On random Boolean threshold networks

Reinhard Heckel, Steffen Schober and Martin Bossert
Institute of Telecommunications and Applied Information Theory
Ulm University
Albert-Einstein-Allee 43, 89081Ulm, Germany
Email: {reinhard.heckel, steffen.schober, martin.bossert}@uni-ulm.de

***Abstract*—Ensembles of Boolean networks using linear random threshold functions with memory are considered. Such ensembles have been studied previously by Szejka et al. [1]. They obtained analytical results for the order parameter which can be used to predict the expected behavior of a network randomly drawn from the ensemble. Using numerical simulations of randomly drawn networks, Szejka et al. [1] found marked deviations from the predicted behavior. In this work improved analytical results are provided that better match up the numerical results. Furthermore, the critical point in their analysis is identified. In the model studied, each node is not only dependent on the K regular inputs, but also on the previous state of the node. The results show that this feedback loop accounts for the low order parameter and tolerance on random errors, even for networks with high in-degree.**

I. Introduction

A (synchronous) Boolean network (BN) can be viewed as a collection of N nodes $V = \{v_1, v_2, \ldots, v_N\}$ with memory. The state of a node v denoted by $s_v(t) \in \{0,1\}$ for $t \in \mathbb{N}$ and is determined by

$$s_v(t) = f_v\left(s_{v_1}(t-1), \ldots, s_{v_{k_v}}(t-1)\right). \tag{1}$$

where $\{v_1, \ldots, v_{k_v}\}, 1 \leq k_v \leq N$ are the *controlling nodes* and $f_v : \{0,1\}^{k_v} \to \{0,1\}$ is a Boolean function. [1] Boolean networks serve as an abstract model of interacting agents. For example, BNs have been used to model (small scale) genetic regulatory networks, see [3]–[5]. In the late 1960's Stuart Kauffman proposed to study Boolean networks chosen at random from well defined ensembles of networks to understand large scale genetic regulatory networks [6], [7]. He was interested in finding general properties that possibly underlie all genetic networks and that could explain features of living organisms. Such *random Boolean networks* (RBNs) are constructed as follows: First for each node a function is chosen according to a well defined probability distribution from a predefined set of Boolean functions. Second for each node, the controlling nodes are chosen from V according some probability distribution. Finally, a random initial state is chosen and the network is evolved according Equation (1).

Kauffman discovered that depending on the networks topology and the choice of functions, a RBN operate in different dynamic regimes. In the so-called *ordered* regime, most of its components are *frozen*, i.e. keep their state when being updated. Further, single transient errors that change the state of a randomly chosen node from 0 to 1 or vice versa tend to vanish. Contrary, in the *disordered* regime, only a few frozen nodes exist and single transient errors propagate to many other nodes.

In this work random networks using linear threshold functions are considered. Let $0 < K \in \mathbb{N}$ and $h \in \mathbb{R}$ be fixed parameters. To each node v, a function is assigned, defined by

$$f_v(x_1, \ldots, x_K, x_{K+1}) = \begin{cases} 1, & \text{if } r > h \\ x_{K+1} & \text{if } r = h \\ 0, & \text{if } r < h \end{cases} \tag{2}$$

where

$$r := r(x_1, \ldots, x_K) = \sum_{i=1}^{K} w_i \cdot x_i. \tag{3}$$

The weights w_i are chosen uniformly at random from $\{-1, 1\}$, which defines a probability distribution on the set of possible functions. Then for $1 \leq i \leq K$ the controlling nodes v_i are chosen uniformly at random from V. The state $s_v(t)$ is obtained by setting $x_i = s_{v_i}(t-1)$ for $1 \leq i \leq K$ and $x_{K+1} = s_v(t-1)$. In other words, K inputs are chosen randomly, whereas the input $K+1$ is always connected to the function's output. Hence the state of any node v at time t depends on its own state at time $t-1$. These networks were studied by Szejka et al. [1], motivated by the fact that such networks have successfully been used to study small scale gene regulatory networks, see for example [3], [5] for modeling the cell cycle in yeast.

Using the so-called annealed approximation Szejka et al. [1] derived expressions in dependence of the parameters h and K, for the time evolution of the proportion of nodes being 1, and for the expected sensitivity of a random Boolean function. The expected sensitivity can be used as an order parameter for the ensemble to predict its dynamical regime. Szejka et al. [1] found deviations between their analytical results for integer valued thresholds and numerical simulations, which they attributed to the annealed approximation. In this work improved analytical results are given, that better match up the numerical results. Further the critical point in their analysis is identified.

The outline is as following: In Section II, a brief introduction in random Boolean networks and their order parameter

[1] It should be noted that the state of all nodes is computed in parallel in each time step, hence the name synchronous Boolean network. In fact other *updating schemes* are used, for example the updating schemes can be parameters of the network by using different *response times* of nodes on regulative actions [2].

is given. In the following sections our main results are derived. In Section III-A the so-called bias map is derived for random threshold networks and some results that will be needed for further calculations are given. In Section III-B the order parameter is derived, by first calculating the average sensitivity of threshold functions, where the distinctions is made between integer and non integer thresholds. In section III-C the average sensitivity of functions whose input x_{K+1} is randomly chosen is considered, instead of being set to the output. In Section III-D, the previously derived results are used to obtain the *phase diagram*. The phase diagram visualizes for which combination of h and K, random threshold networks operate in the chaotic regime. Furthermore random threshold function with $h = 0$ are discussed in more detail.

II. Random Boolean networks and their order parameters

An important question for random Boolean networks is their expected dynamical behavior and the expected robustness against single transient errors. Let us concentrate on the last point. If a random chosen node is disturbed, i.e. its state is changed from one to zero or vice versa, one is interested in the evolution of this disturbance. Will it tend to spread through the whole network, possibly affecting all nodes? Or will the disturbance die out, indicating the so-called ordered phase? In some case, depending on the parameters of the random construction process, there exists a single *order parameter* that can be used to answer this question. Some well known example are the so-called NK-networks studied by Kauffman [6]. These functions are chosen uniformly at random from the set of all Boolean functions with K arguments. The controlling nodes are chosen uniformly at random from all $\binom{N}{K}$ possibilities. This ensemble can be described by the single parameter K. It is well known that ordered behavior is only found if and only if $K \leq 2$.

Often the order parameter is obtained by the so-called annealed approximation [8]. It provides the probabilistic framework to determine the dynamical regime of a random network. The annealed approximation is a mean-field theory that neglects correlations between nodes. It is assumed that at each time step the functions and the controlling nodes are drawn at random again. If N is large, this procedure allows for quite accurate predictions [8]. Also for ensembles like those studied here the predictions for the annealed model coincides the non-annealed model if $N \to \infty$ [9].

Suppose the network operates in a stationary state. Under the assumptions of the annealed analysis we may assume that a node chosen uniformly at random has probability b of being in state 1. Let $p(f)$ denote the probability of choosing the function f. Then the order parameter is defined as

$$\lambda = \sum_f p(f) \sum_{\mathbf{x}\in\Omega^K} s(f,\mathbf{x}) b^{|\mathbf{x}|}(1-b)^{K-|\mathbf{x}|} \tag{4}$$

where $|\cdot|$ denotes the Hamming weight and $s(f,\mathbf{x})$ denotes the sensitivity of f at $\mathbf{x}$ defined by

$$s(f,\mathbf{x}) = \#\{\mathbf{y}\in\Omega^K \,|\, |\mathbf{x}\oplus\mathbf{y}| = 1 \text{ and } f(\mathbf{x}) \neq f(\mathbf{y})\}$$

where $\oplus$ denotes the component wise addition mod 2. It is well know that if $N \to \infty$ and

$$\lambda \leq 1 \tag{5}$$

any single perturbation introduced at a randomly chosen node will vanish with probability one [10]. The parameter b can be obtained as follows (cnf. [11]). Suppose that at t the probability of a random chosen node to be 1 is equal to b_t. Then the so-called *bias map* is defined by

$$b_{t+1} = \sum_f p(f) \sum_{\mathbf{x}\in\Omega^K} f(\mathbf{x}) b_t^{|\mathbf{x}|}(1-b_t)^{K-|\mathbf{x}|}. \tag{6}$$

Let $b_0 = 0.5$. The fix point of the bias map (if it exists), denoted by b, is the expected number of nodes that are one in the stationary state.

Some comments on (5) and (4) are necessary. For convience let us introduce the *average sensitivity* with respect to b

$$as(f) = \sum_{\mathbf{x}\in\Omega^K} s(f,\mathbf{x}) b^{|\mathbf{x}|}(1-b)^{K-|\mathbf{x}|}.$$

By $\mathbf{x}\oplus i$ we denote a vector that is obtained from $\mathbf{x}$ by flipping its ith position. Then

$$P_e := \Pr\left[f(\mathbf{x}) \neq f(\mathbf{x}\oplus i)\right] = \frac{as(f)}{K}$$

where i is chosen uniformly at random from $\{1,\ldots,K\}$. Hence (4) may be written as

$$\lambda = \mathop{\mathbb{E}}_f[as(f)] = K \cdot \mathop{\mathbb{E}}_f[P_e]$$

where the expectation is taken with respect to the whole function ensemble. Now assume a random perturbation at the input of some random function. We expect the state of the node to be changed with $\mathbb{E}[P_e] = \lambda/K$ which in turn will spread to K other nodes [2]. Hence after t time steps λ^t nodes are affected by the perturbation on average. Therefore if $\lambda \leq 1$ the perturbation will vanish with high probability. A rigorous treatment of this topic is given in [10].

III. Random threshold networks

A. The bias map

In this section, we aim at deriving an expression for the bias map in order to find fixed points b. Denote the number of positive weights of a function f by

$$m = m(f) = \#\{i | i \leq 1 \leq K \text{ and } w_i = 1\},$$

where $\#\{\cdot\}$ denotes the cardinality of the set. For convinience we write f_m for a function with $m = m(f_m)$. The corresponding function $r(x_1,\ldots,x_K)$ is denoted by r_m.

Proposition 1: Consider a random threshold network where all functions $f = f(x_1,\ldots,x_K,x_{K+1})$ depend on $K+1$ variables. The threshold h is a fixed constant and $P(w_i = 1) = P(w_i = -1) = 1/2$ for all i independently. Any

[2] As the in-degree of any node is K, the average out-degree of any node is also K.

function receives the previous state of the corresponding node as argument y_{K+1}. Then

$$b = \frac{1}{2^K} \sum_{m=0}^{K} \binom{K}{m} b_m, \tag{7}$$

with

$$b_m = \frac{P(r_m(b) > h)}{1 - P(r_m(b) = h)}. \tag{8}$$

Above

$$P(r_m(b) = h) = \Pr\left[\sum_i w_i \cdot x_i = h \,|\, m \text{ weights are +1}\right],$$

where $P(x_i = 1) = b$ independently for all i. Also

$$P(r_m(b) > h) = \sum_{h+1}^{K} P(r_m(b) = h).$$

For the proof, the following lemma is needed.

Lemma 1: Let f_1 and f_2 be threshold functions with r_1 and r_2 respectively and $m(f_1) = m(f_2) = m$. Then

$$P(r_1 = u) = P(r_2 = u).$$

Proof: Lets consider the probability distribution for r, $\mathbf{w}$, $P(r = u|\mathbf{w})$. As each x_i follows the same distribution,

$$r = \sum_{i \in \{j | w_j = 1\}} x_i - \sum_{i \in \{j | w_j = -1\}} x_i,$$

and the lemma follows. ■

Proof: First Equation 8 is proved. Let x_i^t be the state of an input variable at time t. $f(x_1^t, ..., x_K^t, x_{K+1}^t) = 1$ if $r(\mathbf{x}^t) > 0$. Also $f(x_1^t, ..., x_K^t, x_{K+1}^t) = 1$ if $r(\mathbf{x}^t = 0)$ and $x_{K+1}^t = 1$. Then, from Lemma 1 and Equation 1 follows

$$b_m^{t+1} = P(r_m(b^t) > h) + b_m^{t-1} \cdot P(r_m(b^t) = h). \tag{9}$$

If a fixed point b is reached, also a fixed point for each b_m is reached. Then $b_m^{t+1} = b_m^{t-1} = b_m$ and Equation 8 follows from 9. From Lemma 1 and the fact that w_i is uniform distributed, Equation 7 follows. ■

Lemma 2:

$$P(r_m > h) = \sum_{k=\lfloor h \rfloor}^{m} \binom{m}{k} b^k (1-b)^{m-k} \cdot \sum_{l=0}^{k-1-\lfloor h \rfloor} \binom{K-m}{l} b^l (1-b)^{K-m-l}.$$

Proof: For $y \in \{-1, 1\}$ define the random variable

$$w_m(y) = \#\{i | x_i = 1, w_i = y\}.$$

Clearly

$$P(w_m(1) = k) = \binom{m}{k} b^k (1-b)^{m-k}, \tag{10}$$

and

$$P(w_m(-1) = k) = \binom{K-m}{k} b^k (1-b)^{K-m-k}. \tag{11}$$

$P(r_m > \lfloor h \rfloor) = P(r_m > h)$ as r_m is integer valued. $r_m > h$ holds true, if at least $\lfloor h \rfloor + 1$ more of the positive weighted inputs are 1, than there are negative weighted inputs that are 1. Considering all possible constellations gives:

$$P(r_m > h) = \sum_{k=max(\lfloor h \rfloor, 0)}^{m} P(w_m(1) = k) P(w_m(-1) \geq k+h-1).$$

Note that, if $k - 1 - h \leq 0$, the second sum in equation 2 is 0. ■

Lemma 3: Let the threshold h be integer valued, then

$$P(r_m = h) = \sum_{k=0}^{m} \binom{K-m}{k-h} \binom{m}{k} \cdot b^{2k-h} \cdot (1-b)^{-2k+K+h}$$

and let h be non integer valued, $h \in \mathbb{R} \setminus \mathbb{N}$ then $P(r_m = h) = 0$.

Proof: The proof follows the notation of Lemma 2. For integer valued $h \in \mathbb{N}$, $P(r_m = h)$ can be calculated as following. When k denotes the number of positive inputs that are on, $r_m = h$ holds true, if there are also $k - h$ of the negative inputs on. Considering all possible constellations gives:

$$P(r_m = h) = \sum_{k=0}^{m} P(w_m(1) = k) \cdot P(w_m(-1) = k - h).$$

The result follows from (10) and (11).

If h isn't integer valued, then $P(r_m = h) = 0$ because r_m is integer valued and h not.

■

1) Results for b: In order to find fixed points b, Equation 7 can be solved numerically. As mentioned in the introduction, in the context of the annealed approximation, fixed points are found and interpreted (stable or instable) with the bias map. There, a fixed point is stable if the gradient is smaller then 0. For threshold functions the mapping

$$b^{t+1} = \frac{1}{2^K} \sum_{m=0}^{K} \binom{K}{m} P(r_m(b_t) > h) + b_m^{t-1} \cdot P(r_m(b_t) = h),$$

is found, with the difference that b^{t+1} is not only dependent on b^t but also on b_m^{t-1}. Fixed points can be found numerically, by starting at $b_0 = 0.5$ and $b_m^0 = 0.5, m \in \{1, K\}$. An example iteration is shown in Figure 1.

The fixed points found by solving numerically Preposition 1 are discussed now.

The fixed point for $h = 0$, $b = 0.5$ is independent of K. For each each negative threshold ($-K \leq h < 0$), exactly one stable fixed point between 0.5 and 1 could be found. For $h = 1$ and $K \leq 10$ the $b = 0$. For $K > 11$, fixed points unequal to zero are found. For larger positive thresholds h, K has to be very large for the existence of fixed points unequal to 0.

To validate the results, random threshold networks of the size $N = 10000$ with K inputs per function have been generated and initialized with an bias of $b^0 = 0.5$. The fixed points for $h \leq 0$ agree with less than 0.5% deviation with the predicted ones. For a threshold of $h = 1$ the simulation doesn't

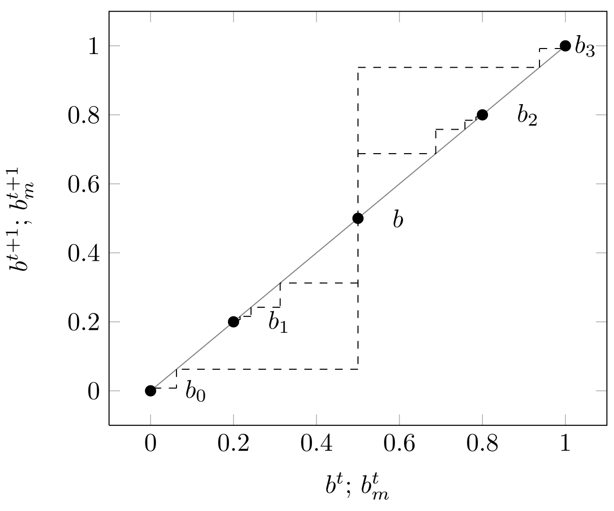

Fig. 1. The cobweb diagram for a threshold function with $K = 3$ and $h = 0$. It can be seen how each b_m evolve to its fixed point, when starting with $b_m^0 = 0.5, m \in \{1, K\}$ and $b^0 = 0.5$.

agree with the predicted results, e.g. for $K = 10$ a fixed point of 0 is expected, whereas in the simulation, a average b of 0.23 is found.

B. Sensitivity and order parameter

Let us consider a threshold function f attached to some node v at time t. Remember that $f = f(x_1, \ldots, x_K, x_{K+1})$ where $x_{K+1} = s_v(t-1)$, i.e. it is set to previous state of the attached node. To derive λ it is assumed that the network operates in its stationary state. Hence for all $1 \le i \le K$ the arguments x_i are independently chosen at random with $\Pr[x_i = 1] = b$. Furthermore it is assumed that x_{K+1} is chosen randomly with $\Pr[x_{K+1} = 1] = \mathbb{E}[f] =: b_f$. As b_f only depends on $m(f)$, b_m is used [3]. From Lemma 1 and $P(w_i = 1) = 1/2$ it follows that

$$\lambda = K \mathop{\mathbb{E}}_{f}[P_e] = \frac{1}{2^K} \cdot \sum_{m=0}^{K} \binom{K}{m} \cdot K \mathop{\mathbb{E}}_{f:w(f)=m}[P_e].$$

For convenience $\mathop{\mathbb{E}}_{m}[P_e]$ is written instead of $\mathop{\mathbb{E}}_{f:w(f)=m}[P_e]$. Here P_e must be computed under the assumption mentioned above, i.e. the first K arguments are binomial distributed with parameter b whereas x_{K+1} is one with probability b_m and otherwise zero. It will be shown that if h is an integer then

$$K \cdot \mathop{\mathbb{E}}_{m}[P_e] = p_m(h) \cdot \{b_m(2h - 2m + K) - h + m\}, \quad (12)$$

and if h is not integer valued

$$K \cdot \mathop{\mathbb{E}}_{m}[P_e] = p_m(\lceil h \rceil) \cdot (K - m + h) + p_m(\lfloor h \rfloor) \cdot (m - \lfloor h \rfloor). \quad (13)$$

In both cases $p_m(k) = P(r_m = k)$ as given by Lemma 3.

Proof of Equation (12)*:* First let $r_m > h$ which implies that $f = 1$. If r_m would be lowered by one the function will still output its previous state namely 1. Hence a flip of any argument will not affect the output. This is also true for $r_m < h$ with the same type of arguments.

Hence consider $r_m = \sum_{i=1}^{m} x_i \cdot w_i = h$. For $(a, b) \in (0, 1) \times (-1, 1)$ define the sets

$$A(a, b) = \{x_i | (x_i, w_i) = (a, b)\}$$

Assume argument $x_i \in A(0, 1)$ is flipped from 0 to 1. The function will change its output if and only if it was 0 before the flip. But the later event has probability $1 - b_m$. Consider all other cases and assume that an x_i is chosen uniformly at random yields

$$K \cdot \mathop{\mathbb{E}}_{m}[P_e] = \Pr[r = h] \{|A(0, 1)| \cdot (1 - b_m) + |A(0, -1)| \cdot b_m + |A(1, 1)| \cdot b_m + |A(1, -1)| \cdot (1 - b_m)\}.$$

Now by definition $|A(0, 1)| = m - |A(1, 1)|$ and $|A(1, -1)| = |A[1, 1]| - h$. Also the sum of the cardinalities of all sets is equal to K. Substituting the constraints into the equation completes the proof. ■

Proof of Equation (13)*:* A threshold function with a non integer threshold can only change its output by changing one input when r is directly above $r_m = \lceil h \rceil$ or directly below $r_m = \lfloor h \rfloor$ the threshold. Suppose $r_m = \lceil h \rceil$, and j of the positive inputs are 1. Then also $j - \lceil h \rceil$ of the negative inputs must be 1. The function's output can change, by lowering r_m which can be done by changing one of the j positive inputs that have value 1 to 0 and one of the $K - m - (j - \lceil h \rceil)$ negative inputs from 0 to 1. Suppose $r_m = \lfloor h \rfloor$, and j is again the number of positive inputs that have value 1. To change the output of a function, r_m has to be increased by one. This can be done by changing one of the $m - j$ positive inputs which have value 0 to 1 or by changing one of the $j - h$ negative inputs that have the value 1 to 0. That gives the lemma. ■

Corollary 1: The average sensitivity is equal for all threshold values in between two consecutive integers, $i < h < i + 1$ where i is an integer. Further, the function's output is unambiguously defined by the K regular inputs of the function, and independent from the input x_{K+1} and therfore also from the previous output.
This result is derived differently, but in accordance with [1].

C. A changed decision rule for $r_m = h$

The dependency on the previous state is an important property of the threshold functions discussed here, and it is closely related to the functions low average sensitivity. To demonstrate what impact this decision rule has on the average sensitivity and order parameter, another decision rule will be discussed: the input x_{K+1} is 1 with probability b, $P(x_{K+1} = 1) = b$, instead of being set to the output. Deriving an equation for fixed points b is analogous as in section **??** and skipped because of limited space.

Lemma 4: Let f_m be a threshold function, where

[3] Follows from Lemma 1.

$P(x_{K+1} = 1) = b$. If h is an integer then

$$\begin{aligned} K \cdot \underset{m}{\mathbb{E}}[P_e] =& P(r_m = h-1) \cdot (m-h+1) \cdot b + \\ & P(r_m = h) \cdot (m + h - 2bm - 2bh + bK) + \\ & P(r_m = h+1) \cdot (K - m + h + 1) \cdot (b-1), \end{aligned} \tag{14}$$

and if h is non integer valued, 13 holds.

Proof: For the constellations $r_m = h-1$, $r_m = h$ and $r_m = h+1$, the output can change by changing one input.

If $r_m = h-1$ the output changes if r_m is increased by one with probability b. Suppose $j = \#\{i | x_i = 1, w_i = 1\}$, then also $j - (h-1) = \#\{x_i = 1, w_i = -1\}$ holds. r_m can be increased by 1 if one of the $m - j$ positive inputs that are 0 changes, or if one of the $j - (h-1)$ negative inputs that are 1 changes. The output changes with probability b because this is the probability for the output to be 1 if $r_m = h$.

Now the case $r_m = h$ is considered. Then $P(f_m = 1) = b$ by definition. Suppose $j = \#\{i | x_i = 1, w_i = 1\}$ then $j - h = \#\{x_i = 1, w_i = -1\}$. If one of the j positive inputs that are 1, or one of the $K - m - (j+h)$ negative that are 0 is flipped, then the output is $f_m^{t+1} = 1$, if $f_m^t = 0$. Changing one of the $m - j$ positive inputs that are 0, or one of the $i + h$ negative inputs that are 1, changes the output, f_m^{t+1}, if $f_m^t = 1$.

If $r_m = h+1$ the output changes if r is decreased by one with probability $1 - b$. Changing one of the j positive inputs that are 1, or one of the $K - m - (j-h)$ negative inputs that are 0 will decrease h.

Putting all together and shortening j for each constellation gives Equation 14. For non-integer thresholds, $r = h$ is not possible, and therefore the input x_{K+1} doesn't have an influence, which gives the rest of the Lemma. ■

D. Numerical results

In this section, the analytical results from the previous sections are evaluated. After choosing parameters for an ensemble, determining the order parameter is a two step process: First a stable fixed point is calculated using the results from Section III-A. Then the order parameter can be obtained with the results from Section III-B.

1) Order parameter for $h = 0$: In Table I are some results of the order parameter λ: λ_{org} refers to the functions whose input x_{K+1} is set to the output, and λ_{rdd} refers to the functions whose input x_{K+1} is 1 with probability b. Further, λ_{org}^{approx} and λ_{rdd}^{approx} are obtained by simulation and will be explained later.

K	λ_{org}	λ_{org}^{approx}	λ_{rdd}	λ_{rdd}^{approx}
3	0.3375	0.3515	**0.9375**	0.9371
4	0.4270	0.4374	**1.0937**	1.0937
15	**0.9789**	0.9875	2.1670	2.1670
16	**1.0155**	1.0240	2.2391	2.2391

TABLE I
ANALYTICAL AND ESTIMATED RESULTS FOR THE ORDER PARAMETER λ OF BOOLEAN THRESHOLD FUNCTIONS.

Random Boolean networks are expected to operate in the chaotic regime for $\lambda > 2$. In contrast (see Table I), random threshold networks with K regular inputs per function and the argument x_{K+1} set to the function's output are expected to operate in the chaotic regime, for $K \geq 16$. But if a random decision is made for the case $r = 0$, instead of keeping the previous state, the networks are expected to operate in the chaotic regime already for $K \geq 4$. Therefore if x_{K+1} is set to the function's previous output makes a huge difference for the stability and spread or random transient errors, for a system modeled as a threshold network. The dependency on the previous output of the function results in a low average sensitivity and has a highly stabilizing effect on the threshold networks.

Szejka et. al. [1] where also deriving an expression for λ, however, but they neglected that each function has its own expectancy value. Substituting b_m with b in equation 12 leads to the results of Szejka et. al. for λ. Consequently they concluded that for $h = 0$ the networks are in the chaotic regime for $K > 12$, obviously this results differs significantly from the one derived here. By using simulations to validate their results, Szejka et. al. found that in "all simulations of networks with connectivities up to $K = 16$, we find only fixed point attractors, which means that these networks are in the frozen phase". Therefore their simulations are in accordance with the result given here.

For a RBN the average sensitivity can be obtained by simulation using the Derrida plot as it has been done e.g. in [12]. This is not feasible here, due to the dependency of the functions on their previous state. Therefore, to further validate the results, Algorithm 1 has been used. Algorithm 1 estimates the average sensitivity for a given function, by estimating $K \cdot \Pr[f(\mathbf{x}) \neq f(\mathbf{x} \oplus i)]$ and therefore also works if a function has memory, e.g. when the function is dependent on its previous state. To obtain λ_{org}^{approx} and λ_{rdd}^{approx}, for each function f_m, Algorithm 1 has been used to estimate $K \cdot \underset{m}{\mathbb{E}}[P_e]$.

Algorithm 1: Average Sensitivity estimation for a Boolean function with memory

```
Data: Boolean Threshold function $f$ of order $K$, $t$ trials
Result: estimated average sensitivity $\tilde{as(f)}$
begin
    $\tilde{as(f)} \longleftarrow 0$
    for $i \longleftarrow 1$ to $t$ do
        generate a random $\mathbf{x} \in \Omega^K$ according to the
        probability distribution of $\mathbf{x}$
        generate a random $\mathbf{e}_i \in \Omega^K$ with $wt(\mathbf{e}_i) = 1$
        $f_i \rightarrow f(\mathbf{x})$
        $\hat{f}_i \rightarrow f(\mathbf{x} \oplus \mathbf{e}_i)$
        if $f_i \neq \hat{f}_i$ then
            $\tilde{as(f)} \longleftarrow \tilde{as(f)} + 1$
    $\tilde{as(f)} \longleftarrow K \cdot \frac{\tilde{as(f)}}{t}$
end
```

Then $K \cdot \underset{m}{\mathbb{E}}[P_e]$ has been averaged over all functions. It can be seen that the analytical results match very good with the estimated ones.

2) The Phase diagram: In this part it is visualized in the so-called *phase diagram*, for which combination of the parameters K and h, threshold networks operate in the chaotic regime ($\lambda > 1$).

To obtain Figure III-D2, for each pair of values K and h first a stable fixed point b was calculated with the results from section III-A, and then the order parameter λ was calculated with the results from section III-B. It can be seen in Figure III-D2 that threshold networks demonstrate how very different behaviour for integer an non integer valued thresholds are. This is because for non-integer thresholds the case $r = h$ isn't possible, therefore the functions are not dependent on their previous output.

In contrast to negative valued thresholds, where a fixed point exists for each K, for positive valued thresholds, the fixed point is $b = 0$ until K becomes large enough, to find a $b \neq 0$. Then, $\lambda > 1$ for the considered values, therefore b determines there the edge of chaos.

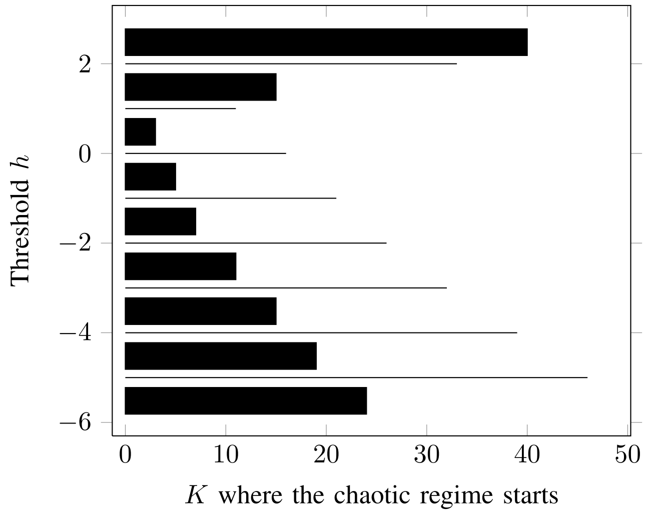

Fig. 2. The Phase diagram for random threshold networks (RTNs), using functions whose input x_{K+1} is set to its output. The bars indicate values for K for which RTNs operate in the ordered regime. Thin bars refer to the integer valued thresholds, whereas the solid ones refer to non integer valued thresholds.

IV. Discussion & Conclusion

Random threshold networks (RTNs) were considered. In the model studied here each node is not only controlled by K other nodes, but is also dependent on the previous state of the node, i.e. possess a local feedback loop. The assumptions of the annealed approximation [8] were used in order to determine the dynamical regime of the RTNs. First the fixed points of the biased map were obtained and solved numerically. In order to obtain the order parameter λ, an expression for the average sensitivity was derived. Evaluating the analytical results, the focus was first on a threshold of $h = 0$. Due to the low average sensitivity of the functions a RTN with $h = 0$ is expected to operate in the ordered regime for $K < 16$. This is a very high degree compared to the so-called NK-networks, that are expected to operate in the chaotic regime already for $K > 2$. The stability even for networks with high in-degree can be traced back to the local feedback. These results are in agreement with simulations done here and in [1]. To demonstrate the effect of the local feedback loop on the network stability, it was shown that if the local feedback is omitted the networks enter the chaotic regime already for $K \geq 4$. These results imply that, if a dynamical system is modeled with a threshold network, assumptions about the feedback are crucial. If the dependence on the previous state is neglected, λ is increased significantly and random errors are more likely to spread.

References

[1] A. Szejka, T. Mihaljev, and B. Drossel, "The phase diagram of random threshold networks," *New Journal of Physics*, vol. 10, no. 6, p. 063009, 2008.

[2] R. Thomas, "Boolean formalization of genetic control circuits," *J Theor Biol*, vol. 42, no. 3, pp. 563–85, Dec. 1973, PMID: 4588055.

[3] M. I. Davidich and S. Bornholdt, "Boolean network model predicts cell cycle sequence of fission yeast," *PLoS ONE*, vol. 3, no. 2, p. e1672, 2008, PMID: 18301750.

[4] J. Saez-Rodriguez, L. Simeoni, J. A. Lindquist, R. Hemenway, U. Bommhardt, B. Arndt, U. Haus, R. Weismantel, E. D. Gilles, S. Klamt, and B. Schraven, "A logical model provides insights into t cell receptor signaling," *PLoS Computational Biology*, vol. 3, no. 8, p. e163, Aug. 2007, PMC1950951.

[5] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, "The yeast cell-cycle network is robustly designed," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 14, p. 4781–4786, Apr. 2004, PMC387325.

[6] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, Mar. 1969, PMID: 5803332.

[7] S. Kauffman, "Homeostasis and differentiation in random genetic control networks," *Nature*, vol. 224, no. 5215, pp. 177–178, Oct. 1969, PMID: 5343519.

[8] B. Derrida and Y. Pomeau, "Random networks of automata: A simple annealed approximation," *EPL (Europhysics Letters)*, vol. 1, no. 2, pp. 49, 45, 1986.

[9] S. Schober and G. Schmidt, "Connections beetween random boolean networks and their annealed model," in *Proc. of the European Conference of Complex Systems (ECCS 2007)*, Dresden, Germany, Oct. 2007.

[10] J. F. Lynch, "Dynamics of random boolean networks," in *Current Developments in Mathematical Biology: Proceedings of the Conference on Mathematical Biology and Dynamical Systems*, R. C. K. Mahdavi and J. Boucher, Eds. World Scientific Publishing Co., 2007, pp. 15–38.

[11] J. Kesseli, P. Rämö, and O. Yli-Harja, "Iterated maps for annealed boolean networks," *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 74, no. 4 Pt 2, p. 046104, Oct. 2006, PMID: 17155130.

[12] J. Kesseli, P. Rämö, and O. Yli-Harja, "Tracking perturbations in boolean networks with spectral methods," *PHYSICAL REVIEW*, vol. E72, 2005.