i that expresses the internal state after $i+1$ iterations by means of the initial internal state and the input sequence. More precisely, for the sequence $(y) = (y_0, \ldots, y_i) \in \mathbb{F}_2^{i+1}$ and $x \in \mathbb{F}_2^n$, the value $\phi_i(y, x)$ is:

$$\phi_i(y, x) = f(y_i, f(y_{i-1}, f(\ldots, f(y_0, x)\cdots))) \tag{12}$$

This function plays a central role in the sequel.

4 Walsh Transform of the Iterated Function

In this section, the Walsh spectrum of the iterated function ϕ_i is expressed by means of the spectrum of the next-state function f. We then observe the consequences on the synchronization properties of f.

Let us denote by f^0 (respectively f^1) the (n, n)–function which is the restriction of f to the input bit $y = 0$ (respectively to $y = 1$). For a given fixed input sequence $y = (y_0, \ldots, y_i)$, we denote by ϕ_i^y the (n, n)–function that expresses the internal state after $i+1$ iterations: $\phi_i^y : x \longmapsto \phi_i(y, x)$. We express the spectrum of the function ϕ_i^y.

Proposition 2. *The Walsh matrix of* ϕ_i^y *is*

$$W_{\phi_i^y} = \frac{1}{2^{n\cdot i}} W_{f^{y_i}} W_{f^{y^{i-1}}} \times \cdots \times W_{f^{y_0}}. \tag{13}$$

Proof. The proof is a direct consequence of Proposition 1.

For two vectors $u = (u_0, \ldots, u_i)$ and $v = (v_0, \ldots, v_{n-1})$, their concatenation, denoted $u|v$ is by definition the $(n+i+1)$–dimensional vector $u|v = (u_0, \ldots, u_i, v_0, \ldots, v_{n-1})$.

Proposition 3. *Let* $v,\ t \in \mathbb{F}_2^n,\ u \in \mathbb{F}_2^{u+1},\ z = u|v$ *and* $\left(w_{t,v}^{\phi_i^y}\right) = W_{\phi_i^y}$. *The entries of the Walsh matrix of the iterated function* ϕ_i *are defined by*

$$w_{t,z}^{\phi_i} = w_{t,u|v}^{\phi_i} = \sum_{y \in \mathbb{F}_2^{i+1}} (-1)^{u \cdot y} w_{t,v}^{\phi_i^y} \tag{14}$$

Proof. By definition of the Walsh coefficients,

$$\begin{aligned} w_{t,z}^{\phi_i} = w_{t,u|v}^{\phi_i} &= \textstyle\sum_{x \in \mathbb{F}_2^n, y \in \mathbb{F}_2^{i+1}} (-1)^{t \cdot \phi_i(y,x) + (u|v) \cdot (y|x)} \\ &= \textstyle\sum_{y \in \mathbb{F}_2^{i+1}} \sum_{x \in \mathbb{F}_2^n} (-1)^{t \cdot \phi_i(y,x) + u \cdot y + v \cdot x} \qquad = \sum_{y \in \mathbb{F}_2^{i+1}} (-1)^{u \cdot y} w_{t,v}^{\phi_i^y} \end{aligned}$$

According to (13), the Walsh matrix of ϕ_i can be expressed as sums and differences of the Walsh matrices $W_{\phi_i^y}$ obtained for all the possible sequences (y) of length $i+1$. Therefore, we get the expression of the spectrum of ϕ_i as a function of the spectrum of f.

5 Self-Synchronizing Sequences

In this section we are interested in characterizing the sequences (y) that self-synchronize the function based on the spectrum of the function ϕ_i^y.

Proposition 4. *The Walsh matrix of the iterated function is*

$$W_{\phi_i^y} = \begin{pmatrix} 2^n & 0 \cdots & 0 \\ \pm 2^n & 0 \cdots & 0 \\ \vdots & \vdots & \vdots \\ \pm 2^n & 0 \cdots & 0 \end{pmatrix} \tag{15}$$

if and only if (y) *is a self-synchronizing sequence for this function.*

Proof. By definition, if (y) is a self-synchronizing sequence $\phi_i^y(x)$ does not depend on x thus, ϕ_i^y is a constant function. The converse can be derived by applying (2) to the rows of the above matrix (which are the Walsh transforms of the linear combinations of the component functions f_j.

The matrix $W_{\phi_i^y}$ can easily be worked out with (13).

Remark 3. If (y) is a self-synchronizing sequence for the function f then, any other sequence that contains (y) as a subsequence is also self-synchronizing for f.

Proposition 5. *If* f *has at least one self-synchronizing sequence then* f *is statistically self-synchronizing.*

Proof. A self-synchronizing sequence has a finite length, therefore, its probability of occurrence is one in a sequence whose length approaches infinity.

These results can be used to check in the spectral domain whether a given sequence is self-synchronizing or not. But more importantly it gives some conditions on the Walsh spectrum of the function that can be used to build self-synchronizing systems that have the form (9).

6 Influence of Variables

Roughly speaking, influence reveals the ability of a variable to change the output of a function. Let us stress that the notion of variable influence has been used in several papers (e.g. [3–5] to mention a few). For reasons explained later on, we must revisit the existing formal definitions of such a notion because they are not suited for our purpose.

6.1 Influence of a Single Variable

Let f be a boolean function of the variable x. The influence of one variable x_i over a Boolean function f is defined as the probability that the value of $f(x)$ changes if the value of the component x_i is changed, the other components being set randomly. This definition may be expressed in an equivalent way.

Definition 7. *Let $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$ be a Boolean function and $i \in \{1, \ldots, n\}$ a set of integers. Let e_i be the n–dimensional vector whose components are zero except the i^{th} one which equals 1. The influence of x_i on f is:*

$$I_f(i) = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} \big[f(x) + f(x + e_i) \big]$$

Remark 4. This is related to the so called auto-correlation function of f which is $r_f(u) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+f(x+u)}$ that is, $I_f(i) = 2^{n-1} - \frac{1}{2} r_f(e_i)$.

6.2 Influence of a Set of Variables

There exists several ways to extend Definition 7. None is more natural than the others. The choice of the right definition depends on what need to be studied. The influence of a subset[3] S of components of x can be defined as the probability that the value of $f(x)$ changes if one of the variables in S changes too. It does not take into account the number of possibilities of choosing the values of the variables that changes the output of the function. A more suitable definition for our purpose should involve the balancedness of the restricted function obtained by fixing the variables not in the set S. Next, the expression of the influence

[3] Note that a variable may be identified to its index. Thus for short S may be also considered as a set of indexes in $\{1, \ldots, n\}$.

of a set containing more than one element is a complex function of its spectral representation and is therefore not suitable for the proposed approach. This is the reason why we rather introduce a new definition of the influence that takes these points into account. The support of a vector u is by definition: $\mathrm{supp}(u) = \{i \in \{1,\ldots,n\} \mid u_i \neq 0\}$.

Definition 8. *Let $f(x)$ be a Boolean function of n variables, S be a set of k components of x. The influence $I_f(S)$ is:*

$$I_f(S) = \frac{1}{2^n(2^k-1)} \sum_{x\in\mathbb{F}_2^n} \sum_{u\in\mathbb{F}_2^n | u\neq 0,\ \mathrm{supp}(u)\subset S} [f(x)+f(x+u)] \tag{16}$$

In other words, the influence of a set of variables is the mean of the probabilities that $f(x)$ changes when x is uniformly randomly chosen, the mean being computed for all possible changes of the value of the variables in S.

Remark 5. When the set S contains one element, then Definitions 7 and 8 are equivalent.

6.3 Spectral Expression of the Influence

The influence of a set of variables over a Boolean function f can simply be expressed by means of its spectral representation.

Proposition 6. *Let $f(x)$ be a Boolean function of n variables, S be a set of k components of x. The influence $I_f(S)$ is:*

$$I_f(S) = \frac{2^{k-1}}{2^{2n}(2^k-1)} \sum_{v\in\mathbb{F}_2^n \,|\, \mathrm{supp}(v)\cap S\neq\emptyset} \widehat{f_\chi}^2(v) \tag{17}$$

Proof. For any vector u, let $f^u : x \longmapsto f(x) - f(x+u)$. It is easy to see that $f^u(x) = 0$ if $f(x) = f(x+u)$ and $f^u(x) = \pm 1$ if $f(x) \neq f(x+u)$. This implies that $[f^u(x)]^2 = f(x) + f(x+u)$. Therefore, by using (3),

$$I_f(S) = \frac{1}{2^{2n}(2^k-1)} \sum_{v\in\mathbb{F}_2^n} \sum_{u\in\mathbb{F}_2^n | u\neq 0,\ \mathrm{supp}(u)\subset S} \left[\widehat{f^u}(v)\right]^2. \tag{18}$$

By expressing $\widehat{f^u}(v)$ by means of $\widehat{f}(v)$, we get

$$\widehat{f^u}(v) = \sum_{x\in\mathbb{F}_2^n} f(x)(-1)^{v\cdot x}(1-(-1)^{v\cdot u}) = (1-(-1)^{v\cdot u})\widehat{f}(v),$$

By using this expression of $\widehat{f^u}(v)$ in (18), we get

$$I_f(S) = \frac{1}{2^{2n}(2^k-1)} \sum_{v\in\mathbb{F}_2^n} \widehat{f}^2(v) \sum_{u\in\mathbb{F}_2^n | u\neq 0,\ \mathrm{supp}(u)\subset S} [1-(-1)^{v\cdot u}]^2,$$

and thus:

$$I_f(S) = \frac{1}{2^{2n-2}(2^k-1)} \sum_{v/\,\mathrm{supp}(v)\cap S\neq\emptyset} \widehat{f}^2(v),$$

and finally using (5) *the desired result stands.*

Remark 6. This definition of the influence of variables is very close, up to a factor that depends on the cardinality of S, to the definition of the so called *variable variation* given in [4].

Proposition 7. *Let f be a Boolean function.*

1. *f is bent if and only if for all non-empty subset S of variable indexes, one has $I_f(S) = \frac{1}{2}$,*
2. *f does not depend on the variables in the subset S if and only if $I_f(S) = 0$.*

Proof. *1. If f is bent, then $\forall u \in \mathbb{F}_2^n$, $\widehat{f_\chi}(u) = \pm 2^{n/2}$. Then replacing this expression in* (17)*, we get*

$$I_f(S) = \frac{2^{k-1}\times 2^n}{2^{2n}(2^k-1)} \sum_{v/\,\mathrm{supp}(v)\cap S\neq\emptyset} 1 = \frac{2^{k-1}\times 2^n}{2^{2n}(2^k-1)}(2^n - 2^{n-k}) = \frac{1}{2}$$

Conversely, if for all non-empty subset of variable indexes S of k elements, one has $I_f(S) = 1/2$, then, by replacing in relation (17)*, one gets*

$$\frac{2^{k-1}}{2^{2n}(2^k-1)} \sum_{v|\,\mathrm{supp}(v)\cap S\neq\emptyset} \widehat{f_\chi}^2(v) = \frac{1}{2}.$$

By Parseval's Theorem, and as $\mathrm{supp}(v)\cap S = \emptyset \iff \mathrm{supp}(v) \subset \overline{S}$*, where $\overline{S}$ denotes the complementary set of S,*

$$2^{2n} - \sum_{v|\,\mathrm{supp}(v)\subset\overline{S}} \widehat{f_\chi}^2(v) = 2^{2n-k}(2^k-1),$$

Thus:

$$\sum_{v|\,\mathrm{supp}(v)\subset\overline{S}} \widehat{f_\chi}^2(v) = 2^{2n} - 2^{2n-k}(2^k-1) = 2^{2n-k} \tag{19}$$

When applying this relation with $S = \{1,\ldots,n\}$, the sum (19) *has only one term which is $\widehat{f_\chi}(0)^2 = 2^{2n-n} = 2^n$. The other values are obtained by induction on the weight of vector u. Let v be a non-zero vector. Let us choose S such that $\overline{S} = \mathrm{supp}(u)$. One can split the sum of relation* (19) *into the term $\widehat{f_\chi}^2(v)$ and the $2^{n-k}-1$ others terms which all equal 2^n by the induction hypothesis as they all have weight strictly lower than the weight of v. Thus, $\widehat{f_\chi}^2(u) + (2^{n-k}-1)\cdot 2^n = 2^{2n-k}$ and the result holds.*

2. If f is constant with respect to the variables in S, $\forall x, \forall v \in \mathrm{supp}(S)$, $f(x) + f(x+v) = 0$ thus, $I_f(S) = 0$. Conversely, $I_f(S) = 0$ implies that for all v the terms $f(x) + f(x+v)$ equal 0 since $I_f(S)$ is a sum of positive terms.

6.4 Extension of the Influence to Vectorial Boolean Functions

In this section, we extend the definition of the influence to vectorial Boolean functions in order to characterize the self-synchronization property of f.

Definition 9. *The influence of a set of variables S over a vectorial Boolean function f is the mean of the influence of S over each coordinate function f_j.*

$$I_f(S) = \frac{1}{q}\sum_{j=1}^{q} I_{f_j}(S) \tag{20}$$

Proposition 8. *If f does not depend on the variables in S then, the influence is $I_f(S) = 0$.*

Proof. This is a simple consequence of Proposition 7.

6.5 Self-synchronization vs influence

We aim at relating the self-synchronization property of the function f stated in Definition 5 and 6 to the influence of the initial state on the corresponding iterated function ϕ_i. Let S_x denote the subset of variables that corresponds to the initial state x.

Proposition 9. *The function f is finite time self-synchronizing if and only if, there exists an integer i large enough so that for any finite sequence (y) of length $i+1$, the iterated function $\phi_i(y, x)$ does not depend on the internal state component x. In other words, the variable x of $\phi_i(y, x)$ has no longer influence after a transient time that is, $I_{\phi_i}(S_x) = 0$*

Proposition 10. *The function f is statistically self-synchronizing if and only if, there exists an integer i large enough so that for at least one sequence (y) of length $i+1$, the iterated function $\phi_i(y, x)$ does not depend on the internal state component x. In other words, there is at least one input sequence (y) so that the variable x of $\phi_i^y(x)$ has no influence over ϕ_i^y thus, $I_{\phi_i^y}(S_x) = 0$.*

It can be inferred from (14) that this implies for W_{ϕ_i} to be sparse. The only possible non-zero coefficients are located on the column v so that $\mathrm{supp}(v) \cap S = \emptyset$.

7 Examples

7.1 Academic Example

We now show how to use the previous results to build a $(n+1, n)$–function f that is statistically self-synchronizing. Let $f : \mathbb{F}_2 \times \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$, and f^0 (respectively f^1) the restriction of f to $y = 0$ (respectively $y = 1$). A statistically self-synchronizing function f can be obtained by selecting the appropriate functions

f^0 and f^1 so that there exists an admissible way to multiply the corresponding Walsh matrices W_{f^0} and W_{f^1} (or their powers) yielding (15). From this perspective, we can consider a lower triangular matrix with zeros on the diagonal except the entry located at row 0 and column 0. Such a matrix has the interesting property that the successive right multiplications with any other matrix tends to produce a matrix of form (15). Therefore we can select f^0 such that its Walsh matrix has the aforementioned structure, f^1 being any vectorial Boolean function. Note that this choice is arbitrary and the role of f^0 and f^1 can be reversed.

Let us provide a constructive approach to find out a suitable (n, n)–function f^0. First, let us recall that the u^{th} row of W_{f^0} is the Walsh transform of the Boolean function defined by $x \longmapsto u \cdot f^0(x)$. That is, the u^{th} row is the Walsh transform of the linear combination of the coordinate functions f_j^0 such that the components u_j equal 1. Let e_j be the canonical vector whose components are 0 except the j^{th} one which equals 1. Considering the rows $u = e_j$ for $j \in \{1, \ldots, n\}$ is equivalent to select each coordinate function f_j^0. Thus, the other rows can be obtained by calculating the Walsh transform of the linear combinations of the functions f_j^0. Interestingly, the functions that depend only on the first k variables have zeros after the first 2^k coefficients.

Proposition 11. *Let f be a n–variable Boolean function. The function f depends only on the first j^{th} variables ($j \leq n$) if and only if*

$$\forall u,\ \mathrm{supp}(u) \notin \{1, \ldots, j\},\ \widehat{f_\chi}(u) = 0$$

Proof. Let us express the Walsh transform of a n–variable function f that indeed depends only on the first j^{th} variables. It can be expressed, for $u \in \mathbb{F}_2^j$ and $v \in \mathbb{F}_2^{n-j}$, as

$$\widehat{f_\chi}(u|v) = \sum_{y \in \mathbb{F}_2^{n-j}} (-1)^{v \cdot y} \sum_{x \in \mathbb{F}_2^j} (-1)^{f(x|0) + u \cdot x}.$$

This implies that $\widehat{f_\chi}(u|v) = 0$ if $v \neq 0$, which proves that Conversely, for $x \in \mathbb{F}_2^j$ and $y \in \mathbb{F}_2^{n-j}$, one has $f_\chi(x|y) = \frac{1}{2^n} \widehat{\widehat{f_\chi}}(x|y) = \frac{1}{2^n} \sum_{u,v} \widehat{f_\chi}(u|v)(-1)^{x \cdot u + v \cdot y}$. As it is assumed that, for $v \neq 0$, one has $\widehat{f_\chi}(u|v) = 0$, we deduce $f(x|y) = \frac{1}{2^n} \sum_u \widehat{f_\chi}(u|0)(-1)^{u \cdot x}$. It is clear that this expression does not depend on y and the result holds.

This proposition implies that if the coordinate functions f_j^0 are chosen so that it depends only on the first $j - 1$ variables then, W_{f^0} is of the form (15). This is true since the rows $u < e_j$ are Walsh transforms of linear combinations of functions that depend on the first $j - 1$ variables. Note that the function f_0^0 has to be constant, its value is therefore either 0 or 1.

We propose to construct a $(3, 3)$–function f^0 so that its Walsh matrix has the desired structure of an upper triangular matrix. According to the aforementioned

considerations, a function f^0 which fulfills the required constraints can be

$$f^0 = \begin{cases} f_0^0 = 0 \\ f_1^0 = x_0 \\ f_2^0 = x_1 + x_0 x_1 \end{cases} \tag{21}$$

Its Walsh transform is

$$W_{f^0} = \begin{pmatrix} 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & -4 & 4 & 4 & 0 & 0 & 0 & 0 \\ 4 & -4 & 4 & 4 & 0 & 0 & 0 & 0 \\ -4 & 4 & 4 & 4 & 0 & 0 & 0 & 0 \\ -4 & 4 & 4 & 4 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{22}$$

As pointed out before, there are no particular restriction on the function f^1. The sole lower triangular structure of W_{f^0} suffices to guarantee that any sequence that contains three 0s self-synchronizes the system.

This is one approach to build self-synchronizing functions. But as it can be seen the lower triangular structure of W_{f^0} implies a very specific structure to f^0. It would now be interesting to find out other constructions that release the constraint on f^0.

7.2 Application to Self-Synchronizing Stream Ciphers

In this section, we are interested in the self-synchronizing property for cryptographic purposes and more exactly for the design of a so called Self-Synchronizing Stream Cipher (SSSC for short). The reader may refer to [6, 7] for examples of SSSC proposed through the eSTREAM European project devoted to stream ciphers. At the transmitter side, the canonical equations governing an SSSC read

$$\begin{cases} x_k = \varphi_\ell(y_{k-1}, \ldots, y_{k-\ell}) \\ z_k = h(x_k, y_{k-1}) \\ y_k = z_k + u_k \end{cases} \tag{23}$$

and at the receiver side, the equations read

$$\begin{cases} \widehat{x}_k = \varphi_\ell(y_{k-1}, \ldots, y_{k-\ell}) \\ \widehat{z}_k = g(\widehat{x}_k, y_{k-1}) \\ \widehat{u}_k = \widehat{z}_k + y_k \end{cases} \tag{24}$$

The sequences (z) and $(\widehat{z})$ are the respective key-streams, x_k and $\widehat{x}_k$ are the respective internal states. The ciphering is performed by the exclusive-OR between the key-stream and the plain-text while the deciphering is performed by the exclusive-OR between the key-stream and the cipher-text. Let us note that (23) and (8) can not directly be identified. It is clear that proper decryption

is achieved whenever $\widehat{z}_k = z_k$. Actually, since φ_ℓ depends at both ends on the same arguments, such a condition is always fulfilled. It is nothing but a synchronization condition.
We propose to resort to the dynamical system (9) for delivering the key-stream instead of a static function like φ_ℓ. The objective of resorting to a recursive approach is to get a more complex ciphering function with a same computational cost. However not all dynamical systems are admissible. Indeed, in (9), f must have the self-synchronization property. Assuming that (9) is finite time self-synchronizing, the state vector $\widehat{x}_k$ must have to be precisely expressed as a function φ_ℓ that does not depend on $\widehat{x}_{k-\ell}$. It must read

$$\begin{cases} \widehat{x}_k = \phi_\ell(y_{k-1}, \ldots, y_{k-\ell}, \widehat{x}_{k-\ell}) = \varphi_\ell(y_{k-1}, \ldots, y_{k-\ell}) \\ \widehat{z}_k = g(\widehat{x}_k, y_{k-1}) \end{cases}, \tag{25}$$

where ϕ_ℓ is the iterated function. It has been stressed in [8] and [9] that this is related to the flatness property borrowed from control theory.

If f has the statistical self-synchronization property, it means that ℓ is not bounded and this may increase the complexity of the next-state function f causing the diffusion/confusion properties of the cipher to increase. Besides, if ℓ is not bounded, the canonical representation cannot be obtained in an explicit way. That prevents from any practical implementation. We illustrate the statistical self-synchronizing property. Let us turn back to the example of Section 7.1. It has been pointed out that f^1 can be arbitrary. We define for example f^1 as

$$f^1 = \begin{cases} f_0^1 = x_0x_1 + x_1x_2 + x_0x_1x_2 \\ f_1^1 = x_0x_1 + x_2 \\ f_2^1 = x_1x_2 + x_0 \end{cases} \tag{26}$$

Consequently, the function f is defined as

$$f(y, x) = (y + 1)f^0(x) + yf^1(x) \tag{27}$$

Below is given the third iterated function ϕ_2.

$$\phi_2 = \begin{cases} (\phi_2)_0 = x_0x_2y_{k-2}y_{k-1}y_k + x_0x_1x_2y_{k-2}y_{k-1}y_k \\ (\phi_2)_1 = x_0x_1y_{k-2}y_{k-1} + x_0x_2y_{k-2}y_{k-1} + x_1x_2y_{k-2}y_{k-1} + x_0x_1x_2y_{k-2}y_{k-1} \\ \quad +x_0y_k + x_0y_{k-2}y_k + x_2y_{k-2}y_k + x_1x_2y_{k-2}y_k + x_0y_{k-1}y_k \\ \quad +x_0y_{k-2}y_{k-1}y_k + x_0x_1y_{k-2}y_{k-1}y_k + x_2y_{k-2}y_{k-1}y_k + x_0x_2y_{k-2}y_{k-1}y_k \\ (\phi_2)_2 = x_0x_1y_{k-2} + x_1x_2y_{k-2} + x_0x_1x_2y_{k-2} + x_1y_{k-1} + x_0x_1y_{k-1} \\ \quad +x_0y_{k-2}y_{k-1} + x_1y_{k-2}y_{k-1} + x_0x_1y_{k-2}y_{k-1} + x_0x_2y_{k-2}y_{k-1} \\ \quad +x_1x_2y_{k-2}y_{k-1} + x_0x_1y_{k-2}y_k + x_1x_2y_{k-2}y_k + x_0x_1x_2y_{k-2}y_k \\ \quad +x_1y_{k-1}y_k + x_0x_1y_{k-1}y_k + x_0y_{k-2}y_{k-1}y_k + x_1y_{k-2}y_{k-1}y_k \\ \quad +x_0x_2y_{k-2}y_{k-1}y_k \end{cases} \tag{28}$$

Such a simple example illustrates the relevance of resorting to a recursive approach. Indeed we can easily imagine the complexity of implementing the canonical form instead of the recursive equations when ℓ is large. Besides, as stressed above, when ℓ is not bounded, an explicit expression cannot be obtained.

8 Conclusion

In this paper a spectral characterization of the synchronization property of Boolean dynamical systems has been provided. Conditions on the spectrum of the next-state function have been derived for two systems coupled in a unidirectional way to guarantee self-synchronization. Two kinds of self-synchronization have been considered: the statistical one and the finite one. Next some conditions have been stated for a specific input sequence to allow the system to be self-synchronizing. Some of the results have been based on the notion of influence of variables, a notion that have been extended to vectorial Boolean functions for the purpose of the paper. A potential application to cryptography has finally been given as an illustrative example. To obtain a complete cryptosystem setup, further work will investigate relevant classes of boolean functions as well as cryptanalysis aspect.

References

1. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, chapter Boolean Functions for Cryptography and Error-Correcting Codes. Cambridge Press, 2010.
2. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, chapter Vectorial Boolean Functions for Cryptography. Cambridge Press, 2010.
3. J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. In *SFCS '88: Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 68–80, Washington, DC, USA, 1988. IEEE Computer Society.
4. Guy Kindler and Shmuel Safra. Noise-resistant boolean-functions are juntas, 2003. Availaible at: http://www.math.tau.ac.il/ safra/PapersAndTalks/nibfj.ps.
5. Jean-Luc Marichal. The influence of variables on pseudo-boolean functions with applications to game theory and multicriteria decision making. *Discrete Applied Mathematics*, 107(1-3):139 – 164, 2000.
6. J. Daemen and K. Paris. The self-synchronzing stream cipher moustique. Technical report, e-Stream Project, 2006. Available at: http://www.ecrypt.eu.org/stream/p3ciphers/mosquito/mosquito_p3.pdf.
7. J. Daemen, J. Lano, and B. Preneel. Chosen ciphertext attack on sss. *eSTREAM, ECRYPT Stream Cipher Project, Report 2005/044*, June 2005. Available online at http://www.ecrypt.eu.org/stream/papers.html/044.pdf.
8. Gilles Millérioux, Philippe Guillot, José Maria Amigó, and Jamal Daafouz. Flat dynamical systems and self-synchronizing stream ciphers. 2008. http://hal.archives-ouvertes.fr/docs/00/33/18/33/PDF/FlatDS_SSSC.pdf.
9. Gilles Millérioux and Philippe Guillot. Self-synchronizing stream ciphers and dynamical systems: state of the art and open issues. *International Journal of Bifurcation and Chaos, 20(9)*, September 2010.
10. U. M. Maurer. New approaches to the design of self-synchronizing stream cipher. *Advance in Cryptography, In Proc. Eurocrypt '91, Lecture Notes in Computer Science*, pages 548–471, 1991.
11. Nathan Keller. On the influence of variables on boolean functions in product spaces, May 2009. Availaible at: http://arxiv.org/PS_cache/arxiv/pdf/0905/0905.4216v1.pdf.