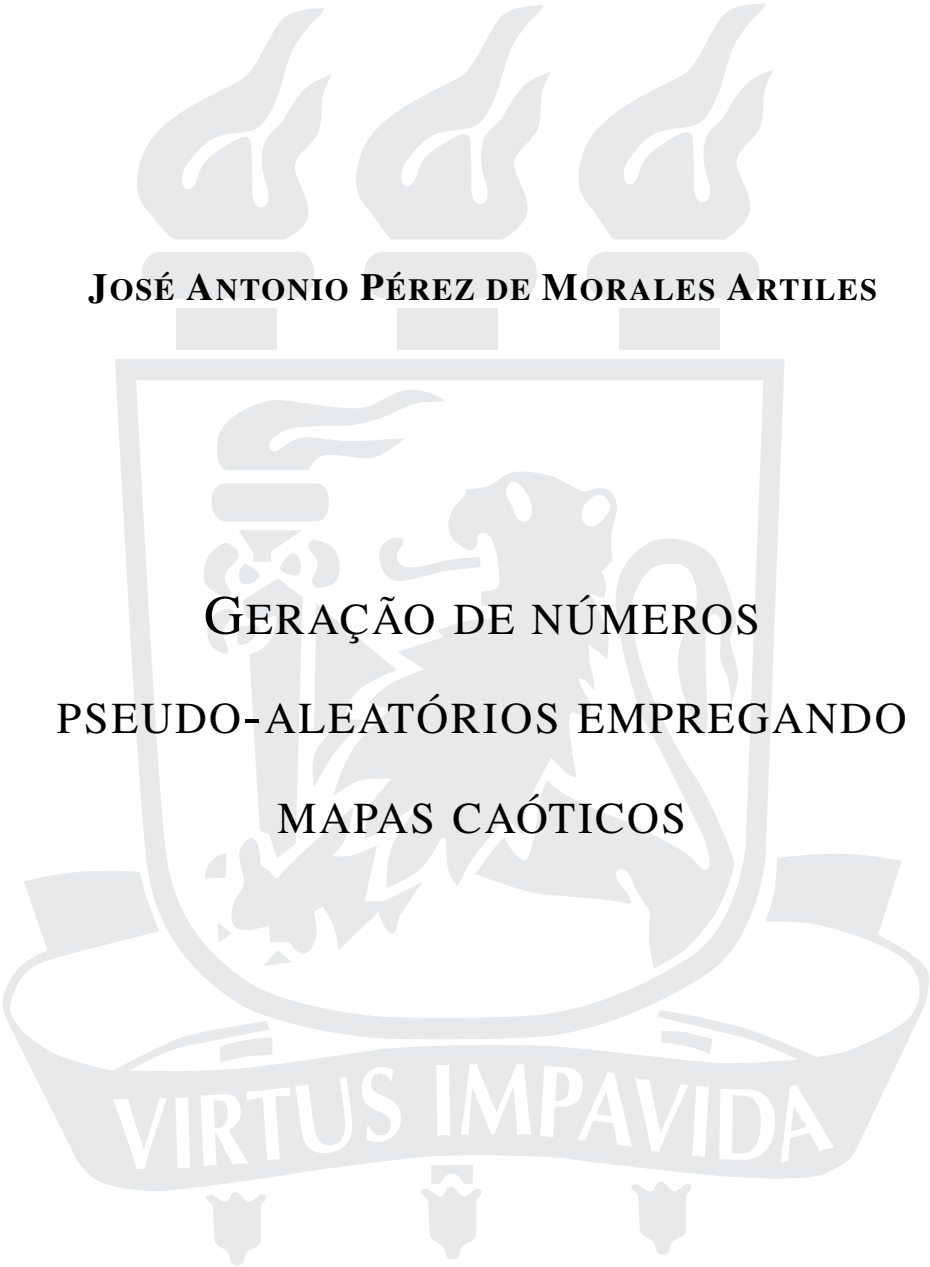
UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE TECNOLOGIA E GEOCIÊNCIAS

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



JOSÉ ANTONIO PÉREZ DE MORALES ARTILES

GERAÇÃO DE NÚMEROS PSEUDO-ALEATÓRIOS EMPREGANDO MAPAS CAÓTICOS

Recife
2016

JOSÉ ANTONIO PÉREZ DE MORALES ARTILES

GERAÇÃO DE NÚMEROS PSEUDO-ALEATÓRIOS EMPREGANDO MAPAS CAÓTICOS

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para obtenção do grau de **Mestre em Engenharia Elétrica**.

Orientador: Prof. Cecilio José Lins Pimentel.

Co-orientador: Prof. Daniel Pedro Bezerra Chaves.

Área de Concentração: Comunicações

Recife
2016

A Deus por me dar a força necessária para não desistir nos momentos mais difíceis.

Para meus pais por sempre me apoiar em minhas decisões.

Meus orientadores e amigos da UFPE, porque sem eles não teria sido possível realizar este trabalho.

RESUMO

Geradores de números aleatórios são amplamente utilizados em aplicações científicas e tecnológicas. Particularmente em criptografia, estes são empregados em sistemas de chave secreta, como geradores de sequência de cifragem. Neste trabalho, propomos várias metodologias para o projeto destes geradores a partir de mapas caóticos. A primeira é baseada em duas técnicas: salto de amostras e discretização codificada variante no tempo. Mostra-se que o procedimento possui alta taxa de geração de bits por amostra caótica quando comparado com codificação fixa no tempo, além de dispensar pós-processamento para melhoria de suas propriedades aleatórias. Outra metodologia utilizada é o emprego de sequências-m para eliminar a correlação residual na sequência codificada. A discretização variante no tempo apresenta uma característica de correlação bem definida que é aproveitada por um novo bloco de pós-processamento que utiliza sequências-m de menor complexidade linear que a metodologia anterior. Validamos os métodos propostos empregando a bateria de teste NIST.

Palavras-chaves: Mapas caóticos, sistemas dinâmicos, geração de números aleatórios, taxa de entropia, função de autocorrelação, teste NIST.

ABSTRACT

Random number generators are widely used in scientific and technological applications. Particularly in cryptography, they are used in secret-key systems, such as key sequence generators. In this work, we present various methodologies for the design of these generators from chaotic maps. The first one is based on two techniques: Skipping and time-varying coded discretization. We show that the proposed method has higher bit generation rate when compared to fixed-time coded discretization and dispenses post-processing in order to improve their random properties. Another methodology is the use of m-sequences to eliminate the residual correlation of the coded sequence. The time-varying coded discretization has a well-defined correlation characteristic that is exploited by a new block of post-processing using m-sequences that requires less memory than the previous methodology. The effectiveness of this procedure is verified through the NIST test.

Keywords: Chaotic maps, dynamical systems, random numbers generetors, entropy rate, autocorrelation function, NIST test.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

Capítulo 1

Introdução

As propriedades inerentes aos sistemas dinâmicos caóticos, como a dependência às condições iniciais, o comportamento pseudoaleatório e o espectro faixa larga, por vezes evocadas para sua caracterização [1, 2], reservam similaridade com as funções requeridas por alguns blocos de um sistema de comunicação clássico. Nos anos 90 surgiram novas concepções de sistemas de comunicação [3] que propuseram o aproveitamento dessas propriedades para transmissão de informação de forma eficiente. Desde então, diversos trabalhos científicos propiciaram o amadurecimento da área, demonstrando a vocação de sistemas caóticos para compressão de dados [4, 5], modulação [6],[7],[8] e criptografia [9],[10].

Em decorrência da diversidade de sistemas caóticos, sua aplicação em criptografia é ubíqua, sendo utilizado em sistemas criptográficos de chave privada, geradores de números pseudo-aleatórios (PRNG, *pseudo-random number generator*), e sistemas criptográficos de chave pública [9],[11]. Contudo, certas aplicações em criptografia requerem o emprego de geradores de números aleatórios (RNG, *random number generator*). Neste caso, o núcleo do RNG deve ser um processo físico inerentemente aleatório, o que limita consideravelmente sua aplicação, como consequência da dificuldade técnica em implementa-lo. Como alternativa, os sistemas caóticos podem ser considerados como RNG's quando, apesar de sua natureza determinística se definidos sobre o conjuntos dos números reais, forem "observados" através de sequências discretas de símbolos, gerados a partir de uma partição finita do seu espaço de fase [12]. Isso equivale a empregar a dinâmica simbólica associada ao sistema, ao invés de uma sequência de números reais obtida pela interação do mapa que define o sistema caótico.

A despeito da complexidade das sequências geradas a partir de sistemas caóticos, esses são muitas vezes de baixa dimensão e definidos por recursões simples. No entanto, como decorrência da estrutura inerente aos sistemas dinâmicos, as propriedades estatísticas das sequências obtidas diretamente dos sistemas caóticos precisam comumente ser incrementadas para que possam ser atribuídas a um RNG. Em [13], três classes de quantificadores para essas propriedades são listadas: (i) baseados em teoria da informação, (ii) baseados em gráficos de recorrência, e (iii) baseados em técnicas computacionais. Os quantificadores propostos visam avaliar basicamente a uniformidade da distribuição invariante do sistema caótico e a independência entre iterações do mapa [13], que podem ser melhorados significativamente por técnicas de randomização de dinâmica simbólica [14].

Duas técnicas adotadas para geração de sequências aleatórias a partir de mapas caóticos são a discretização e o salto de amostras [14]. A análise destas sequências demonstra que o discretização atua tanto na uniformização da distribuição invariante quanto na redução de correlação. Já o salto de amostras só atua na redução de correlação, mantendo a distribuição invariante inalterada. Contudo, a redução da correlação é bem mais contundente ao emprego-se o salto de amostras que a discretização [13, 14]. A entropia condicional constitui um meio para verificar a eficácia destas técnicas, já que indica a uniformidade da distribuição invariante e a correlação da sequência [15]. De fato, a entropia condicional pode ser interpretada como um quantificador relevante [16], por refletir propriedades estatísticas das sequências, servindo como guia para adoção de estratégias que melhorem estas propriedades.

1.1 Objetivos e Contribuições da dissertação

O objetivo deste trabalho é a geração de sequências pseudo-aleatórias utilizando mapas caóticos. A saída do mapa caótico é transformada em uma sequência binária utilizando um processo de codificação. A correlação residual entre os símbolos desta sequência binária requer a utilização de uma unidade de pós-processamento para a geração de um PRNG. Este trabalho contribui com novas estratégias de codificação e pós-processamento. Para testar a aleatoriedade das sequências geradas empregamos a versão 800-22 do teste NIST.

As contribuições deste trabalho são:

- Especificação de como particionar o espaço de fase, com a proposição de um esquema de codificação que garanta uma distribuição de probabilidade uniforme da sequência gerada, denominado de discretização codificada variante no tempo. Esta codificação reduz a correlação da sequência discretizada, diminuindo os requerimentos da unidade de pós-processamento.

- ▷ Empregar a entropia condicional para determinar o limiar inferior do salto de amostras para a quebra da correlação na sequência caótica.
- ▷ Propor técnicas de dimensionamento de uma unidade de pós-processamento baseada em registradores de deslocamento com realimentação linear (LFSR, *Linear feedback shift register*) para gerar PRNG's .
- ▷ Optimização da técnica de pós-processamento por LFSR através do emprego da função de autocorrelação da sequência codificada.

1.2 Aplicações de Sistemas Caóticos

O precursor da teoria do caos foi o matemático francês Henri Poincaré. Em seu estudo sobre um problema específico de três corpos em mecânica celeste, Poincaré percebeu que um comportamento complexo (sensibilidade às condições iniciais e número infinito de órbitas periódicas) poderia aparecer em sistemas com número pequeno de graus de liberdade [17]. Ele também foi o primeiro a fornecer uma série de ferramentas para o estudo sistemático do caos, tais como, topologia, equações diferenciais, entre outras.

Em 1961, Edward Lorenz, um meteorologista do MIT, fez uma descoberta chave para difusão da teoria do caos. Trabalhando com um modelo climático simplificado, ele verificou que um sistema determinístico com três graus de liberdade poderia gerar sinais aperiódicos que apresentam sensibilidade às condições iniciais [18]. Posteriormente importantes descobertas começaram a aparecer em diferentes áreas. Na área da biologia, Robert May mostrou que um sistema discreto simples com um grau de liberdade pode levar a comportamento irregular e imprevisível [19]. Na área da astronomia, Jack Sabedoria demostra as trajetórias caóticas de asteroides [20]. Na área da medicina, Agnessa Babloyantz analisando os dados de um electroencefalograma do cérebro humano durante o ciclo do sono, revelou a existência de atratores caóticos no sono [21]. Nos últimos anos, numerosos experimentos encontraram sintomas de comportamento caótico em várias áreas do conhecimento. A aplicação de caos no campo da engenharia elétrica e eletrônica, e em particular na área das comunicações, é recente, e é favorecido por três descobertas chave [22]:

- ▷ Implementação e caracterização de vários circuitos eletrônicos caóticos na década de 80 [23], [24].
- ▷ A descoberta de sincronização caótica por Pecora e Carroll em 1990 [25], que forneceu um mecanismo para aplicação de sistemas caóticos em comunicações.

▷ A descoberta e desenvolvimento de técnicas de controle de caos por Ott, Yorke e Grebogi [26], que permitem induzir o sistema caótico a seguir trajetórias específicas do atrator.

Uma das aplicações dos sistemas caóticos é a geração de sequências pseudo-aleatórias. A ideia fundamental de sistemas de cifragem mediante caos, chamados de "sistemas de comunicação seguros baseados em caos", é a utilização de um sistema dinâmico em regime caótico para gerar uma sequência pseudo-aleatória e combiná-la com a mensagem para produzir um texto cifrado que é transmitido por um canal inseguro. Em seguida, usando a sincronização destes sistemas, o receptor reproduz um sinal pseudo-aleatório e combina-o com o sinal recebido para se recuperar a mensagem original.

Este tipo de cripto-sistema tenta aproximar o conhecido sistema de cifra em fluxo ou cifra Vernam [27]. Este cifrador consiste em fazer uma soma módulo dois entre o texto claro e uma sequência binária pseudo-aleatória (sequência cifrante), a sequência resultante é transmitida pelo canal inseguro. A operação de decodificação é realizada por uma soma modulo-2 entre a sequência recebida e a sequência cifrante, o que resulta no texto claro.

1.3 Organização

Esta dissertação esta organizada em seis capítulos.

O **Capítulo 1** apresenta a motivação, objetivos, contribuições da dissertação.

O **Capítulo 2** apresenta uma introdução à teoria do caos, os conceitos básicos dos mapas caóticos unidimensionais e suas características.

O **Capítulo 3** apresenta as diferentes formas de codificação das amostras caóticas e as unidades e pós-processamento necessárias para a geração de PRNG's. Apresentam-se os resultados e faz-se comparação entre os diferentes métodos de codificação e pós-processamentos realizados. Apesar dos métodos considerados serem independentes do mapa caótico, empregamos para o estudo de caso o mapa tangente hiperbólico [28].

O **Capítulo 4** apresenta uma nova unidade de pós-processamento para a discretização variante no tempo.

O **Capítulo 5** estende o análise dos capítulos anteriores para outros mapas caóticos.

O **Capítulo 6** apresenta as conclusões deste trabalho.

CAPÍTULO 2

REVISÃO DE SISTEMAS CAÓTICOS

ESTE capítulo faz uma revisão de sistemas caóticos, concentrando-se em mapas caóticos unidimensionais e suas características. Informações detalhadas sobre sistemas caóticos podem ser obtidas em [2].

2.1 Sistemas dinâmicos caóticos

Um sistema é um conjunto de componentes (físicos, biológicos, mecânicos, etc.) que interagem de forma definida. Exemplos: o mercado interno de um país, o sistema circulatório do corpo humano, o conjunto suspensão e amortecimento de um automóvel, um circuito eletrônico, uma coluna de destilação, uma caldeira industrial, o ecossistema de um lago etc. Um conjunto de equações é também um sistema, porém, para evitar confusão, o termo "sistema" é muitas vezes reservado para o processo real, enquanto uma equação ou conjunto de equações recebe a denominação de "modelo". Quando este é interpretado como uma representação daquele, recebe a denominação de modelo do sistema [29].

Denota-se sistema dinâmico aquele cuja saída no instante "k" depende de valores passados da saída e da entrada. Portanto, os sistemas dinâmicos possuem memória e são descritos por equações diferenciais no caso de sistemas contínuos, e por equações de diferença no caso de sistemas discretos. O estado de um sistema dinâmico é um conjunto de variáveis, podendo ou não estar associadas a quantidades físicas, cuja especificação (na ausência de excitações externas) determina completamente a evolução do sistema. Tal conjunto de variáveis, apesar de ter cardinalidade definida, não é

unicamente especificado e está associado à memória do sistema dinâmico. Este conjunto recebe a denominação de vetor de estado quando é representado através de um espaço de dimensão n, portanto, contém o valor de todas as variáveis dependentes do sistema dinâmico num dado instante de tempo ou iteração. O conjunto de todos os possíveis vetores de estados é conhecido como *espaço de fase*. Se a regra for aplicada em tempos discretos, este é chamado um sistema dinâmico de tempo discreto, também denominados de *mapas*.

O comportamento dos mapas é observado mediante uma série temporal discreta $\{x_i\}_{i=0}^{\infty}$, obtida pela iteração de uma função $f(x)$, não linear, sob uma condição inicial x_0, da seguinte forma:

$$x_n = f(x_{n-1}), n = 1, 2, 3, ... \tag{2.1}$$

Uma *órbita* de x_0 é um conjunto de pontos $\{x_0, f(x_0), f^2(x_0) ...\}$, em que f^k denota a k-th composição de $f(x)$ [2]. Um ponto fixo x_k é o ponto no qual se tem $f(x_k) = c$, para todo $k \in \mathbb{R}$ sendo c uma constante[2]. Uma órbita periódica tem período k, se $f^k(x_0) = x_0$, ou seja, os primeiros pontos são $\{x_0, f(x_0), f^2(x_0), ..., f^{k-1}(x_0), x_0\}$, onde $f^p(x_0) \neq x_0$, para todo $p = \{1, 2, ..., k-1\}$ [2].

Pode-se representar graficamente uma órbita pelo método conhecido como diagrama cobweb, uma representação do comportamento global do sistema, particularmente útil na análise de sistemas não lineares. O processo consiste em projetar verticalmente a amostra x_0 sob o gráfico da função $f(x)$ e horizontalmente sob o gráfico de $y = x$. Repetindo este processo n vezes obtêm-se os pontos da órbita, formando-se os pares $\{(0, x_0), (x_0, x_1), (x_1, x_1), (x_1, x_2), (x_2, x_2), (x_2, x_3) \cdots\}$. Neste método é possível reconhecer visualmente o comportamento do sistema. Exemplos deste diagrama são apresentados na seção 2.4.

A seguir define-se o conceito de expoente de Lyapunov que quantifica a sensibilidade de um sistema às condições iniciais que costuma ser empregado para caracterizar um comportamento caótico.

2.2 Quantificação da dinâmica caótica

O número de Lyapunov é utilizado para quantificar a taxa média de separação multiplicativa entre duas órbitas infinitesimalmente próximas com condições iniciais x_0 e $x_0+\epsilon$. O expoente de Lyapunov é simplesmente o logaritmo natural do número de Lyapunov. Um número de Lyapunov igual a 2 (ou equivalentemente um expoente de Lyapunov $\ln 2$) significa que a distância entre a órbita iniciada x_0 e a órbita muito próxima iniciada em $x_0 + \epsilon$ é duas vezes maior por cada iteração, em média.Um número de Lyapunov $1/2$ significa que a distância é reduzida para metade em cada iteração, e as órbitas iniciadas em x_0 e em $x_0 + \epsilon$ ficaram mais perto uma da outra.

O número de Lyapunov $L(x_0)$ de uma orbita gerada por um mapa unidimensional $f(x)$ está definido como [2]:

$$L(x_0) = \lim_{n \longrightarrow \infty} (|f'(x_0)|...|f'(x_{n-1})|)^{1/n} \tag{2.2}$$

se o limite existe. O expoente de Lyapunov $h(x)$ é definido como [2]:

$$h(x_0) = \lim_{n \longrightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)|, \tag{2.3}$$

se o limite existe. Note-se que h existe se, e somente se, L existe e é diferente de zero, de modo que $h = \ln L$.

Uma órbita de x_0 é assintoticamente periódica, se está converge para uma órbita periódica quando $n \longrightarrow \infty$. Isto significa que existe uma órbita periódica $\{y, f(y), f^2(y)..., f^k(y), y, f(y), f^2(y)...\}$, tal que:

$$\lim_{n \longrightarrow \infty} |x_n - y_n| = 0. \tag{2.4}$$

Uma órbita é caótica se [2]:

- O expoente de Lyapunov é maior que zero.
- A órbita não é assintoticamente periódica.

Estes parâmetros quantificam a dinâmica caótica de sistemas, mas não há uma definição formal do caos. Entretanto, reconhece que um sistema para ser caótico deve satisfazer as seguintes propriedades [1]:

- **Comportamento aperiódico a longo prazo**: Existem órbitas que não tendem a pontos fixos, órbitas periódicas, ou quase-periódicas quando $t \to \infty$.
- **Determinístico**: O comportamento irregular é dado pela não linealidade do sistema, não por a ruídos externos, ou seja, o sistema deve ser inerentemente determinístico.
- **Sensibilidade à condições inicias**: Órbitas com condições inicias próximas separam-se exponencialmente (quantificado pela expoente de Lyapunov).

2.3 Exemplos de Mapas Caóticos

Nesta seção, são analisados três mapas caóticos unidimensionais:

▷ Mapa Logístico.

▷ Mapa da Tenda.

▷ Mapa Tangente Hiperbólica.

Os dois primeiros mapas (logístico, tenda) são amplamente conhecidos na literatura, enquanto o mapa tangente hiperbólica foi introduzido em [28].

2.3.1 Mapa Logístico

Para uma família de mapas logísticos, $f(x) = ax(1-x)$, tem-se que [2]:

▷ Se o parâmetro a está na faixa $0 < a \leq 1$, o mapa só tem um ponto fixo em $p = 0$, como mostra a Figura 2.1.

▷ Se o parâmetro a está na faixa $1 < a \leq 3,5$, o mapa tem dois pontos fixos em $p = 0$ e $p = \frac{a-1}{a}$, como mostra a Figura 2.2.

▷ Se o parâmetro a está na faixa $3,5 < a \leq 4$, o mapa gera caos, como mostra a Figura 2.3.

A família de mapas logísticos tem um comportamento caótico a partir de um valor do parâmetro a. O valor critico a_c serve como fronteira entre a zona em que ocorrem os fenômenos de convergência e zona de caos. Este foi determinado empiricamente por Mitchell Jay Feigenbaum e seu valor é $a_c = 3,5699456.....$[2].

2.3.2 Mapa da tenda

O mapa da tenda $f : [0,1] \to [0, \beta/2]$ é definido como [30]:

$$f(x) = \begin{cases} \beta x, & se \quad 0 \leq x < \frac{1}{2} \\ \\ \beta(1-x), & se \quad \frac{1}{2} \leq x < 1, \end{cases} \tag{2.5}$$

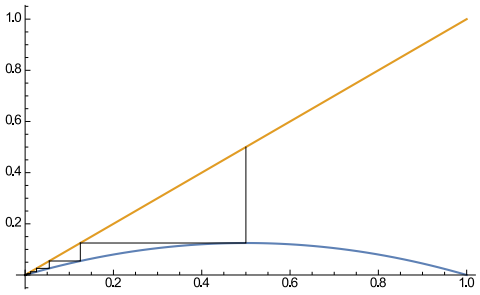


Figura 2.1: *Cobweb do mapa da logístico para* $a = 0,5$.

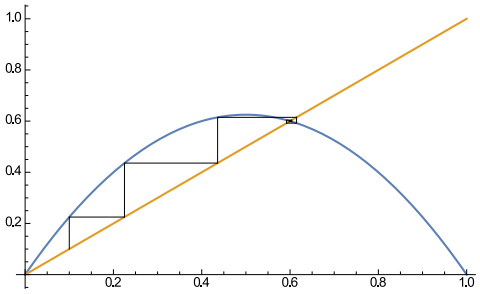


Figura 2.2: *Cobweb do mapa da logístico para* $a = 2.5$.

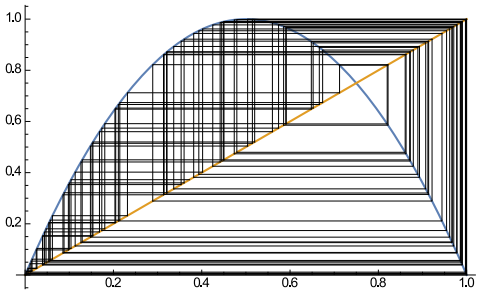


Figura 2.3: *Cobweb do mapa logístico para* $a = 4$.

em que o valor do parâmetro β está no intervalo $0 < \beta \leq 2$. Dependendo do valor de β existem seis tipos de comportamento assintótico do mapa:

- $0 < \beta \leq 1$: O mapa tem um ponto fixo em $x = 0$. Para qualquer condição inicial, as iterações convergem para este, como mostra a Figura 2.4.
- $\beta = 1$: Todos os pontos da região $[0; 0, 5]$ são pontos fixos. Os pontos da região $(0, 5; 1]$ mapeiam na região $[0; 0, 5]$, sendo pontos eventualmente fixos.
- $1 < \beta < \sqrt{2}$: As orbitas são quasi-periódicas. O mapa não ocupa o espaço de fase completo $[0, \beta/2]$, estando restrito numa região $[\beta(2 - \beta)/2, \beta/2]$, como mostra a Figura 2.5.
- $\sqrt{2} < \beta < 2$: Região caótica. O comportamento assintótico das sequências geradas se limitam na região $[\beta(2 - \beta)/2, \beta/2]$ e o comportamento nesta região é caótico, como mostra a Figura 2.6.
- $\beta = 2$: As órbitas obtidas são caóticas e cobrem o espaço completo $[0, 1]$, como mostra a Figura 2.7.

2.3.3 Mapa Tangente Hiperbólica

O mapa tangente hiperbólica (tanh) [28], $f : [-1, 1] \to [-1, 1]$, é baseado na função tangente hiperbólica e é definido por:

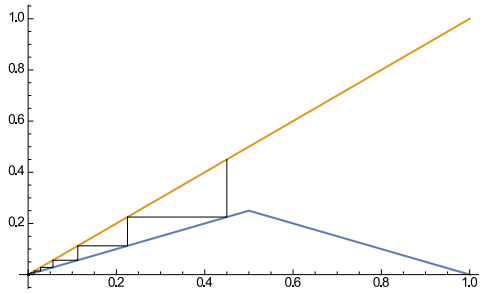


Figura 2.4: *Cobweb do mapa da tenda para* $0 < \beta \leq 1$.

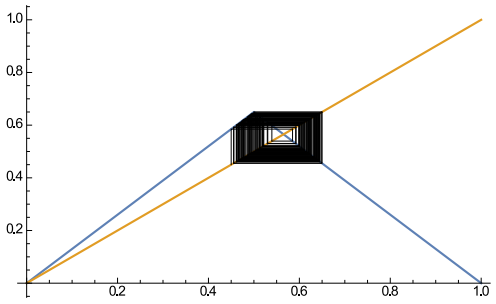


Figura 2.5: *Cobweb do mapa da tenda para* $1 < \beta < \sqrt{2}$.

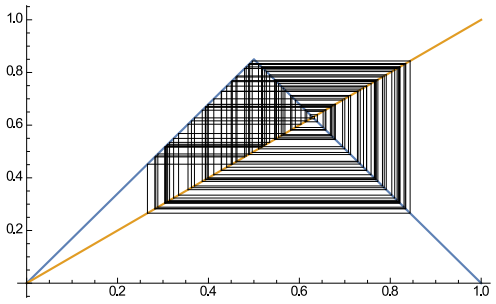


Figura 2.6: *Cobweb do mapa da tenda para* $\sqrt{2} < \beta < 2$.

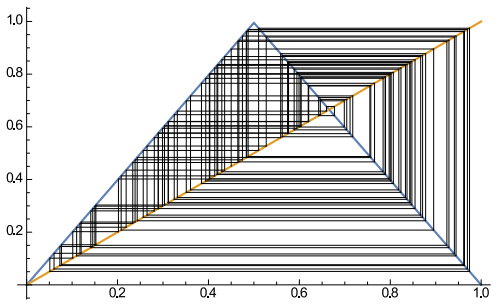


Figura 2.7: *Cobweb do mapa da tenda para* $\beta = 2$.

$$f(x) = \begin{cases} e \cdot \tanh\big(r \cdot (x+1)\big) - 1, & x < 0 \\ (-1)^b \cdot \big[e \cdot \tanh\big(-r \cdot (x-1)\big) - 1\big], & x \geq 0 \end{cases} \tag{2.6}$$

em que o fator de escala e é dado por:

$$e = \frac{2}{\tanh(r)}. \tag{2.7}$$

Este mapa tem dois parâmetros de controle especificados pela dupla (b, r). O parâmetro b define a simetria do mapa, podendo ser 0 (simetria par) ou 1 (simetria ímpar). O parâmetro r é um número real positivo que controla a extensão da região planar em torno do eixo de simetria. As Figuras 2.8 e 2.9 mostram os mapas tanh par, denominado de e-tanh, e o mapa tanh ímpar, denominado de o-tanh, para diferentes valores de r. Observa-se nas Figuras 2.8 e 2.9 que quando o parâmetro r tende a 0 ($r \to 0$), os mapas e-tanh e o-tanh tendem a dois mapas conhecidos na literatura, o mapa da tenda e o mapa de Bernoulli, respectivamente.

A concentração dos pontos do mapa tanh pode ser observada nos histogramas dos mapas e-tanh e o-tanh, como mostra as Figuras 2.10 e 2.11. Estas mostram a distribuição de pontos de uma órbita de 1 000 000 pontos para ambos mapas com $r = 3$ e condição inicial aleatória. Existe uma clara assimetria na distribuição de pontos do histograma do mapa e-tanh em torno de $x = 0$. com maior concentração de pontos em torno de $x = 1$. Para mapa e-tanh é observado que para $r < 7,1$ os

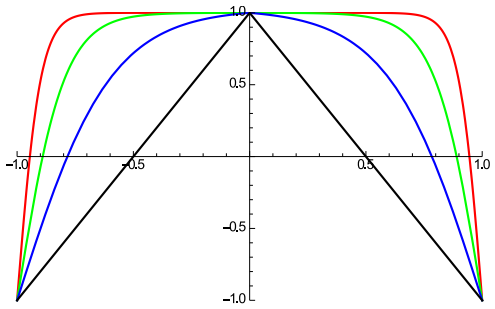


Figura 2.8: *Mapa e-tanh para* $r = \{0,01; 2,5; 5; 10\}$.

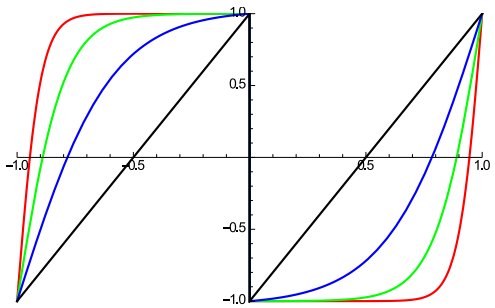


Figura 2.9: *Mapa o-tanh para* $r = \{0,01; 2,5; 5; 10\}$.

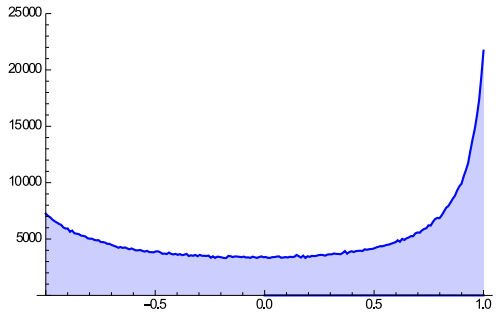


Figura 2.10: *Histograma do Mapa e-tanh, com* $r = 3$.

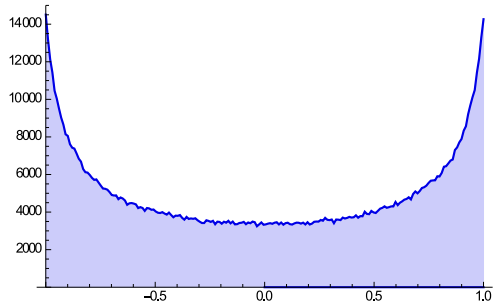


Figura 2.11: *Histograma do Mapa o-tanh, com* $r = 3$.

pontos são mais concentrados em torno a $x = 1$, e quando $r > 7,1$ esta tendência se inverte e os pontos se concentram em $x = -1$. A Figura 2.11 ilustra que o histograma do mapa o-tanh é simétrico em torno de $x = 0$. Uma característica desta família é uma concentração dos pontos da órbita em torno dos seus valores extremos, resultando em uma redução na ocorrência de valores em torno de $x = 0$ com o aumento do parâmetro r, ou seja, a probabilidade de ocorrência de pontos na região central do domínio decresce, como consequência do alargamento da região horizontal.

Para a avaliação do comportamento caótico do mapa tanh, empregamos conceitos bem conhecidos para sistemas caóticos, como o expoente de Lyapunov e o diagrama de órbita.

Expoente de Lyapunov

A Figura 2.12 mostra o valor do expoente de Lyapunov para $0 < r < 10$ para o mapa e-tanh. O expoente de Lyapunov do mapa o-tanh apresenta o mesmo comportamento do mapa e-tanh. Consideramos uma orbita de $N = 14000$ e desprezamos as primeiras 400 iterações para eliminar o transiente da órbita. Quando o parâmetro r tende a zero, o expoente de Lyapunov tende ao do mapa da tenda (0,693), e decresce suavemente quando o parâmetro r cresce. Podemos observar que este é sempre positivo, então o mapa apresenta caos para todos os valores de r na faixa considerada na Figura 2.12.

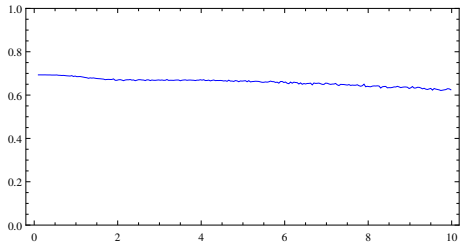


Figura 2.12: *Expoente de Lyapunov versus o parâmetro r para o mapa e-tanh.*

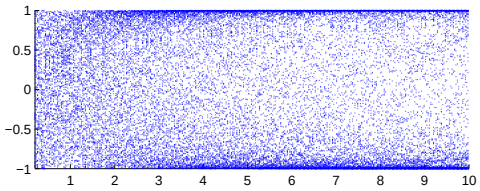


Figura 2.13: *Digrama de orbita versus o parâmetros r para o mapa e-tanh.*

Diagrama de Órbita

O comportamento caótico do mapa e-tanh para $0 < r < 10$ também é observado a partir do diagrama de órbita mostrado na Figura 2.13. Deve-se observar que os pontos se concentram em torno dos valores extremos ± 1 quando $r > 2,0$. O diagrama de órbita do mapa o-tanh apresenta o mesmo comportamento e os pontos nos dois extremos ocorrem com a mesma probabilidade.

2.4 Implementação do mapa e-tanh

Uma vantagem do mapa e-tanh é a facilidade de implementação do mapa em circuitos eletrônicos. O mapa e-tanh é reescrito a partir de (2.6) da seguinte forma:

$$f(x) = e \cdot \tanh\big[r \cdot (1 - |x|)\big] - 1. \tag{2.8}$$

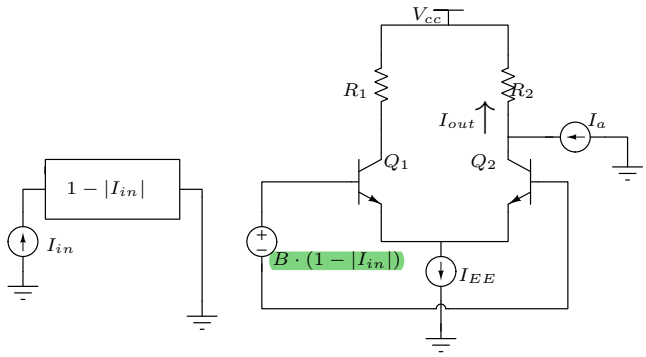


Figura 2.14: *Circuito para implementação do mapa e-tanh.*

Para implementação deste mapa, emprega-se o circuito da Fig. 2.14, que é baseado em um par diferencial com transistor bipolar de junção (TBJ). Denominamos por $v_{id} = B(1 - |I_{in}|)$ a tensão diferencial entre os terminais base de Q_1 e Q_2, I_{EE} a corrente de polarização do par diferencial e V_T a tensão térmica, em torno de $26mV$ a temperatura ambiente. A entrada do circuito é I_{in} e a saída é o inverso da corrente de coletor de Q_2 mais uma corrente constante I_a, ou seja, $I_{out} = -I_2 + I_a$. Pode-se mostrar que I_2 é função da tangente hiperbólica da tensão diferencial por um fator de escala $2 \cdot V_T$ [31], o que permite especificar I_{out} da seguinte forma

$$I_{out} = \frac{I_{EE}}{2} \cdot \tanh\left(\frac{v_{id}}{2 \cdot V_T}\right) - \left(\frac{I_{EE}}{2} - I_a\right). \tag{2.9}$$

Para especificar (2.8) em termos dos parâmetros do circuito, a corrente I_{out} deve ser normalizada. Aplicando esse fator de normalização para toda a equação, a transresistência B é escolhida para gerar o respectivo argumento da tangente hiperbólica em (2.8) e a corrente I_a é escolhida para que o segundo termo do lado direito da equação (2.9) seja igual a unidade quando normalizado.

Com o auxílio de um circuitos periféricos apropriado é possível gerar uma sequência do mapa e-tanh. Entre esses periféricos, destaca-se o papel do circuito *sample-and-hold* [32], que permite gerar uma sequência caótica a partir do núcleo apresentado na Fig. 2.14. Isso ocorre com a amostragem de I_{out} em um ciclo de máquina (sample) e o emprego deste como entrada no ciclo subsequente.

Aproveitando que o mapa tanh é um mapa implementável, o próximo capítulo foca na geração de sequências pseudo-aleatórias a partir deste mapa.

CAPÍTULO 3

GERAÇÃO DE PRNG MEDIANTE MAPA CAÓTICO

NESTE capítulo propomos técnicas para o projeto de RNG a partir de mapas caóticos e introduzimos técnicas de discretização codificada variante no tempo. Apesar dos métodos considerado serem independente do mapa caótico, empregamos para o estudo de caos o mapa tanh. Para testar a aleatoriedade das sequências geradas empregamos a entropia condicional e o teste NIST.

3.1 Entropia Condicional

A entropia condicional é uma medida importante para determinar a aleatoriedade de uma sequência binária [16]. Seja $\{Z_n\}_{n=1}^{\infty}$ uma sequência de variáveis aleatórias binárias estacionárias. Define-se a entropia condicional de Z_n dado os $(n-1)$ últimos símbolos Z^{n-1} ($Z^{n-1} = Z_{n-1}Z_{n-2}\cdots Z_1$) da seguinte forma [15]

$$H(Z_n \mid Z^{n-1}) = \sum_{z^n \in \{0,1\}^n} \Pr(z^n) \log_2 \left(\frac{1}{\Pr(z_n \mid z^{n-1})} \right).$$

A entropia condicional $H(Z_n \mid Z^{n-1})$ é uma função não crescente com n para um processo estacionário e converge para a taxa de entropia H [15]

$$H = \lim_{n\to\infty} H(Z_n \mid Z^{n-1}).$$

Uma sequência aleatória ideal deve atingir $H(Z_n \mid Z^{n-1}) = 1$ para todos os valores de n. Para gerar uma sequência binária $\{Z_n\}$ a partir da família de mapas tanh, procedemos da seguinte forma. Fixamos um valor de r e uma condição inicial x_0 e obtemos uma órbita finita usando $x_n = f(x_{n-1}), n =$

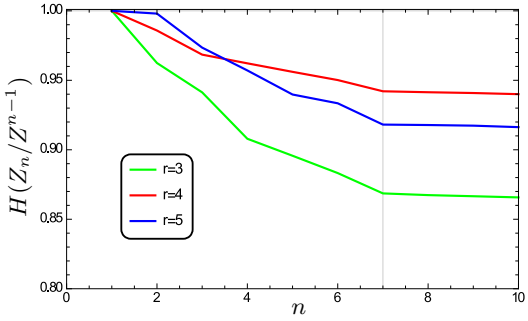


Figura 3.1: *$H(Z_n \mid Z^{n-1})$ versus n para mapa e-tanh para $r = 3, 4, 5$.*

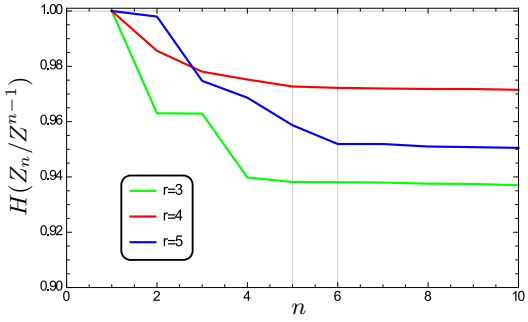


Figura 3.2: *$H(Z_n \mid Z^{n-1})$ versus n para mapa o-tanh para $r = 3, 4, 5$.*

$1, 2, \ldots$. Os primeiros 400 pontos gerados são descartados devido ao transiente inicial da órbita. O intervalo $[-1, 1]$ é particionado em duas regiões $\mathcal{R}_0 = [-1, \varepsilon)$ e $\mathcal{R}_1 = [\varepsilon, 1]$ satisfazendo $\Pr(x_n \in \mathcal{R}_0) = \Pr(x_n \in \mathcal{R}_1) = 1/2$, tal que se $x_n \in \mathcal{R}_0$ então $z_n = 0$, ou se $x_n \in \mathcal{R}_1$ então $z_n = 1$.

As seguintes Figuras 3.1 e 3.2 mostram $H(Z_n \mid Z^{n-1})$ versus n para uma sequência binária gerada por simulação para os mapas e-tanh e o-tanh, respectivamente, para diferentes valores do parâmetro r. Observa-se em todas as curvas mostradas que a entropia condicional decresce até um valor de n, e após este valor converge para a taxa de entropia. Denomina-se de n^* este valor limite de n a partir do qual a entropia condicional permanece constante, e igual a H. Por exemplo, observamos na Figura 3.1 que $n^* = 7$.

Observe-se nas curvas da Figura 3.1 o valor da taxa de entropia tem valores $H = 0,867$ para $r = 3$, $H = 0,94$ para $r = 4$ e $H = 0,917$ para $r = 5$. A incerteza sobre uma variável Z_n decresce com o conhecimento de valores passados até $n = 7$, o que implica a presença de uma memória finita na sequência binária. Não espera-se uma correlação relevante entre as variáveis Z_n e Z_{n+m} para $m > 7$, para todos os valores de r considerados.

As curvas da Figura 3.2 mostram que a entropia condicional do mapa o-tanh decresce até $n^* = 5$ para $r = 3$ e 4, já para $r = 5$ o valor de n^* é 6. Os valores da taxa de entropia neste caso são

$H = 0,937$ para $r = 3$, $H = 0,971$ para $r = 4$ e $H = 0,950$ para $r = 5$.

A próxima seção descreve o procedimento de geração de um RNG proposto nesta dissertação a partir da família de mapas tanh.

3.2 Geração de um RNG Baseado na família Mapa tanh

A geração de um RNG é composta de três etapas: geração de uma órbita finita a partir de um mapa caótico, emprego da técnica de saltos de amostras [14] para reduzir a correlação da sequência caótica e, finalmente, a sequência binária é gerada usando a técnica de discretização codificada (fixa no tempo ou variante no tempo). Como se mostra no diagrama de blocos da Figura 3.3.

O salto de amostras tem como entrada uma órbita finita gerada pelo mapa $\{x_k\}_{k=1}^{N}$ e produz na saída $\{x'_k\}_{k=1}^{N'} = \{x_{k \cdot p}\}_{k=1}^{N'}$, para um valor fixo do parâmetro p que especifica o salto de amostras. Se for aplicado o procedimento usual de mapear a sequência de saída em uma partição com duas regiões (conforme discutido na seção anterior), a sequência binária também tem comprimento p vezes menor que a sequência caótica de entrada, o que compromete a taxa de *bits* por amostra caótica gerada na saída do codificador.

Para aumentar esta taxa, propomos particionar o intervalo $[-1, 1]$ em 2^q regiões, $\mathcal{R}_i$, $i = 0, \cdots 2^q - 1$, de tal forma que $\Pr(x_k \in \mathcal{R}_i) = 1/2^q$. Por exemplo, para o mapa e-tanh com $r = 3$, as quatro regiões (para $q = 2$) são [-1,-0.65), [-0.65,0.19), [0.19,0.87), [0.87,1]. Para o mapa o-tanh estas regiões são, [-1, -0.77), [-0.77, 0), [0, 0.77), [0.77, 1). Esta partição reflete a concentração do histograma em valores extremos do mapa, como mostrado nas Figuras 2.10 e 2.11. Para o mapa o-tanh mostra que as duas regiões dos extremos e as duas regiões internas são iguais. Enquanto o mapa e-tanh não apresenta o mesmo comportamento, devido a acumulação de mais pontos em uma extremidade que na outra. Cada região R_i é rotulada com uma sequência binária de q *bits*. O emprego do salto de amostras em conjunto com a discretização de 2^q regiões produz uma taxa de *bits* por amostra caótica, denotado de R_x, da seguinte forma:

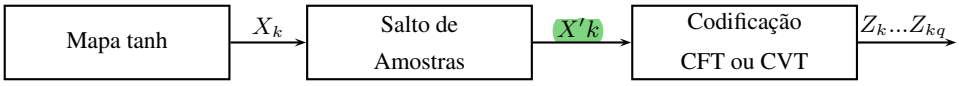


Figura 3.3: *Diagrama de bloco do esquema da geração de RNG.*

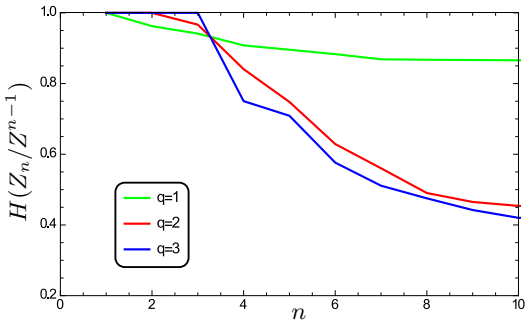


Figura 3.4: $H(Z_n \mid Z^{n-1})$ *versus* n *para mapa e-tanh para* $r = 3$, $p = 1$ *e* $q = 1, 2, 3$, *com CFT.*

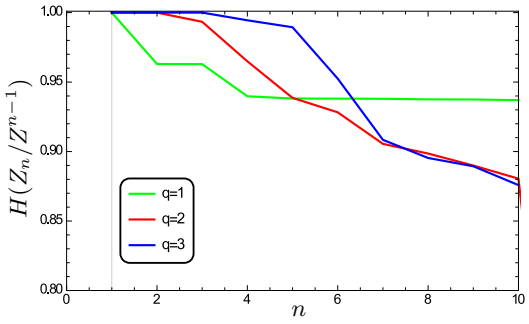


Figura 3.5: $H(Z_n \mid Z^{n-1})$ *versus* n *para mapa o-tanh para* $r = 3$, $p = 1$ *e* $q = 1, 2, 3$, *com CFT.*

$$R_x = q/p. \tag{3.1}$$

Por exemplo, para $R_x = 1$ a sequência binária e a sequência caótica têm o mesmo comprimento, ou seja, gera-se a mesma quantidade de *bits* por amostras que os saltos dados. Por outro lado, o particionamento em duas regiões implica que $R_x = 1/p$.

O processo de codificação consiste em atribuir a cada região uma sequência de *bits* (denominada de sequência código). Por exemplo, uma codificação fixa das regiões, denominada de CFT, para $q = 2$ é $(00, 01, 10, 11)$. Por exemplo, se $x'_k \in \mathcal{R}_0$ então a sequência binária gerada neste intervalo é 00, se $x'_k \in \mathcal{R}_1$ a sequência gerada é 01, e assim sucessivamente. O emprego de $q > 1$ propicia o surgimento de uma correlação entre os *bits* das sequências código de cada região, o que enfraquece a aleatoriedade da sequência binária gerada.

Este comportamento é observado nas curvas de entropia condicional mostradas nas Figuras 3.4 e 3.5. Nestas curvas, fixa-se os parâmetros do $r = 3$ e $p = 1$ (sem saltos de amostras), e considera-se três valores de q. Observe-se que a taxa de entropia decresce e o valor de n^* cresce com o aumento de q para os mapas e-tanh (Figura 3.4) e o-tanh (Figura 3.5).

Na Figuras 3.4 e Figura 3.5 se observa o comportamento esperado, a codificação CFT introduz

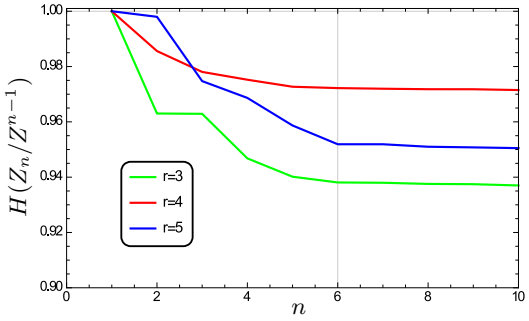


Figura 3.6: $H(Z_n \mid Z^{n-1})$ *versus* n *para mapa e-tanh para* $r = 3, 4, 5$*, com CVT.*

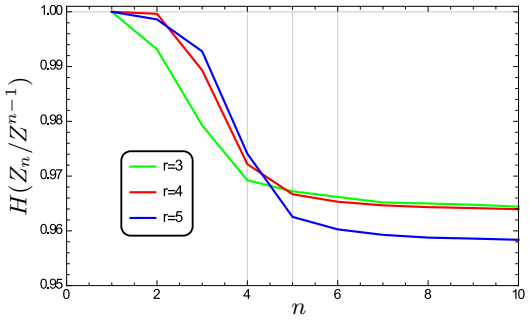


Figura 3.7: $H(Z_n \mid Z^{n-1})$ *versus* n *para mapa o-tanh para* $r = 3, 4, 5$*, com CVT.*

uma correlação a mais na sequência binária de saída. Para diminuir o efeito desta correlação propõe-se uma codificação variante no tempo , denominada de CVT, que consiste em um deslocamento cíclico de q *bits* para esquerda entre os *bits* que rotulam regiões adjacentes. Por exemplo, se para a n-ésima amostra caótica a codificação é $(00, 01, 10, 11)$, para a próxima amostra a codificação passa a ser $(01, 10, 11, 00)$. As curvas de entropia condicional utilizando a CVT são mostradas nas Figuras 3.6 e 3.7. Utilizam-se os mesmos valores de r das Figuras 3.1 e 3.2, e parâmetros de $p = 1$ e $q = 1$.

Esta codificação apresenta um aumento de taxa de entropia. O valor de n^* para o mapa e-tanh (Figura 3.6) é igual a 6, e na Figura 3.7 note-se que $n^* = 4$ para $r = 3$, e $n^* = 5$ para $r = 4$ e 5. Por exemplo, comparando-se as Figuras 3.2 e 3.7 para $r = 3$, pode-se observar que o valor de n^* é menor utilizando a codificação CVT. Na Tabela 3.1 se faz um resumo dos valores de n^* para os diferentes valores de r utilizados. Também obtemos uma taxa de entropia melhor para CVT ($H = 0,9423$) que para CFT ($H = 0,8756$).

A seguir, empregaremos a bateria de testes NIST para analisar a aleatoriedade de sequências binárias geradas com o procedimento proposto nesta seção. Para cada valor de r, discutiremos valores apropriados de p e q que quando associados às técnicas CFT e CVT produzem RNG's com maior

Tabela 3.1: *Valores de n^* para os mapas e-tanh e o-tanh com $r = 3, 4, 5$.*

		mapa e-tanh	mapa o-tanh
r=3	CFT	7	5
	CVT	6	4
r=4	CFT	7	5
	CVT	6	5
r=5	CFT	7	6
	CVT	6	5

taxa R_x.

3.3 Teste de Aleatoriedade NIST

Para concluir se uma estratégia de geração um RNG é criptograficamente segura, as sequências geradas devem ser submetidas a uma variedade de testes estatísticos projetados para detectar características específicas esperadas de sequências aleatórias. Existem várias opções disponíveis de testes estatísticos, entre as quais destacamos: NIST [33], DIEHARD [34], testes de Knuth [35].

Neste trabalho empregaremos o NIST (versão 800-22) que é um pacote que compreende 15 testes estatísticos baseado em teste de hipóteses e é largamente usado na literatura. No Apêndice A, apresenta-se uma breve explicação dos testes estatísticos realizados no NIST.

Cada teste é utilizado para determinar a aceitação ou rejeição da hipótese que detecta um desvio da aleatoriedade ideal e gera um parâmetro $P\text{-}value$ que indica a probabilidade de um gerador de números aleatórios ideal produzir uma sequência menos aleatória que a sequência testada. Este é calculado com nível de significância α, tipicamente compreendido no intervalo [0,001;0,01]. Se $P\text{-}value$ é igual a 1, então a sequência parece ter aleatoriedade perfeita, enquanto $P\text{-}value$ igual a 0 indica que a sequência parece ser completamente não-aleatória (para um dado teste). Para o valor adotado, $\alpha = 0,01$, uma sequência é aprovada com nível de significância de 99%.

Nas simulações realizadas, a sequência binária de entrada do NIST tem comprimento 524288000 (formada pela concatenação de 500 subsequências de comprimento 1048576 geradas com condições iniciais escolhidas aleatoriamente). O teste fornece como resultado o valor de $P\text{-}value$ bem como a razão de subsequências aprovadas para cada teste da bateria NIST. A razão mínima para que a sequência passe em cada teste é calculada usando o procedimento mostrado em [33] que depende do número de subsequências e do valor de α. Para os valores considerados, esta é 0,9767. Portanto, se $P\text{-}value$ é maior que α e a razão é maior que 0,9767 então a sequência é considerada aprovada em

Tabela 3.2: *Maiores taxas R_x alcançadas com CFT e CVT, para o mapa e-tanh com $r = 3, 4, 5$.*

Salto	$r = 3$		$r = 4$		$r = 5$	
	CVT	CFT	CVT	CFT	CVT	CFT
$p = 6$	5/6	X	5/6	X	4/6	X
$p = 7$	7/7	1/7	7/7	X	5/7	X
$p = 8$	–	2/8	–	1/8	–	X
$p = 9$	–	3/9	–	–	–	1/9

Tabela 3.3: *Maiores taxas R_x alcançadas com CFT e CVT, para o mapa o-tanh com $r = 3, 4, 5$.*

Salto	$r = 3$		$r = 4$		$r = 5$	
	CVT	CFT	CVT	CFT	CVT	CFT
$p = 3$	1/3	X	1/3	X	1/3	X
$p = 4$	4/4	X	4/4	X	3/4	X
$p = 5$	–	1/5	–	1/5	–	X
$p = 6$	–	5/6	–	4/6	–	2/6

um teste, caso contrário, a sequência é rejeitada.

A Tabela 3.2 apresenta as maiores taxas R_x alcançadas, com codificação CFT e CVT, que passaram no teste NIST para cada valor do salto p para o mapa e-tanh para três valores de r. A Tabela 3.3 apresenta a mesma análise para o-tanh. Nestas tabelas, o símbolo X indica que o gerador não passa no teste para os parâmetros indicados, enquanto o símbolo "–" indica que não é possível aumentar a taxa em relação à mostrada na mesma coluna para um valor menor de p. O emprego de CVT permite aumentar substancialmente a taxa, como por exemplo é observado na Tabela 3.2. Para $r = 3$ e $p = 7$ no mapa e-tanh, obtemos uma taxa de *bits* $R_x = 1/7$ que é obtida com a codificação CFT. Utilizando a codificação variante no tempo (CVT) obtêm-se uma taxa $R_x = 7/7$. O gerador não passa no teste para $p = 7$ e codificação fixa quando utilizamos o parâmetro $r = 4$ e $r = 5$. O aumento do salto para $p = 8$ ou $p = 9$ ainda permite obter um bom gerador com codificação fixa para $q > 1$, apesar de uma baixa taxa de *bits*. Observa-se que é possível obter um gerador com taxa menor que 1 que passe nos testes com $p = 6$ e CVT. O esquema CVT atinge a maior taxa (unitária) com $p = 7$ para os valores de $r = 3$ e $r = 4$, para $r = 5$ não chega-se obter a taxa de *bits* unitária. O aumento da taxa propiciado pela codificação CVT tambem é observado na Tabela 3.3

Como se pode observar o mapa o-tanh perde a correlação mais rápido que o mapa e-tanh. A taxa R_x unitária é atingida com um valor de saltos menor para o mapa o-tanh ($p = 4$) que para o mapa e-tanh ($p = 7$) para o mesmo valor de $r = 3$ e utilizando a CVT. A Tabela 3.4 ilustra que a taxa de entropia deve ter um valor muito elevado para que a sequência binaria passe em todos os testes NIST.

Tabela 3.4: *Taxa de Entropia para codificação CFT.*

Mapa.	Parâmetros (q e p)	Entropia Condicional	Resultado
Mapa e-tanh CFT	$q = 1\ p = 5$	H= 0,9997	Não Sucesso
	$q = 1\ p = 6$	H= 0,9997	Não Sucesso
	$q = 1\ p = 7$	H= 0,9999	Sucesso
Mapa o-tanh CFT	$q = 1\ p = 3$	H= 0,9990	Não Sucesso
	$q = 1\ p = 4$	H= 0,9995	Não Sucesso
	$q = 1\ p = 5$	H= 0,9999	Sucesso

Convém ressaltar que a taxa de entropia para estas sequências é maior que $H = 0,9999$ para todos os casos sendo um indicador adicional da aleatoriedade da sequência binária gerada.

Apresentam-se nas Tabelas 3.5, 3.6, 3.7 e 3.8 os resultados dos testes NIST realizados para as sequências geradas pela família de mapas tanh. Fixamos o parâmetro $r = 3$ e mostramos as melhores taxas encontradas para as duas codificações. Este gerador passa em todos os testes realizados, portanto é um RNG de alta qualidade de acordo com o teste NIST.

Convém ressaltar que $n^* = 7$ e $n^* = 5$ é o valor do salto indicado pela entropia condicional mostrada nas Figuras 3.1 e 3.2, para a quebra da correlação das sequências binárias geradas pelos

Tabela 3.5: *P-value e a razão de testes aprovados para cada teste do NIST, para o mapa e-tanh com $r = 3$, $R_x = 7/7$, com CVT.*

Teste Estatístico	P-value	Aprovados	Resultado
Frequency	0,353733	0,98800	Sucesso
Block Frequency	0,041169	0,98600	Sucesso
Runs	0,911413	0,98800	Sucesso
Longest-Run-of-Ones	0,448424	0,99600	Sucesso
Binary Matrix Rank	0,021407	0,99600	Sucesso
FFT	0,197981	0,98200	Sucesso
Non-Overlapping Template	0,603841	0,99400	Sucesso
Overlapping Template	0,682823	0,99000	Sucesso
Universal	0,150340	0,99200	Sucesso
Linear Complexity	0,463512	0,98400	Sucesso
Serial	0,649612	0,98800	Sucesso
Approximate Entropy	0,073201	0,98800	Sucesso
Random Excursions	0,023545	0,98200	Sucesso
Random Excursions Variant	0,122325	0,98800	Sucesso
Cumulative Sums	0,767582	0,99600	Sucesso

Tabela 3.6: *P-value e a razão de testes aprovados para cada teste do NIST, para o mapa e-tanh com $r = 3$, $R_x = 1/7$, com CFT.*

Teste Estatístico	P-value	Aprovados	Resultado
Frequency	0,350485	0,99500	Sucesso
Block Frequency	0,035174	0,98800	Sucesso
Runs	0,914463	0,99500	Sucesso
Longest-Run-of-Ones	0,534146	0,99600	Sucesso
Binary Matrix Rank	0,739918	0,99800	Sucesso
FFT	0,197981	0,98500	Sucesso
Non-Overlapping Template	0,911413	0,98900	Sucesso
Overlapping Template	0,213309	0,98900	Sucesso
Universal	0,122325	0,99100	Sucesso
Linear Complexity	0,535129	0,98600	Sucesso
Serial	0,901653	0,99800	Sucesso
Approximate Entropy	0,350485	0,99800	Sucesso
Random Excursions	0,125695	0,98500	Sucesso
Random Excursions Variant	0,325585	0,99300	Sucesso
Cumulative Sums	0,534146	0,99100	Sucesso

Tabela 3.7: *P-value e a razão de testes aprovados para cada teste do NIST, para o mapa o-tanh com $r = 3$, $R_x = 4/4$, com CVT.*

Teste Estatístico	P-value	Aprovados	Resultado
Frequency	0,911413	0,98100	Sucesso
Block Frequency	0,534146	0,98900	Sucesso
Runs	0,739918	0,99400	Sucesso
Longest-Run-of-Ones	0,215519	0,99100	Sucesso
Binary Matrix Rank	0,750116	0,99800	Sucesso
FFT	0,350485	0,98300	Sucesso
Non-Overlapping Template	0,528462	0,98800	Sucesso
Overlapping Template	0,213309	0,98200	Sucesso
Universal	0,122325	0,97900	Sucesso
Linear Complexity	0,684381	0,99700	Sucesso
Serial	0,136457	0,99600	Sucesso
Approximate Entropy	0,832484	0,99800	Sucesso
Random Excursions	0,469832	0,98300	Sucesso
Random Excursions Variant	0,658771	0,99200	Sucesso
Cumulative Sums	0,657894	0,99600	Sucesso

Tabela 3.8: *P-value e a razão de testes aprovados para cada teste do NIST, para o mapa o-tanh com $r = 3$, $R_x = 1/5$, com CFT.*

Teste Estatístico	P-value	Aprovados	Resultado
Frequency	0,925478	0,99700	Sucesso
Block Frequency	0,569871	0,98600	Sucesso
Runs	0,125478	0,98200	Sucesso
Longest-Run-of-Ones	0,289635	0,984100	Sucesso
Binary Matrix Rank	0,515896	0,99100	Sucesso
FFT	0,478956	0,98800	Sucesso
Non-Overlapping Template	0,157896	0,98700	Sucesso
Overlapping Template	0,158963	0,98200	Sucesso
Universal	0,089637	0,98000	Sucesso
Linear Complexity	0,735961	0,99400	Sucesso
Serial	0,358942	0,99100	Sucesso
Approximate Entropy	0,351789	0,99300	Sucesso
Random Excursions	0,75489	0,99000	Sucesso
Random Excursions Variant	0,656983	0,98900	Sucesso
Cumulative Sums	0,485236	0,98500	Sucesso

mapas e-tanh e o-tanh, respectivamente, com $q = 1$ e valores de r entre 3 e 5. Portanto, a informação gerada pela entropia condicional é importante para a determinação do valor do salto de amostras p.

3.4 Geração de PRNG com pós-processamento usando sequências-m

Nas seções anteriores, empregamos a técnica de saltos de amostras para eliminar a correlação entre as amostras caóticas. Com esta técnica o valor máximo alcançado por R_x é igual a 1 (com CVT). Nesta seção, não empregamos salto de amostras (portanto, adotamos nesta seção $p = 1$) e esta correlação é eliminada por um pós-processamento após codificação, empregando a sequência gerado por um LFSR. O foco desta seção é o projeto do comprimento do LFSR. Algumas propriedades destas sequências são descritas na próxima subseção.

O esquema para a geração de um PRNG considerado nesta seção tem três blocos, conforme é mostrado na Figura 3.8. Considerando uma codificação de q bits, cada variável X_n é mapeado em uma sequência código $Z_{(k-1)q+1}Z_{(k-1)q+2}...Z_{kq}$. Por exemplo, para $q = 4$, X_1 é mapeado em $\{Z_1Z_2Z_3Z_4\}$, X_2 é mapeado em $\{Z_5Z_6Z_7Z_8\}$ e assim sucessivamente. Cada sequência de símbolos na saída do codificador é mapeada na saída da unidade de pós-processamento com o mesmo

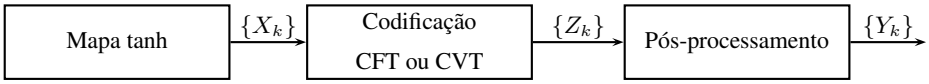


Figura 3.8: *Diagrama de blocos para geração de um PRNG.*

Tabela 3.9: *Exemplos de polinômios primitivos de grau até 50.*

(1, 0)	(11, 2, 0)	(21, 2, 0)	(31, 3, 0)	(41, 3, 0)
(2, 1, 0)	(12, 6, 4, 1, 0)	(22, 1, 0)	(32, 7, 6, 2, 0)	(42, 5, 4, 3, 2, 1, 0)
(3, 1, 0)	(13, 4, 3, 1, 0)	(23, 5, 0)	(33, 13, 0)	(43, 6, 4, 3, 0)
(4, 1, 0)	(14, 5, 3, 1, 0)	(24, 4, 3, 1, 0)	(34, 8, 4, 3, 0)	(44, 6, 5, 2, 0)
(5, 2, 0)	(15, 1, 0)	(25, 3, 0)	(35, 2, 0)	(45, 4, 3, 1, 0)
(6, 1, 0)	(16, 5, 3, 2, 0)	(26, 6, 2, 1, 0)	(36, 11, 0)	(46, 8, 5, 3, 2, 1, 0)
(7, 1, 0)	(17, 3, 0)	(27, 5, 2, 1, 0)	(37, 6, 4, 1, 0)	(47, 5, 0)
(8, 4, 3, 2, 0)	(18, 7, 0)	(28, 3, 0)	(38, 6, 5, 1, 0)	(48, 7, 5, 4, 2, 1, 0)
(9, 4, 0)	(19, 5, 2, 1, 0)	(29, 2, 0)	(39, 4, 0)	(49, 9, 0)
(10, 3, 0)	(20, 3, 0)	(30, 6, 4, 1, 0)	(40, 5, 4, 3, 0)	(50, 4, 3, 2, 0)
Notação: $(50, 4, 3, 2, 0) \Rightarrow p(x) = x^{50} + x^4 + x^3 + x^2 + 1$				

comprimento, portanto, esta unidade tem taxa unitária. A taxa deste sistema em *bits* por amostras caótica é $R_x = q$. A sequência $\{Y_k\}$ é a entrada da bateria de teste NIST.

3.4.1 Sequências-m

Uma sequência $\{W_k\}$ é gerada por um LFSR de comprimento N se satisfaz a seguinte relação:

$$W_k = c_1 W_{k-1} \oplus c_2 W_{k-2} \oplus ... \oplus c_{N-1} W_{k-(N-1)} \oplus W_{k-N} \tag{3.2}$$

em que $c_i \in \{0, 1\}$ e $\oplus$ denota soma modulo dois. A conexão de realimentação do LFSR pode ser representada por um polinômio de realimentação

$$p(x) = 1 + c_1 x + c_2 x^{N-2} + ... + c_{N-1} x^{N-1} + x^N. \tag{3.3}$$

O polinômio de realimentação define o período e o comportamento estatístico da sequência $\{W_k\}$. Para garantir o maior período possível $2^N - 1$ desta sequência, o polinômio $p(x)$ deve ser primitivo. A sequência gerada por um LFSR com um polinômio primitivo é chamada sequência de comprimento máximo, ou simplesmente de sequência-m. Os polinômios primitivos usados nesta dissertação são apresentados na Tabela 3.9.

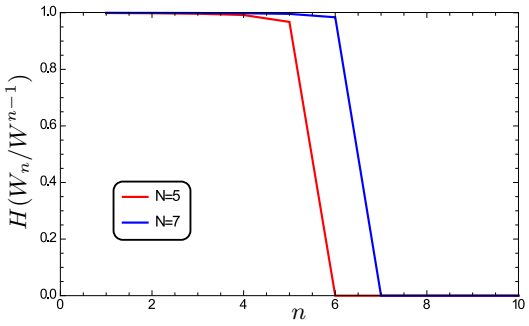


Figura 3.9: $H(W_n \mid W^{n-1})$ *versus* n *para sequências-m com* $N = 5$ *e* $N = 6$.

As sequências-m geradas por polinômios de realimentação de grau N apresentam uma entropia condicional $H(W_n/W^{n-1})$ aproximadamente igual a um para valores de n até N, enquanto que, para $n > N$, tende a zero devido a repetição dos valores das subsequência de comprimento maior que N, como ilustra a Figura 3.9. As subsequências de comprimento $n \leq N$ acontecem com probabilidade aproximadamente igual a $\frac{1}{2^n}$, apresentado uma estatística local quase ideal de ordem N. Esta aproximação melhora com o aumento do N e será considerada nas análises a seguir. Esta característica da sequência-m é utilizada para eliminar a memória existente nas sequências binárias obtidas da família de mapas tanh.

3.4.2 Esquema de geração de PRNG com pós-processamento

Para aproveitar as boas propriedades estatísticas das sequências-m propomos nesta subseção um método para gerar um PRNG que é uma combinação da técnica de discretização assosiada à codificação de amostras de um mapa caótico, seguido por um pós-processamento com uma sequência-m. Inicialmente, obtém-se uma sequência binária a partir da discretização de um mapa caótico (usando um dos métodos, CFT ou CVT) e realiza-se uma soma módulo 2 desta com uma sequência-m gerada por um LFSR. Este método é ilustrado na Figura 3.10. Seja $\{Z_k\}$ a sequência na saída de um codificador CFT ou CVT e $\{W_k\}$ a sequência-m gerada pelo LFSR, então a k-ésima amostra de saída $\{Y_k\}$ é dada por:

$$Y_k = Z_k \oplus W_k. \tag{3.4}$$

Suponha que a sequência $\{Z_k\}$ é formada com $q = 1$ (dois níveis de quantização) e é independente de $\{W_k\}$. Suponha ainda que estas sequências satisfazem as seguintes condições:

▷ A sequência $\{W_k\}$ tem estatística local ideal de ordem N, isto é, para $n \leq N$:

$$\Pr(w_1 ... w_n) = \Pr(W_1 = w_1, ..., W_n = w_n) = \frac{1}{2^n}. \tag{3.5}$$

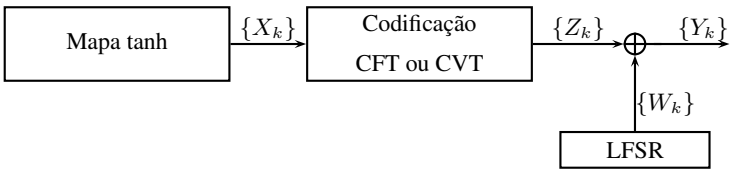


Figura 3.10: *Diagrama de blocos do esquema de geração de um PRNG usando sequências-m.*

▷ Cada variável aleatória Z_k tem distribuições uniforme, $\Pr(Z_k = 0) = \Pr(Z_k = 0) = \frac{1}{2}$, e duas variáveis aleatórias Z_k e Z_{k+n} são estatisticamente independentes para $n \geq N$, ou seja:

$$\Pr(z_k z_{k+m}) = \Pr(z_k) \cdot \Pr(z_{k+n}) = \frac{1}{4}. \tag{3.6}$$

Provaremos a seguir que se as condições (3.5) e (3.6) são satisfeitas, a sequência $\{Y_k\}$ é formada por variáveis aleatórias independentes e identicamente distribuídas com distribuição uniforme. Observe-se que a condição (3.6) implica que o valor de N deve ser escolhido de tal forma que seja maior ou igual a memória existente na sequência $\{Z_k\}$ que é devida à memória existente na sequência $\{X_k\}$.

Para $n \leq N$, a probabilidade conjunta de n variáveis aleatórias é dada por:

$$\begin{aligned}
\Pr(y_1 ... y_n) &= \sum_{w_1 ... w_n} \Pr(y_1 ... y_n | w_1 ... w_n) \cdot \Pr(w_1 ... w_n) && (3.7)\\
&= \sum_{w_1 ... w_n} \Pr(Z_1 = y_1 \oplus w_1, ..., Z_n = y_n \oplus w_n) \cdot \Pr(w_1 ... w_n) && (3.8)\\
&= \frac{1}{2^n} \sum_{w_1 ... w_n} \Pr(Z_1 = y_1 \oplus w_1, ..., Z_n = y_n \oplus w_n) && (3.9)\\
&= \frac{1}{2^n} && (3.10)
\end{aligned}$$

em que (3.8) segue da independência entre $\{Z_k\}$ e $\{W_k\}$ e (3.9) segue de (3.5). Considere agora que $n \geq N$:

$$\begin{aligned}
\Pr(y_1 y_{k+n}) &= \sum_{w_k w_{k+n}} \Pr(y_k y_{k+n} | w_k, w_{k+n}) \cdot \Pr(w_k w_{k+n}) && (3.11)\\
&= \sum_{w_k, w_{k+n}} \Pr(Z_k = y_k \oplus w_k, Z_{k+n} = y_{k+n} \oplus w_{k+n}) \cdot \Pr(w_k w_{k+n}) && (3.12)\\
&= \sum_{w_k, w_{k+n}} \Pr(Z_k = y_k \oplus w_k) \cdot \Pr(Z_{k+n} = y_{k+n} \oplus w_{k+n}) \cdot \Pr(w_k w_{k+n}) && (3.13)\\
&= (\frac{1}{2})^2 \sum_{w_k, w_{k+n}} \cdot \Pr(w_k w_{k+n}) && (3.14)
\end{aligned}$$

em que (3.13) e (3.14) segue de (3.6). Observa-se que a variável aleatória Y_k é independente de todas as variáveis em instantes passados $\{Y_\ell\}_{\ell=1}^{k-1}$, para todo k, visto que a independência para $\ell = k-1, ..., k-(N-1)$, segue de (3.10) e para $\ell < k - N$ segue de (3.14). Por tanto para $n > N$:

$$\begin{aligned}
\Pr(y_1 ... y_n) &= \Pr(y_n | y^{n-1}) \cdot \Pr(y_{n-1} | y^{n-2}) \cdots \Pr(y_{N+1} | y^N) \cdot \Pr(y^N) && (3.15)\\
&= \Pr(y_n) \cdot \Pr(y_{n-1}) \cdots \Pr(y_{N+1}) \cdot \frac{1}{2^N} && (3.16)\\
&= \frac{1}{2^{n-N}} \cdot \frac{1}{2^N} && (3.17)\\
&= \frac{1}{2^n} && (3.18)
\end{aligned}$$

o que em conjunto com (3.10) conclui a prova que $\{Y_n\}$ é uma sequência aleatória. Uma consequência das equações (3.5) e (3.6) é que:

$$H(Y_k | Y^{k-1}) = 1 \tag{3.19}$$

para todo k. As Figuras 3.11 e 3.12 apresentam a entropia condicional da sequência $\{Y_k\}$ para os mapas e-tanh e o-tanh, respectivamente, para $r = 3$, $q = 1$, codificação CFT, para LFSR de comprimentos $N = 4, 6, 7$ (e-tanh) e $N = 2, 4, 5$ (o-tanh). Observa-se nestas figuras que o valor de H é menor que 0,9999 para valores menores de n^* (igual a 7 para o mapa e-tanh e igual a 5 para o mapa o-tanh) o que é um indicador que a sequência binaria gerada com $N < n^*$ não é aleatória (vimos na seção anterior que a aprovação no teste NIST requer uma taxa de entropia de ordem de 0,9999). Assim é necessário um polinômio de grau N igual a n^* para gerar uma sequência binária aleatória com $R_x = 1$.

O aumento do nível de discretização ($q > 1$) conduz a uma taxa de *bits* $R_x = q$, mas introduz um aumento na correlação da sequência $\{Y_k\}$ devido à correlação existente nas sequências códigos que rotulam cada amostra caótica. Propomos, então, um valor estimado $\hat{N}$ do grau do polinômio N

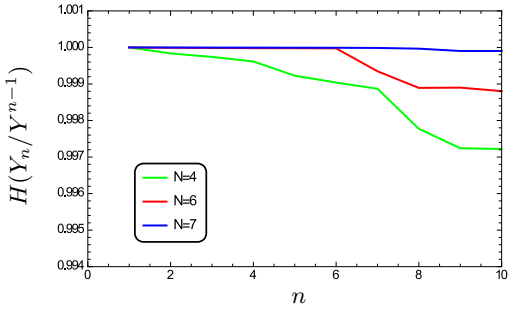


Figura 3.11: $H(Y_n \mid Y^{n-1})$ *versus* n *do mapa e-tanh com* $r = 3$, $q = 1$ *e CFT para* $N = 4, 6, 7$.

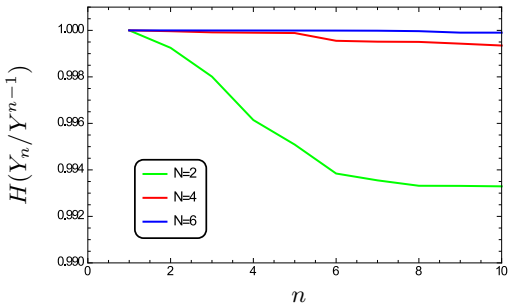


Figura 3.12: $H(Y_n \mid Y^{n-1})$ *versus* n *do mapa o-tanh com* $r = 3$, $q = 1$ *e CFT para* $N = 2, 4, 6$.

do LFSR para gerar uma sequência $\{Y_k\}$ com boas propriedades de aleatoriedade da seguinte forma:

$$\hat{N} = n^* \cdot q. \tag{3.20}$$

Seja N_{min} o menor valor do grau do polinômio N para que a sequência $\{Y_k\}$ passe no teste NIST. A Tabela 3.10 compara os valores de $\hat{N}$ e N_{min} para diferentes taxas $R_x = q$ para os mapas e-tanh e o-tanh com $r = 3$ e para codificação CFT (O caso CVT será tratado no próximo capítulo). Observa-se nesta tabela que N_{min} cresce aproximadamente linearmente com q sendo bem aproximado por $\hat{N}$ (uma discrepância entre N_{min} e $\hat{N}$ de no máximo igual a 3 é observada nesta tabela).

Tabela 3.10: *Comparação entre* N_{min} *e* $\hat{N}$ *para o mapa e-tanh* ($n^* = 7$) *e o-tanh* ($n^* = 5$) *com* $r = 3$ *e codificação CFT.*

	Mapa e-tanh		Mapa o-tanh	
$R_x = q$	N_{min}	$\hat{N}$	N_{min}	$\hat{N}$
1	7	7	5	5
2	13	14	10	10
3	18	21	12	15
4	26	28	17	20
5	34	35	22	25
6	40	42	27	30
7	47	49	33	35

Pode-se ver que com o aumento do valor de N pode-se gerar taxas maiores, que no entanto, os requerimentos de memória do LFSR cresce linearmente com a taxa de *bits*. Uma unidade de pós-processamento com menor requerimento de memória será proposta no próximo capítulo.

CAPÍTULO 4

NOVA UNIDADE DE PÓS-PROCESSAMENTO COM CVT

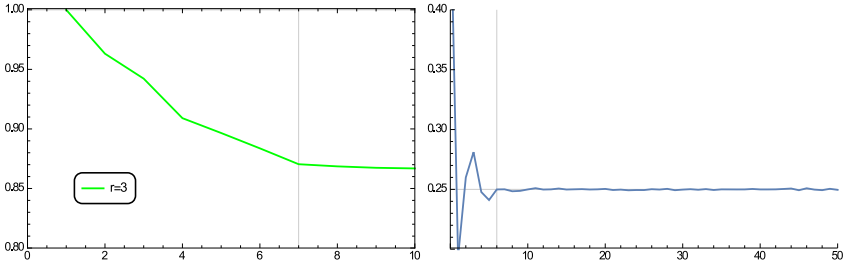
ESTE capítulo emprega a função autocorrelação para quantificar a memória de uma sequência discreta e analisar seu comportamento para a família de mapas tanh usando as duas codificações CFT e CVT. A partir do comportamento desta função para codificação CVT, propõe-se um novo bloco de pós-processamento com menor requerimento de memória em relação os obtidos no capítulo anterior.

4.1 Função autocorrelação

A entropia condicional pode ser interpretada como um quantificador que reflete algumas propriedades estatísticas de uma sequência discreta. A função autocorrelação também pode servir para determinar a memória da sequência $\{Z_k\}$ e consequentemente o dimensionamento do bloco de pós-processamento. A função autocorrelação de um processo estacionário $\{X_k\}$ é definida por:

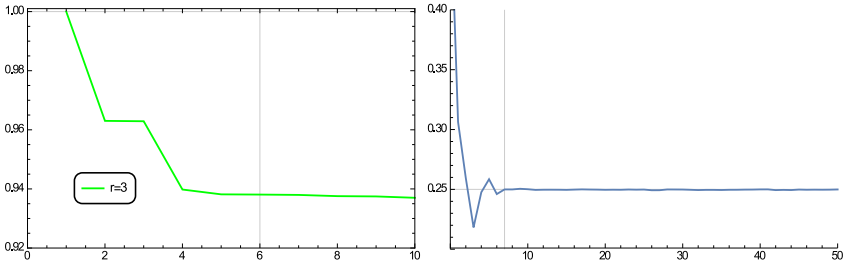
$$R_x[m] = E[X_k X_{k+m}] \tag{4.1}$$

em que $E[X_k]$ é valor esperado da variável aleatória X_k. As Figuras 4.1 e 4.2 apresentam os gráficos de entropia condicional das sequências codificadas $\{Z_k\}$ do sistema apresentado na Seção 3.2, para o mapa e-tanh com $r = 3$, $q = 1$, com codificação CFT e CVT, respectivamente, e a curva da função autocorrelação destas sequências com os mesmos parâmetros. Análises semelhantes para o mapa o-tanh são mostradas nas Figuras 4.3 e 4.4.



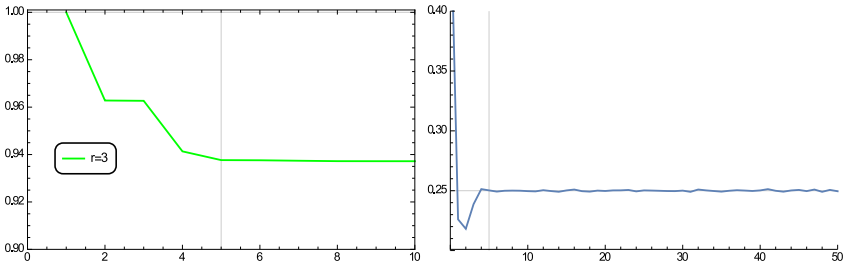
(a) Entropia condicional da sequência $\{Z_k\}$ (b) Função autocorrelação da sequência $\{Z_k\}$

Figura 4.1: *Quantificadores do mapa e-tanh com* $r = 3$, $q = 1$, $n^* = 7$ *e CFT.*



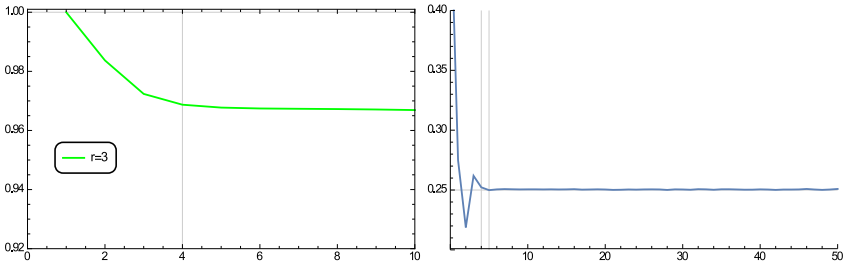
(a) Entropia condicional da sequência $\{Z_k\}$ (b) Função autocorrelação da sequência $\{Z_k\}$

Figura 4.2: *Quantificadores do mapa e-tanh com* $r = 3$, $q = 1$, $n^* = 6$ *e CVT.*



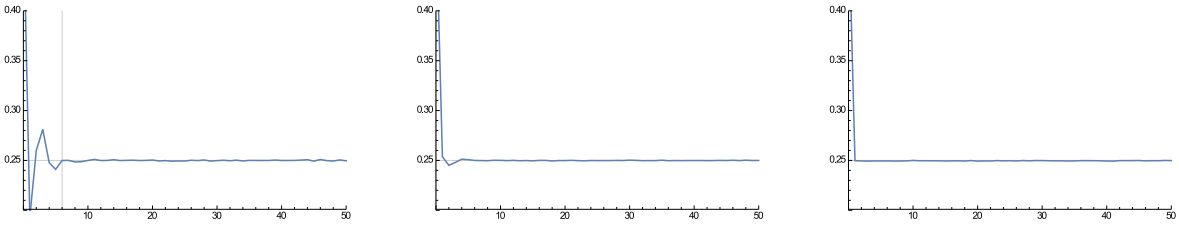
(a) Entropia condicional da sequência $\{Z_k\}$ (b) Função autocorrelação da sequência $\{Z_k\}$

Figura 4.3: *Quantificadores do mapa o-tanh com* $r = 3$, $q = 1$, $n^* = 5$ *e CFT.*



(a) Entropia condicional da sequência $\{Z_k\}$ (b) Função autocorrelação da sequência $\{Z_k\}$

Figura 4.4: *Quantificadores do mapa o-tanh com* $r = 3$, $q = 1$, $n^* = 5$ *e CVT.*

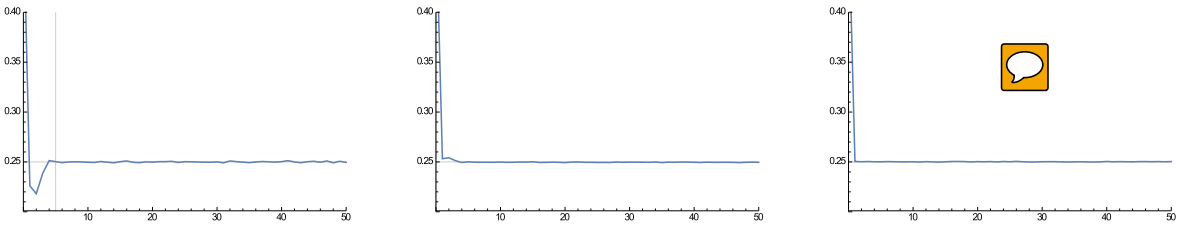


(a) Função autocorrelação da sequência $\{Z_k\}$

(b) Função autocorrelação da sequência $\{Y_k\}$ com $N = 4$

(c) Função autocorrelação da sequência $\{Y_k\}$ com $N = 7$

Figura 4.5: *Função autocorrelação para o mapa e-tanh com $r = 3$, $q = 1$, $n^* = 7$, para $N = 4$ e $N = 7$ com codificação CFT.*



(a) Função autocorrelação da sequência $\{Z_k\}$

(b) Função autocorrelação da sequência $\{Y_k\}$ com $N = 3$

(c) Função autocorrelação da sequência $\{Y_k\}$ com $N = 5$

Figura 4.6: *Função autocorrelação para o mapa o-tanh com $r = 3$, $q = 1$, $n^* = 5$, para $N = 3$ e $N = 5$ com codificação CFT.*

Observa-se que os gráficos da função autocorrelação para cada codificação tem um comportamento semelhante e o valor de n^* estabelecido pela entropia condicional está em consonância com o indicado pela função autocorrelação.

No processo descrito na Seção 3.4.2 foi definido um valor estimado para o comprimento do LFSR ($\hat{N}$) para que a sequência $\{Y_k\}$ tenha propriedades de uma sequência aleatória, dependendo dos parâmetros n^* e q. Também observa-se que o valor de $\hat{N}$ aumenta linearmente com o valor de q para codificação CFT. As Figuras 4.5 e 4.6 apresentam o comportamento da função autocorrelação dos processos $\{Z_k\}$ e $\{Y_k\}$, com a utilização de dois valores de N ($N = 4$ e $N = 7$). Observa-se que para $N = n^*$ a função autocorrelação da sequência de saída $\{Y_k\}$ tem comportamento de um impulso que é típico de uma sequência aleatória.

4.1.1 Comparação da função autocorrelação das codificações CFT e CVT

A utilização da codificação CFT, como foi mostrado no capítulo anterior, o valor de $\hat{N}$ cresce linearmente com o aumento de q. As Figuras 4.7 e 4.8 mostram a função de autocorrelação dos mapas e-tanh e o-tanh, respectivamente, com $r = 3$, $q = 3, 5, 7$, com codificação CFT. Observa-se

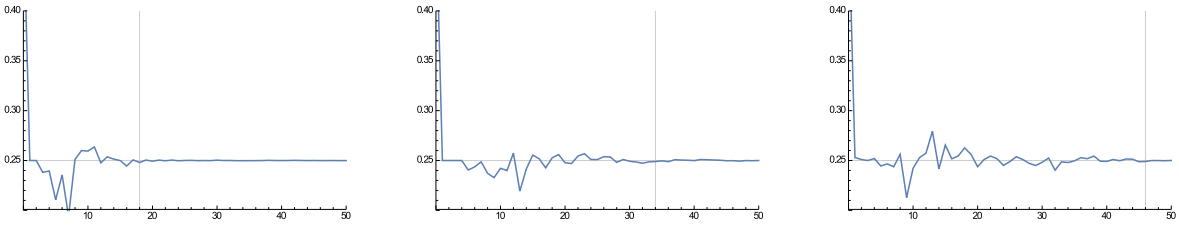
um espalhamento linearmente dependente de q da autocorrelação existente na sequência binária. As Figuras 4.9 e 4.10 mostram a função autocorrelação da sequência $\{Z_k\}$ utilizando codificação CVT, para os mapas e-tanh e o-tanh, respectivamente, com $r = 3$, $q = 3, 5, 7$. Diferentemente do caso CFT, estas figuras mostram que $R_Z[m]$ possui um padrão de concentração para cada valor de q. Existem picos desta função em valores específicos de m dado por:

$$m_p = kq \tag{4.2}$$

sendo m_p o valor de m em que ocorre um pico, para $k = 1, 2, 3....$ Este comportamento indica que existe correlação entre *bits* de sequência códigos distintas separadas de q posições. Por exemplo, três amostras caóticas consecutivas $X_1X_2X_3$ geram três sequências códigos $(Z_1Z_2Z_3)$, $(Z_4Z_5Z_6)$ e $(Z_7Z_8Z_9)$. Portanto existe nestas sequências código correlação apenas entre os *bits* de cada um dos três conjuntos $\{Z_1, Z_4, Z_7\}$, $\{Z_2, Z_5, Z_8\}$, e $\{Z_3, Z_6, Z_9\}$. O comportamento de $R_z[m]$ indica que não existe correlação entre os *bits* de uma mesma sequência código, ou em *bits* de sequências códigos distintas que estão em posições distintas.

Este comportamento implica em uma diferença entre as duas codificações, CFT e CVT, em termos de memória do pós-processamento. A codificação CVT apresenta intervalos de *bits* na sequência $\{Z_k\}$ que são descorrelacionados, precisando de uma quantidade menor de *bits* que apresentam estatística local quase-ideal na sequência-m para eliminar a correlação de $\{Z_k\}$.

A magnitude dos picos de $R_z[m]$ nos intervalos $m_p = kq$ decresce com k e pode-se definir um número finito de picos, denotado por P. Observa-se na Figura 4.9 e 4.10 que o valor de P é praticamente constante com q para os mapas e-tanh e o-tanh. Considera-se neste trabalho $P = 6$ e propõe-se na próxima seção um bloco de pós-processamento que faz uso desta característica da função de autocorrelação.

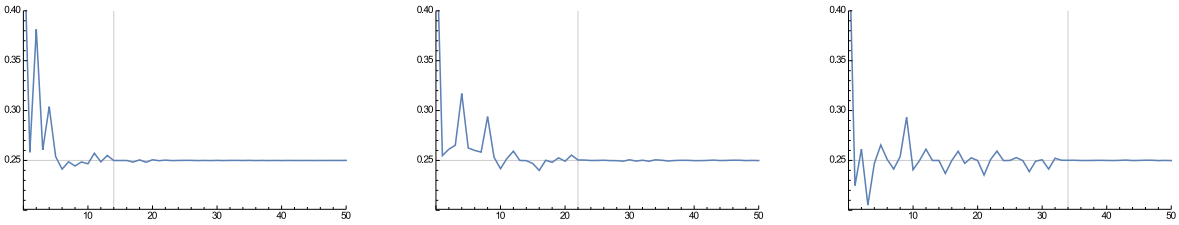


(a) Função autocorrelação da sequência $\{Z_k\}$ para $q = 3$

(b) Função autocorrelação da sequência $\{Z_k\}$ para $q = 5$

(c) Função autocorrelação da sequência $\{Z_k\}$ para $q = 7$

Figura 4.7: *Função autocorrelação para o mapa e-tanh com $r = 3$, $q = 3$, $q = 5$ e $q = 7$ com codificação CFT.*

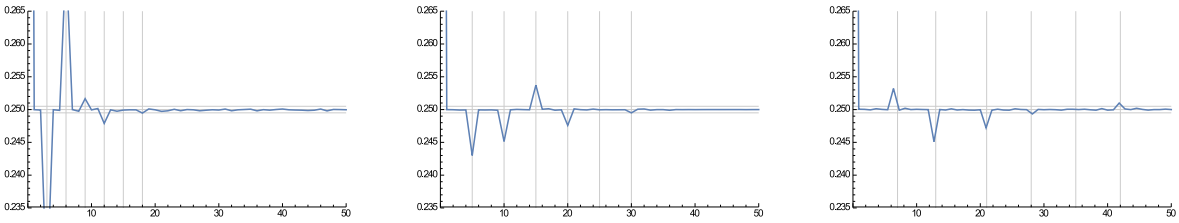


(a) Função autocorrelação da sequência $\{Z_k\}$ para $q = 3$

(b) Função autocorrelação da sequência $\{Z_k\}$ para $q = 5$

(c) Função autocorrelação da sequência $\{Z_k\}$ para $q = 7$

Figura 4.8: *Função autocorrelação para o mapa o-tanh com $r = 3$, $q = 3$, $q = 5$ e $q = 7$ com codificação CFT.*

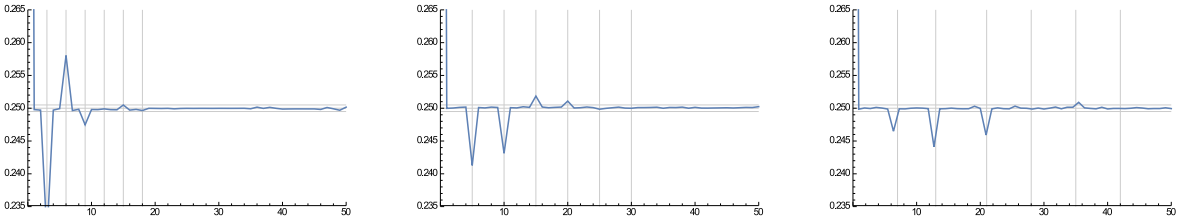


(a) Função autocorrelação da sequência $\{Z_k\}$ para $q = 3$

(b) Função autocorrelação da sequência $\{Z_k\}$ para $q = 5$

(c) Função autocorrelação da sequência $\{Z_k\}$ para $q = 7$

Figura 4.9: *Função autocorrelação para o mapa e-tanh com $r = 3$, $q = 3$, $q = 5$ e $q = 7$ com codificação CVT.*



(a) Função autocorrelação da sequência $\{Z_k\}$ para $q = 3$

(b) Função autocorrelação da sequência $\{Z_k\}$ para $q = 5$

(c) Função autocorrelação da sequência $\{Z_k\}$ para $q = 7$

Figura 4.10: *Função autocorrelação para o mapa o-tanh com $r = 3$, $q = 3$, $q = 5$ e $q = 7$ com codificação CVT.*

4.2 Novo método de pós-processamento utilizando codificação CVT

Para eliminar a correlação existente na sequência $\{Z_k\}$ gerada pela codificação CVT, utiliza-se a característica apresentada nas Figuras 4.9 e 4.10, que é a presença de P picos. A Figura 4.11 mostra a implementação do novo bloco de pós-processamento. A ideia central consiste em projetar um LFSR que quebre a correlação entre amostras (Z_k e Z_j) em posições relativas indicados pelos P picos, isto é, separadas por kq amostras, $k = 1, 2..., P$.

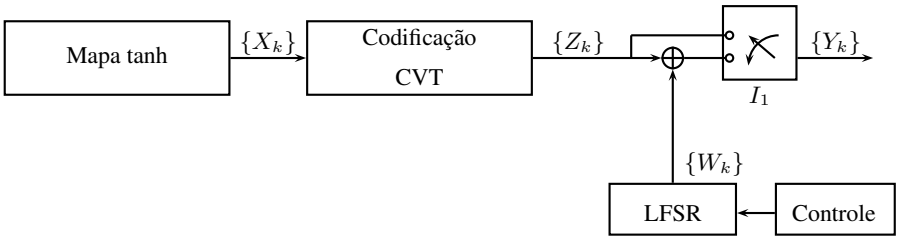


Figura 4.11: *Diagrama de blocos de um esquema da geração de um PRNG usando sequências-m.*

Mostra-se no digrama de blocos da Figura 4.11 uma chave I_1, quando esta está aberta os *bits* do processo de codificação são iguais aos *bits* da saída do sistema. Quando a chave está fechada os *bits* da saída $\{Y_k\}$ são obtidos da soma modulo-2 das sequências $\{Z_k\}$ e $\{W_k\}$. Os *bits* da sequência-m são utilizados de tal forma que para os *bits* correlacionados da sequência $\{Z_k\}$ utiliza-se *bits* descorrelacionados da sequência $\{W_k\}$ que apresentam estatística local quase-ideal entre eles, garantindo que a sequência $\{Y_k\}$ seja descorrelacionada.

A operação das chaves depende dos valores de q e P como é exemplificado na Figura 4.12 para os parâmetros $q = 3$ e $P = 6$. Define-se uma janela de comprimento, $J = q(P + 1)$ *bits* para garantir que todos os picos fiquem no comprimento da janela. Os primeiros q *bits* são os mesmos do processo de quantização do mapa caótico, $Y_1 = Z_1$, $Y_2 = Z_2$ e $Y_3 = Z_3$, os demais *bits* da janela são obtidos da soma módulo-2 entre a sequência $\{Z_k\}$ e a sequência-m como se mostra na Figura 4.12. Na próxima janela faz-se o mesmo procedimento, os primeiros q *bits* da janela são obtidos diretamente do codificador CVT, $Y_{22} = Z_{22}$, $Y_{23} = Z_{23}$ e $Y_{24} = Z_{24}$, os demais *bits* novamente somam-se modulo-2 com a sequência-m. Este processo é controlado mediante a chave I_1. Os *bits* da sequência-m em janelas adjacentes são deslocados em um *bit*, o que é realizado pelo bloco de controle. Por exemplo, a primeira janela começa com W_1 até W_{18}, na segunda utiliza-se de W_2 até W_{19} e a terceira começa de novo com W_1 até W_{18}. O processo de deslocamento é realizado pelo bloco de controle, fazendo que janelas adjacentes comecem alternadamente com W_1 ou W_2.

Uma estimativa do grau do polinômio gerador que gera uma sequência $\{Y_k\}$ com propriedades de uma sequência descorrelacionada, denominado $\hat{N}'$, deve cumprir dois requerimentos, tal que:

- O grau do polinômio deve ser maior ou igual ao número de picos existentes da função autocorrelação da sequência $\{Z_k\}$.

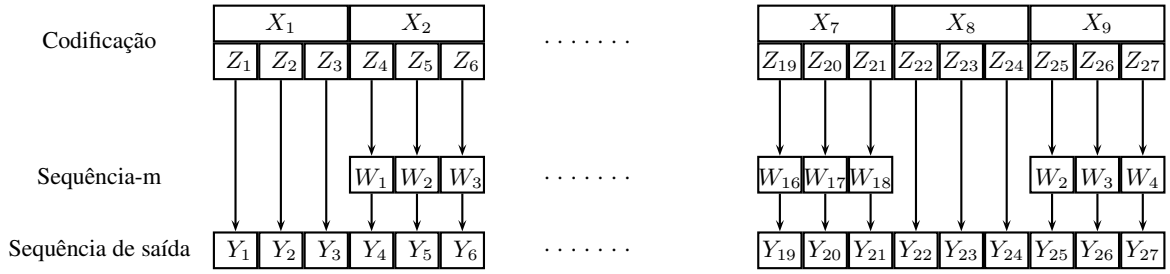


Figura 4.12: *Diagrama de blocos com pós-processamento para geração de um PRNG para* $q = 3$ *e* $P = 6$.

▷ O período da sequência-m deve ser maior que o comprimento da janela.

isto é, que $\hat{N}' \geq P$, o número de *bits* com estatística local quase-ideal na sequência-m seja maior ou igual a quantidade de bits correlacionados na janela, e que $2^{\hat{N}'} - 1 \geq q(P+1)$, garantindo que em uma janela não existam *bits* correlacionados da sequência-m. Uma propriedade relevante da sequência-m é que sua decimação pode formar uma nova sequência-m, conforme é descrito no próximo teorema [36].

Teorema 1[36]: Seja $\{W_k\}$ uma sequência-m gerada por um polinômio primitivo de grau N. Uma sequência de decimação $\{W'_k\}$ de parâmetro s obtida a partir de $\{W_k\}$ é da forma $W'_k = W_{sk}$. Esta sequência de decimação também é uma sequência-m se, e somente se, $\gcd(s, 2^N - 1) = 1$.

Observe que os bits de $\{Z_k\}$ correlacionados na primeira janela da Figura 4.12 são, por exemplo, $Z_1, Z_4, Z_7, Z_{10}, \cdots$. Estes são somados com uma decimação de $\{W_k\}$ de parâmetro q para formar a sequência de saída $Z_1, Z_4 \oplus W_1, Z_7 \oplus W_4, Z_{10} \oplus W_7, \cdots$. De acordo com o Teorema 1 a sequência dezimada será uma sequência-m se, e somente se, q e 2^{N-1} sejam primos entre si. Para a família de mapas tanh com $P = 6$, um valor estimado do cumprimento do LFSR $\hat{N}'$ que cumpre os requerimentos discutidos anteriormente é $\hat{N}' = 6$, exceto para $q = 3, 6, 9, \cdots$, visto que para estes valores de q a decimação não gera uma sequência-m (de acordo com o Teorema 1). Nestes casos, devemos usar $\hat{N}' = 7$.

Denota-se por N'_{min} o menor valor do grau do polinômio gerador da sequência $\{W_k\}$ da Figura 4.12 no qual a sequência $\{Y_k\}$ passa no teste NIST. A Tabela 4.1 apresenta uma comparação dos valores de N_{min} (definido na Seção 3.4) e N'_{min} para a família de mapas tanh com $r = 3$, e diferentes valores de q, com codificação CVT. Fazendo uma comparação entre os valores de N_{min} para as duas codificações, pode-se observar que o aumento de q, não leva a um aumento N_{min}, ficando quase constante. A utilização do novo bloco de pós-processamento leva a uma diminuição de apro-

Tabela 4.1: *Comparação entre N_{min} e N'_{min} para o mapa e-tanh com $r = 3$, com codificação CVT.*

	Mapa e-tanh		Mapa o-tanh	
Taxa de Bits	N_{min}	N'_{min}	N_{min}	N'_{min}
$q = 3$	9	7	11	7
$q = 4$	10	6	10	6
$q = 5$	10	6	9	6
$q = 6$	11	7	10	7
$q = 7$	10	6	10	6

ximadamente 30% da memória do LFSR, para diferentes valores de q, entretanto a complexidade é maior com a introdução da chave I_1 e do bloco de controle. O novo bloco de pós-processamento utiliza os requerimentos antes expostos, $(\hat{N}' \geq P)$ e $(2^{\hat{N}'} - 1 \geq q(P+1))$, assim como que o deslocamento realizado na sequência-m, faz que cada janela soma-se modulo-2 com sequências de decimação diferentes, o que é o mesmo, sequências-m diferentes (geradas por polinômios primitivos diferentes). Os valores de N'_{min} indicados nesta tabela coincidem com o valor estimado de $\hat{N}'$ para esta família de mapas.

As Figuras 4.13 e 4.14 ilustra a função autocorrelação das sequências $\{Z_k\}$ (antes do pós-processamento) e $\{Y_k\}$ (depois do pós-processamento) do mapa e-tanh com $r = 3$, $q = 3$ e $q = 5$, respectivamente, com codificação CVT. Um analise semelhante faz-se para o mapa o-tanh nas Figuras 4.15 e 4.16. A função autocorrelação de $\{Y_k\}$ tem um comportamento de um impulso, o que é esperado para uma sequência descorrelacionada.

Um comportamento semelhante apresenta o quantificador da entropia condicional. As Figuras 4.17 e 4.18 ilustram os gráficos de entropia condicional das sequências $\{Z_k\}$ e $\{Y_k\}$ do processo de pós-processamento proposto. A Figura 4.17 mostra a entropia condicional para o mapa e-tanh com

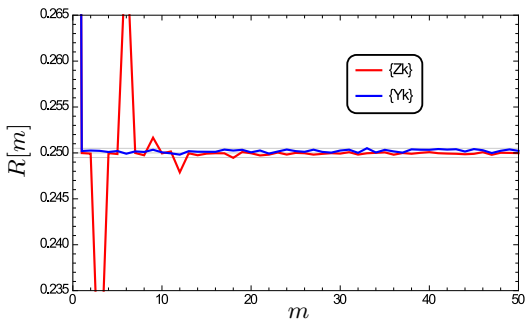


Figura 4.13: *$R[m]$ versus m para mapa e-tanh com $q = 3$ e $N'_{min} = 7$.*

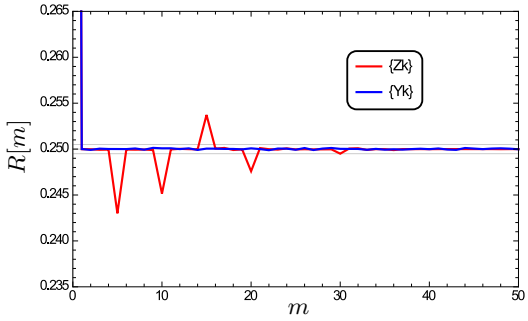


Figura 4.14: *$R[m]$ versus m para mapa e-tanh com $q = 5$ e $N'_{min} = 6$.*

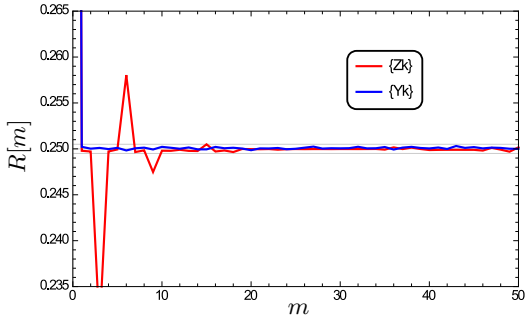


Figura 4.15: *$R[m]$ versus m para mapa o-tanh com $q = 3$ e $N'_{min} = 7$.*

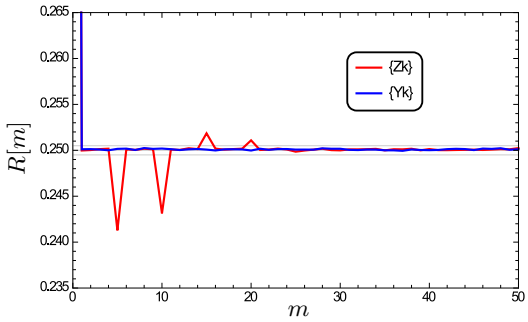


Figura 4.16: *$R[m]$ versus m para mapa o-tanh com $q = 5$ e $N'_{min} = 6$.*

$r = 3$, $q = 3$, e a Figura 4.18 mesmo quantificador e mesmos parâmetros para o mapa o-tanh.

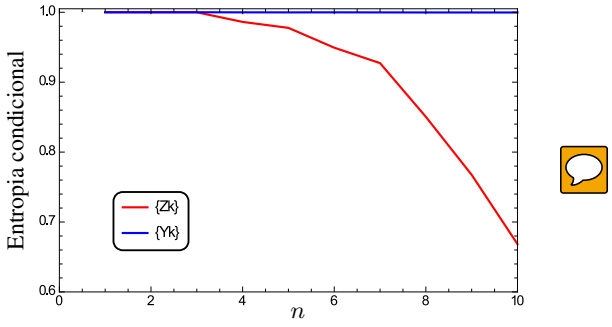


Figura 4.17: *Entropia condicional de* $\{Z_k\}$ *e* $\{Y_k\}$ *versus* n *para mapa e-tanh com* $r = 3$ *e* $q = 3$.

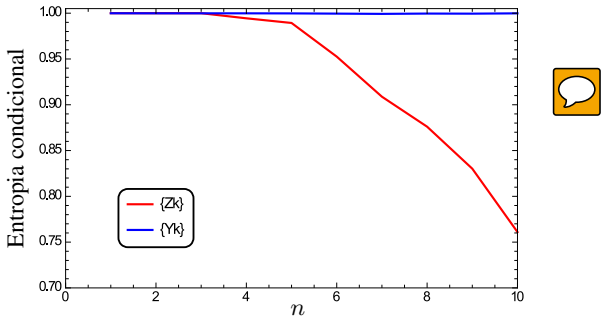


Figura 4.18: *Entropia condicional de* $\{Z_k\}$ *e* $\{Y_k\}$ *versus* n *para mapa o-tanh com* $r = 3$ *e* $q = 3$.

CAPÍTULO 5

IMPLEMENTAÇÃO DA METODOLOGIA PROPOSTA PARA OUTROS MAPAS CAÓTICOS

NESTE capítulo utiliza-se as técnicas mostradas nos capítulos anteriores para a obtenção de sequência pseudo-aleatórias para diferentes mapas caóticos utilizados na literatura. Verifica-se a diferença das duas codificações utilizadas nesta dissertação, como também a característica apresentada pela codificação CVT. Implementa-se o bloco de pós-processamento mostrado no capítulo anterior para diferentes mapas.

5.1 Mapas caóticos

Na literatura existe uma grande variedade de mapas caóticos utilizados em diferentes áreas da ciência. Nesta seção, apresenta-se dois mapas caóticos utilizados na área de telecomunicações. O mapa Cúbico (MC) é definido como [37]:

$$x_{k+1} = 4x_k^3 - 3x_k, \tag{5.1}$$

e o mapa de Hénon (MH) é da forma como [37]:

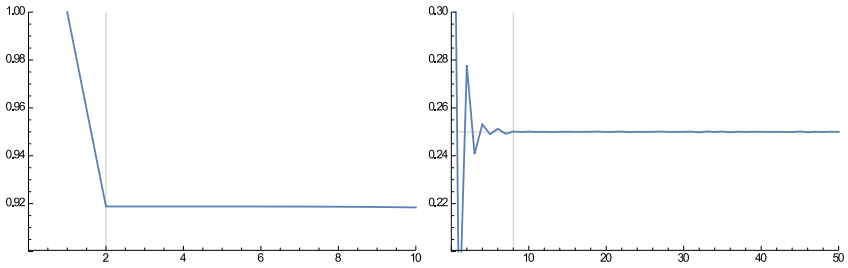
$$x_{k+1} = 1 + 0,3x_{k-1} - 1,4x_k^2. \tag{5.2}$$

Como foi apresentado em capítulos anteriores, esta dissertação utiliza dois quantificadores para mensurar o grau de aleatoriedade da sequência codificada (entropia condicional e função autocorrelação). As Figuras 5.1 e 5.2 apresentam estes quantificadores para $q = 1$ e codificação CFT para os dois mapas.

Denomina-se de n' o mínimo valor de m para o qual $R_Z[m]$ se estabiliza em $1/4$ (amostras com separação maior que n' são descorrelacionadas). As Figuras 5.1 e 5.2 mostram que o valor de n^* e n' não estão em consonância, como é obtido para a família de mapas tanh. Sempre que aconteça essa diferença entre estes valores, toma-se o maior valor para o projeto da sequência-m. As Figuras 5.3 e 5.4 mostram o espalhamento da função autocorrelação com o aumento de q quando utiliza-se a codificação CFT. A Tabela 5.1 mostra o valor de N_{min} definido na Seção 3.4 (menor comprimento do LFSR que passa no NIST) para os dois mapas com codificação CFT. A utilização desta codificação leva a um aumento linear do valor de N_{min} como mostra-se na Figura 5.5.

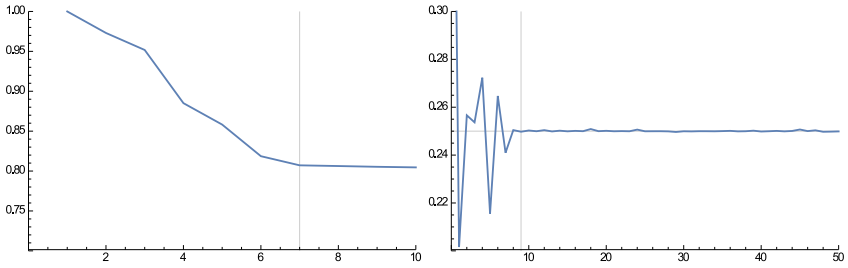
Pode-se observar que o valor de n' observado nas Figuras 5.3 e 5.4 é aproximadamente o valor necessário de N_{min}, como mostra a Tabela 5.1. Exemplo, para o mapa MC com $q = 3$ o valor de $n' = N_{min} = 14$, e para o mapa MH com $q = 5$ o valor de $n' = N_{min} = 36$.

A implementação da sequência-m do esquema da Figura 3.10 com valores de $N < \max\{n^*, n'\}$



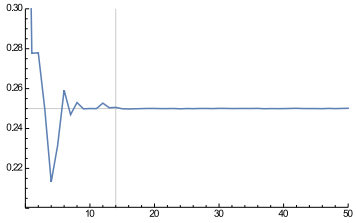
(a) Entropia condicional da sequência $\{Z_k\}$ (b) Função autocorrelação da sequência $\{Z_k\}$

Figura 5.1: *Quantificadores para o mapa MC com $q = 1$ e CFT.*

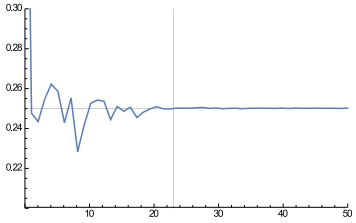


(a) Entropia condicional da sequência $\{Z_k\}$ (b) Função autocorrelação da sequência $\{Z_k\}$

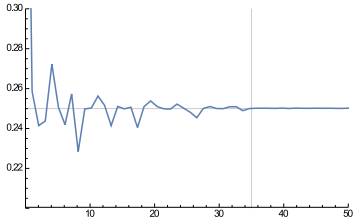
Figura 5.2: *Quantificadores para o mapa MH com $q = 1$ e CFT.*



(a) Função autocorrelação da sequência $\{Z_k\}$ para $q = 3$

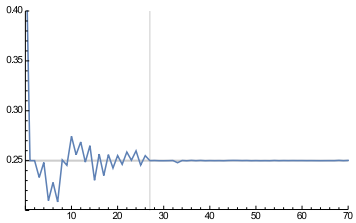


(b) Função autocorrelação da sequência $\{Z_k\}$ para $q = 5$

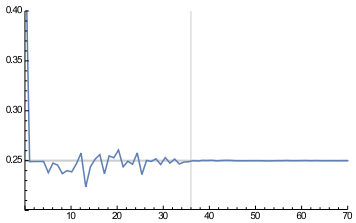


(c) Função autocorrelação da sequência $\{Z_k\}$ para $q = 7$

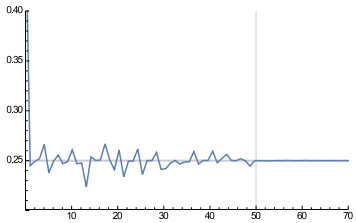
Figura 5.3: *Função autocorrelação do mapa MC para $q = 3$, $q = 5$ e $q = 7$ com codificação CFT.*



(a) Função autocorrelação da sequência $\{Z_k\}$ para $q = 3$



(b) Função autocorrelação da sequência $\{Z_k\}$ para $q = 5$



(c) Função autocorrelação da sequência $\{Z_k\}$ para $q = 7$

Figura 5.4: *Função autocorrelação do mapa MH com $q = 3$, $q = 5$, e $q = 7$ com codificação CFT.*

Tabela 5.1: *N_{min} para os mapas MC e MH com codificação CFT*

q	Mapa Cúbico N_{min}	Mapa Hénon N_{min}
1	8	9
2	12	18
3	14	27
4	18	33
5	23	36
6	28	44
7	35	50

não elimina as características não aleatórias das sequências caóticas $\{Z_k\}$. A Figura 5.6 apresenta o comportamento da entropia condicional para valores de $N = 6$ e $N = 8$ para o mapa MC, observando uma taxa de entropia constante próximo a 1 para $N = n' = 8$. Este resultado também é observado na função autocorrelação, como é ilustrado na Figura 5.7.

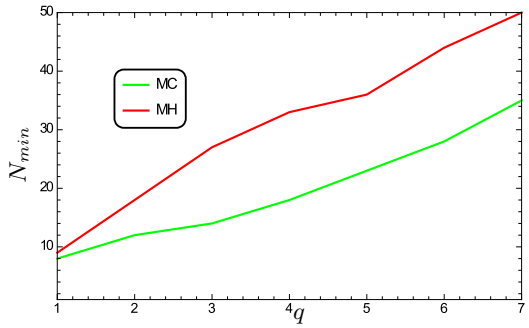


Figura 5.5: *N_{min} versus q para os mapas MC e MH com codificação CFT.*

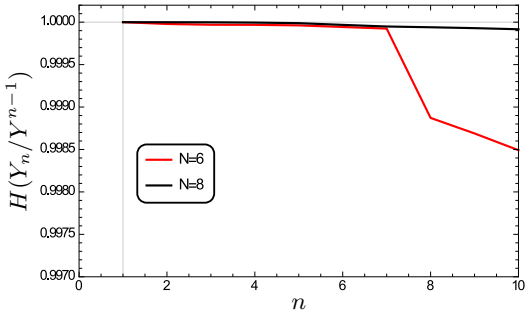


Figura 5.6: *$H(Y_n \mid Y^{n-1})$ versus n para mapa MC com $q = 1$ e codificação CFT para $N = 6, 8$.*

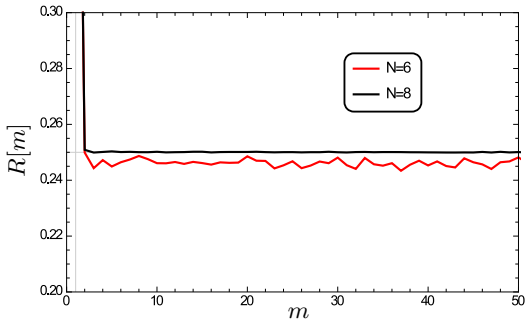
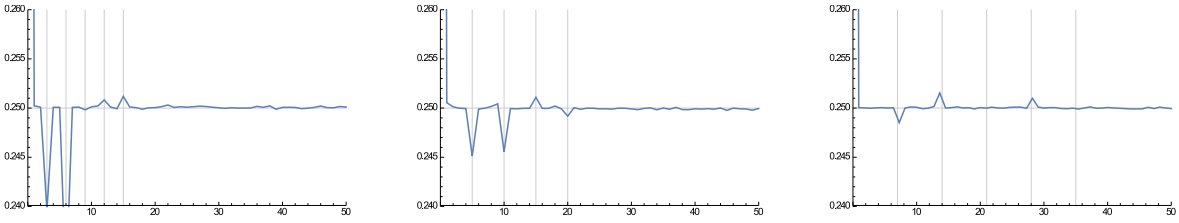


Figura 5.7: *$R[m]$ versus m da sequência $\{Y_k\}$ para o mapa MC com codificação CFT e $N = 6, 8$.*

5.2 Codificação CVT para os novos mapas

A utilização da codificação CVT leva a uma diminuição da memória do sistema, para a família de mapas tanh, devido à concentração dos picos da função autocorrelação em alguns valores de m. As Figuras 5.8 e 5.9 mostram a função autocorrelação dos mapas MC e MH, respectivamente, para diferentes valores do comprimento da palavra código ($q = 3, 5, 7$). É observado nestas que a função autocorrelação apresenta o espalhamento específico igual ao mostrado pela família de mapas tanh. Os picos da função autocorrelação ocorrem em $m_p = kp$. Cada mapa apresenta uma quantidade

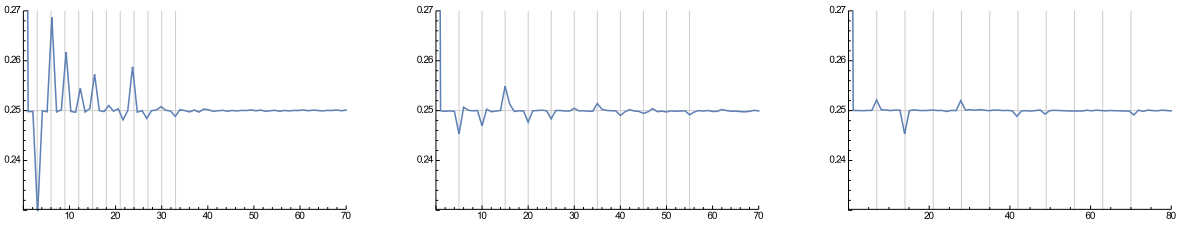


(a) Função autocorrelação da sequência $\{Z_k\}$ com $q = 3$

(b) Função autocorrelação da sequência $\{Z_k\}$ com $q = 5$

(c) Função autocorrelação da sequência $\{Z_k\}$ com $q = 7$

Figura 5.8: *Função autocorrelação para o mapa MC com* $q = 3, 5, 7$ *e codificação CVT.*



(a) Função autocorrelação da sequência $\{Z_k\}$ com $q = 3$

(b) Função autocorrelação da sequência $\{Z_k\}$ com $q = 5$

(c) Função autocorrelação da sequência $\{Z_k\}$ com $q = 7$

Figura 5.9: *Função autocorrelação para o mapa MH com* $q = 3, 5, 7$ *e codificação CVT.*

picos que permanece praticamente constante com q. A maior quantidades de picos que apresenta o mapa MC, denotado por P_C, é 6. Para o mapa MH este valor é $P_H = 11$.

A implementação do bloco de pós-processamento mostrado na Figura 4.11 com os parâmetros adequados, leva a uma estimativa $\hat{N}'$ do tamanho do registrador do LFSR. Para o mapa MC o valor de $\hat{N}'$ é 6. Este valor satisfaz todas as condições mostradas no capítulo anterior, que são $\hat{N}' \geq P$, $2^{\hat{N}'} - 1 \geq q(P + 1)$, e $\gcd(q, 2^{\hat{N}'} - 1) = 1$. Uma análise semelhante é realizada para o mapa MH, obtendo um valor de $\hat{N}' = 11$.

Na Tabela 5.2 apresenta uma comparação dos valores de N_{min} e N'_{min} para os mapas MC e MH, com diferentes valores de q. Pode-se observar que o aumento de q não leva a um aumento do valor de N'_{min}, ficando constante. Comparando-se N_{min} e N'_{min} a utilização deste bloco de pós-processamento leva a uma diminuição do grau do polinômio gerador da sequência-m de 36% aproximadamente para o mapa MC, e 27% para o mapa MH.

Como foi mostrado em capítulos anteriores, após a implementação deste pós-processamento, a sequência de saída apresenta características de uma sequência aleatória. As Figuras 5.10 e 5.11 mostram a função autocorrelação das sequências $\{Z_k\}$ e $\{Y_k\}$ para os mapas MC e MH, para $q = 3$. Comportamento semelhante é observado na entropia condicional das sequências $\{Z_k\}$ e $\{Y_k\}$, para

Tabela 5.2: *Comparação entre N_{min} e N'_{min} para os MC e MH, com codificação CVT.*

q	Mapa Cúbico		Mapa Hénon	
	N_{min}	N'_{min}	N_{min}	N'_{min}
1	8	–	8	–
2	11	–	15	–
3	12	7	15	11
4	12	7	15	11
5	12	7	15	11
6	12	7	15	11
7	12	7	15	11

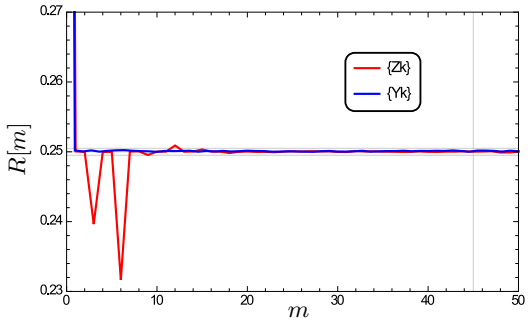


Figura 5.10: *$R[m]$ versus m para o mapa MC com $q = 3$ e $N'_{min} = 7$.*

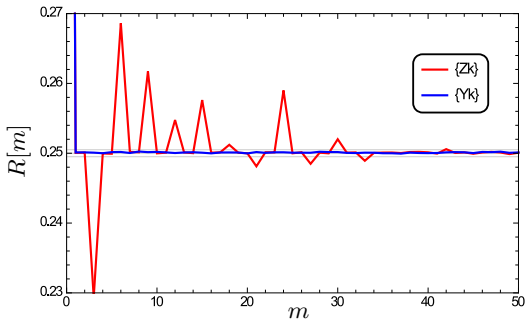


Figura 5.11: *$R[m]$ versus m para o mapa MH com $q = 3$ e $N'_{min} = 11$.*

os mesmos parâmetros, obtendo na sequência de saída uma taxa de entropia maior que $0,9999$, como é mostrado nas Figuras 5.12 e 5.13.

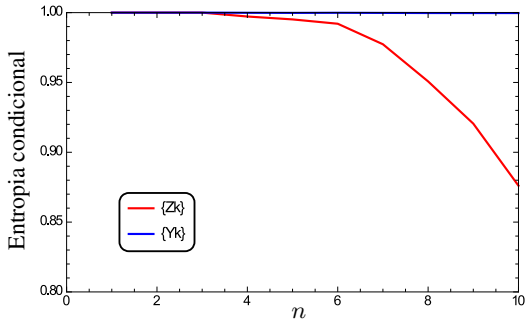


Figura 5.12: *Entropia condicional de* $\{Z_k\}$ *e* $\{Y_k\}$ *versus* n *para MC com* $q = 3$.

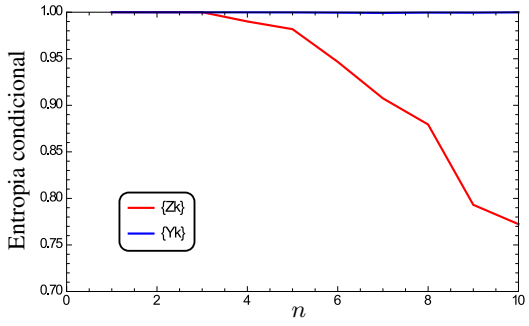


Figura 5.13: *Entropia condicional de* $\{Z_k\}$ *e* $\{Y_k\}$ *versus* n *para MH com* $q = 3$.

CAPÍTULO 6

CONCLUSÕES

ESTE capítulo sumariza as principais contribuições e resultados obtidos nesta dissertação e comenta sobre possíveis trabalhos futuros.

Nesta dissertação apresenta-se diferentes métodos de geração de sequências pseudo-aleatórias empregando mapas caóticos. A sequência binária caótica é obtida mediante um processo de quantização das amostras caóticas. Implementamos dois métodos de codificação: CFT e CVT. As sequências binárias obtidas de mapas caóticos, em geral, apresentam uma memória finita, característica não desejada em sequências aleatórias. Primeiramente utiliza-se um método de salto de amostras para eliminar esta memória. Para a utilização deste método empregamos a entropia condicional para determinar o limiar inferior do salto de amostras necessário para quebrar a correlação da sequência caótica. O emprego de salto de amostra leva a uma taxa de *bits* por amostra caótica baixa.

Para melhorar deficiência, implementamos uma unidade de pós-processamento baseada em LFSR para gerar sequências pseudo-aleatórias. Este bloco de pós-processamento tem uma taxa de *bits* unitária, ou seja, a quantidade de *bits* na entrada é igual ao da saída, e pode-se obter taxa de *bits* por amostra caóticas elevadas. A utilização da codificação CFT para valores elevados de q implica um aumento linear no valor do grau do polinômio gerador da sequência-m, levando ao aumento da ,memoria do sistema.

A função autocorrelação da sequência binária obtida pela codificação CVT tem um padrão de concentração para cada valor de q em forma de picos. Esta característica apresentada pela codificação CVT é aproveitada para a implementação de um novo bloco do pós-processamento que diminui a memoria do LFSR, em comparação à codificação CFT, obtendo um sistema com alta taxa de *bits* por amostra caótica e moderado requerimento de memória.

Um prosseguimento natural deste trabalho é um estudo de novos blocos de pós-processamento para melhorar as características aleatórias das sequências binárias caóticas. Também deve ser realizado um estudo teórico do comportamento da codificação CVT proposta, que permita uma otimização do pós-processamento. Como foi visto, um processo de quantização adequado melhoras as característica aleatórias das sequências binárias caóticas, e um estudo mais aprofundado deste processo deve ser realizado.

Publicações:

J.A.P. Artiles, D.P.B. Chaves, J.V.C. Evangelista, C. Pimentel. Uma Metodologia para Geração de Sequências Aleatórias usando Mapas Caóticos. *XXXIII Simpósio brasileiro de telecomunicações*. Juiz de Fora, MG. September 2015.

APÊNDICE A

TESTES DO NIST

Os testes da bateria NIST focam-se em uma variedade de características de não-aleatoriedade que existem nas sequências. Os 15 testes são [33]:

1. The Frequency Test.
2. Block Frequency Test.
3. The Run Test.
4. Tests for the Longest-Run-of-Ones in a Block.
5. The Binary Matrix Rank Test.
6. The Discrete Fourier Transform (Spectral) Test.
7. The Non-overlapping Template Matching Test.
8. The Overlapping Template Matching Test.
9. Maurer's "Universal Statistical"Test.
10. The Linear Complexity Test.
11. The Serial Test.
12. The Approximate Entropy Test.
13. The Cumulative Sums (Cusums) Test.
14. The Random Excursions Test.
15. The Random Excursions Variant Test.

Faz-se uma breve descrição dos conjunto de teste adotado pelo NIST [33].

The Frequency Test: Este teste determina se o número de uns e zeros é aproximadamente igual, que é o comportamento esperado das sequências verdadeiramente aleatórias. A implementação da suíte de teste do NIST depende da aprovação deste teste.

Block Frequency Test: O propósito deste teste é determinar se a frequência de zeros e uns em um bolco de comprimento M *bits* é aproximadamente $\frac{M}{2}$, como pode-se esperar de uma sequência aleatória. Para blocos de comprimento $M = 1$, o teste torna-se o teste anterior.

The Run Test: Neste teste avalia-se o número de *runs* na sequência testada. Um *run* é uma sequência ininterrupta de *bits* idênticos. Um *run* de comprimento k, consiste em k *bits* iguais limitados antes e depois por *bits* de outro valor. A proposta do teste é determinar se o número de *runs* de zeros e uns de vários comprimentos que acontecem na sequência testada e é esperado em uma sequência verdadeiramente aleatória, ou seja, ele determina se a oscilação entre zeros e uns é rápida ou lenta.

Tests for the Longest-Run-of-Ones in a Block: Este teste determina se o maior comprimento de uma sequência ininterrupta de uns (*run*) dentro da sequencia testada é consistente com o esperado para uma sequência verdadeiramente aleatória. Uma irregularidade no comprimento esperado do maior *run* de uns, implicará uma irregularidade nos *run* de zeros. Então só é necessário testar os *runs* de uns.

Binary Matrix Rank Test: O objetivo deste teste é verificar se há dependência linear entre as subsequências da sequência testada. Isto é feito calculando-se o posto de submatrizes formadas por subsequências disjuntas. Este teste também aparece na bateria de testes DieHard [34].

Discrete Fourier Transform (Spectral) Test: O objetivo deste teste é determinar aspectos periódicos da sequência testada, ou seja, padrões repetitivos que estejam próximos entre si. Estas características periódicas da sequência, se fossem detectadas, indicaria um afastamento da pressuposta aleatoriedade da sequência. A intenção é detectar se o número de picos no gráfico da transformada discreta de Fourier que excede o limite de 95% é significativamente maior que 5%.

Non-overlapping Template Matching Test: Neste teste detecta-se a ocorrência de padrões não-periódicos na sequência testada. A sequência é particionada em N blocos de comprimento M. Para este teste e o próximo, é construída uma janela de m *bits*. A janela é usada para encontrar a ocorrência de padrões. Se o padrão não for encontrado, a janela é deslocada de um bit. Se o padrão for encontrado, a janela é deslocada para o próximo *bit* depois da janela testada, e a busca do padrão recomeça.

Overlapping Template Matching Test: Tanto este teste como ou anterior usam uma janela de m *bits* para encontrar padrões não-periódicos na sequência testada. Tal como o teste anterior, se o padrão não é encontrado, a janela se desloca um *bit*. A diferença entre estes, é que, quando o padrão é encontrado, a janela se desloca apenas um *bit* e começa a busca novamente.

Maurers Universal Statistical Test: O objetivo deste teste é detectar se uma sequência pode ser comprimida sem perda de informação. Uma sequência significativamente compressível é considerada não aleatória.

Linear Complexity Test: Neste teste determina-se se a sequência testada é gerada por um gerador que possui complexidade linear suficiente para que esta seja considerada aleatória. Sequências aleatórias são caracterizadas por serem geradas por LFSR de períodos grandes, ou seja, possuem muitos elementos de retardo. Se o período do LFSR que gera a sequência é curto, a sequência é considerada não aleatória.

Serial Test: O objetivo deste teste é determinar se o número de ocorrências dos 2^m padrões sobrepostos de m *bits* é aproximadamente a mesma, o que é esperado para uma sequência aleatória. As sequências aleatórias apresentam a característica de uniformidade, ou seja, cada padrão de m *bits* tem a mesma probabilidade de acontecer. Note-se que para $m = 1$, este testes é equivalente ao teste The Frequency Test.

Approximate Entropy Test: Tal como o teste anterior, calcula-se a frequência de ocorrência de todos os padrões possíveis de m *bits* ao longo da sequência testada. O objetivo deste teste é comparar a frequência de ocorrência de blocos sobrepostos de dois comprimentos consecutivos, m e $m + 1$, com o resultado esperado para uma sequência verdadeiramente aleatória.

Cumulative Sums (Cusum) Test: O propósito deste teste é determinar se a soma cumulativa das subsequência que ocorrem na sequência testada é grande ou pequena em relação à soma cumulativa de sequências aleatórias. Se a soma cumulativa da sequência testada é muito diferente de zero, que é o valor esperado de uma sequência aleatória, a sequência apresenta um comportamento não-aleatório.

Random Excursions Test: Neste teste a sequência binária é convertida para uma sequência de $+1$ e -1. Forma-se uma soma acumulada de sequências e um ciclo corresponde à soma começar no valor zero e retorna ao valor zero. O propósito deste teste é determinar o desvio do número visitas em um estado particular em um ciclo em relação ao esperado para uma sequência verdadeiramente aleatória. Este teste consiste em uma série de oito testes, um para cada estado: $\{-4, -3, -2, -1e1, 2, 3, 4\}$.

Random Excursions Variant Test: O objetivo deste teste é o mesmo que o anterior, com a diferença que são considerados dezoito estados diferentes $\{-9, -8, -7, ...e..., 7, 8, 9\}$.

APÊNDICE B

CARACTERÍSTICAS PRINCIPAIS DOS LFSR

UMA maneira de obter sequências pseudo-aleatórias de período longo é a utilização de registros de deslocamento de retroalimentação linear (LFSR, *Linear feedback shift register*). Os LFSR são facilmente implementados em hardware e software. Embora um LFSR tenha uma retroalimentação linear, este produz uma sequência com boas propriedades estatísticas. Neste apêndice fazemos uma breve introdução aos LFSR e suas características principais.

Um LFSR compreende elementos de armazenamento e um circuito de retroalimentação. O número de elementos de armazenamento é o grau do LFSR. Em outras palavras, um LFSR com N estados de armazenamento tem grau N [38]. A forma geral de um LFSR de grau N é mostrada na Figura B.1. Os N estados de armazenamento são combinados através dos caminhos de retroalimentação. Estes podem estar ativos ou não, característica dada pelos coeficientes de retroalimentação $\{c_0, c_1, ..., c_{N-1}\}$, ou seja:

- ▷ se $c_i = 1$, o caminho de realimentação esta ativo.
- ▷ se $c_i = 0$, o caminho não é utilizado na realimentação.

Assumindo um LFSR com condições inicias $\{s_0, ..., s_{N-1}\}$, sua saída é $\{s_0, \cdots, s_{N-1}, s_N, s_{N+1}, \cdots\}$. O *bit* de saída do LFSR, s_N, é obtido da forma:

$$s_N \equiv s_{N-1}c_{N-1} + ... + s_1c_1 + s_0c_0 \mod 2. \tag{B.1}$$

A próxima saída do LFSR é:

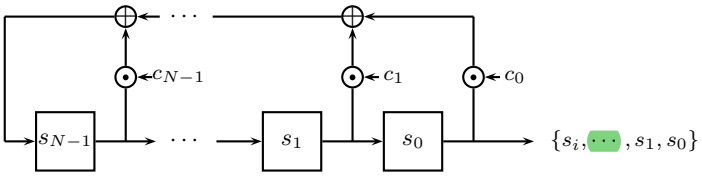


Figura B.1: *Diagrama de LFSR de grau N.*

$$s_{N+1} \equiv s_N c_{N-1} + \ldots + s_2 c_1 + s_1 c_0 \mod 2. \tag{B.2}$$

Em geral obtemos:

$$s_{i+N} \equiv \sum_{j=0}^{N-1} s_{i+j} c_j \mod 2, \tag{B.3}$$

com $\{s_i, c_j \in 0, 1, i = 0, 1, 2, \ldots\}$. Os LFSR muitas vezes são referenciados como recorrências lineares.

Um exemplo de um LFSR de grau $N = 3$ é ilustrado na Figura B.2. Os *bits* dos estados de armazenamento s_i são deslocados à direita, sendo o *bit* do estado mais a direita o *bit* de saída no próximo deslocamento. O *bit* do estado mais à esquerda é calculado pelo circuito de realimentação dado o estado de armazenamento anterior. Assumindo um estado inicial ($s_2 = 1$, $s_1 = 0$, $s_0 = 0$), obtemos uma sequência de saída da forma $\{00101110010111001011 \cdots\}$. Observe que a sequência de saída tem um período 7.

Devido ao número de estados de armazenamento ser finito, a sequência de saída de um LFSR vai se repetir periodicamente. Além disso, as sequências de saída de um LFSR podem ser de diferentes comprimentos, dependendo dos coeficientes de realimentação. O período máximo que pode ser obtido por um LFSR é uma função do seu grau, e é da forma [38]

$$p_{max} = 2^N - 1, \tag{B.4}$$

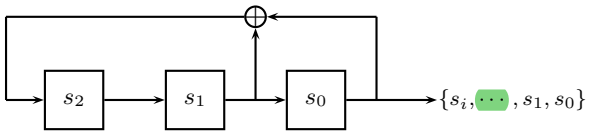


Figura B.2: *Diagrama de LFSR de grau $N = 3$ com estado inicial $\{s_2, s_1, s_0\}$.*

dado que um vetor de N *bits* só pode assumir 2^N estado. Se um LFSR assume o estado nulo, será "preso"no mesmo, ou seja, nunca será capaz de sair dele novamente, pois o comprimento máximo de execução antes da repetição é de $2^N - 1$ *bits*.

Um LFSR com coeficiente de realimentação dado por um vetor $(c_{M-1}, \cdots, c_1, c_0)$ pode ser representado de forma polinomial. Por exemplo, o LFSR do exemplo anterior, tem coeficientes $c_3 = 0$, $c_2 = 0$, $c_1 = 1$ e $c_0 = 1$. Alternativamente o LFSR pode ser representado em forma polinomial, sendo, $x^4 + x + 1$ o polinômio de retroalimentação. Uma das vantagens da notação polinomial é identificar se uma sequência é de período máximo. Isto ocorre se, e somente se, o polinômio de retroalimentação for um polinômio primitivo. A Tabela 3.9 mostra alguns polinômios primitivos para $N = \{2, 3, ..., 50\}$, utilizados nesta dissertação. Por exemplo, a notação $(5, 2, 0)$ refere-se ao polinômio $\{1 + x^2 + x^5\}$.

Uma das característica principais dos LFSR é a complexidade linear. Seja uma sequência finita $S = \{s_0, s_1, s_2, ..., s_n\}$ ou infinita $S = \{s_0, s_1, s_2, ...\}$, a complexidade lineal $(LC(S))$ dela é definida pelo LFSR de menor grau que gera a sequência S [39].

Exemplos:

▷ Se $\{S = 0000...01\}$ (com $[n - 1]$ zeros), a complexidade linear é igual a $LC(S) = n$, um polinômio de realimentação que gera a sequência S é $x^n + 1$.

▷ Utilizando o polinômio $[x^3 + x + 1]$ e condição inicial $[011]$, obtemos uma sequência de saída de $S' = \{0111001011\cdots\}$. A complexidade linear da sequência S' é inferior ou igual 3, pois o polinômio tem grau 3. Utilizando o critério acima, conclui-se que a complexidade linear é exatamente 3.

A complexidade linear de uma sequência S é dada por $LC(S) = L$. Considerando um LFSR de grau L que gera a sequência $S = \{s_0, \cdots, s_{L-1}, s_L, \cdots, s_{n-1}, s_n\}$, de comprimento n (onde n pode ser infinito), então:

▷ Os estados $[L]$ subsequentes do LFSR são linearmente independentes.

▷ O estado $[L + 1]$ subsequentes são linearmente dependentes.

▷ Se, além disso, pelo menos $2L$ termos da sequência são dados, isto é, $n > 2L$, pode-se determinar o polinômio de retroalimentação do LFSR.

SOBRE O AUTOR

O autor nasceu em Santa Clara, Villa Clara, Cuba, no dia 26 de Dezembro de 1984. Formou-se em Engenharia Elétrica, modalidade Eletricista, pela Universidade Central Das Villas em 2008. Seus interesses de pesquisa incluem Teoria da Informação, Códigos Corretores de Erro, Sistemas de Comunicação Digital, Criptografia e Processamento Digital de Sinais.

Endereço: Endereço

e-mail: japa.lepg@gmail.com

Esta dissertação foi diagramada usando LaTeX 2_ε[1] pelo autor.

[1] LaTeX 2_ε é uma extensão do LaTeX. LaTeX é uma coleção de macros criadas por Leslie Lamport para o sistema TeX, que foi desenvolvido por Donald E. Knuth. TeX é uma marca registrada da Sociedade Americana de Matemática ($\mathcal{AMS}$). O estilo usado na formatação desta dissertação foi escrito por Dinesh Das, Universidade do Texas. Modificado por Renato José de Sobral Cintra (2001) e por Andrei Leite Wanderley (2005), ambos da Universidade Federal de Pernambuco. Sua úlltima modificação ocorreu em 2010 realizada por José Sampaio de Lemos Neto, também da Universidade Federal de Pernambuco.

BIBLIOGRAFIA

[1] S.H. Strogatz. *Nonlinear Dynamics and Chaos with Applications to Physics, Biology, Chemistry, and Engineering*. Studies in Nonlinearity Series. Westview Press, 2001.

[2] K.T. Alligood, T.D. Sauer, and J.A. Yorke. *Chaos: An Introduction to Dynamical Systems*. New York, NY, 1997.

[3] S. Hayes, C. Grebogi, and E. Ott. Communicating with chaos. *Phys. Rev. Lett.*, 70:3031–3034, May 1993.

[4] C. Jianyong, Z Junwei, and Kwok-Wo W. A modified chaos-based joint compression and encryption scheme. *Circuits and Systems II: Express Briefs, IEEE Transactions on Circuit and Systems*, 58(2):110–114, February 2011.

[5] A. Masmoudi and W. Puech. Lossless chaos-based crypto-compression scheme for image protection. *IET Image Processing*, 8(12):671–686, 2014.

[6] F.C.M. Lau and C.K. Tse. *Chaos-Based Digital Communication Systems*. Engineering online library. Springer, 2010.

[7] M. Eisencraft, R. Attux, and R. Suyama. *Chaotic Signals in Digital Communications*. Electrical Engineering & Applied Signal Processing Series. Taylor & Francis, 2013.

[8] P. Stavroulakis. *Chaos Applications in Telecommunications*. Taylor & Francis, 2005.

[9] L. Kocarev and S. Lian. *Chaos-based Cryptography: Theory, Algorithms and Applications*. Studies in Computational Intelligence. Springer, 2011.

[10] F. Dachselt and W. Schwarz. Chaos and cryptography. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(12):1498–1509, February 2001.

[11] L. Kocarev, J. Makraduli, and P. Amato. Public-key encryption based on chebyshev polynomials. *Circuits, Systems and Signal Processing*, 24(5):497–517, October 2005.

[12] T. Stojanovski and L. Kocarev. Chaos-based random number generators-part I: analysis [cryptography]. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on Communication*, 48(3):281–288, March 2001.

[13] L. De Micco, H. A. Larrondo, A. Plastino, and O. A. Rosso. Quantifiers for randomness of chaotic pseudo-random number generators. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1901):3281–3296, August 2009.

[14] L. De Micco, C.M. González, H.A. Larrondo, M.T. Martin, A. Plastino, and O.A. Rosso. Randomizing nonlinear maps via symbolic dynamics. *Physica A: Statistical Mechanics and its Applications*, 387(14):3373 – 3383, June 2008.

[15] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2006.

[16] A. Beirami and H. Nejati. A framework for investigating the performance of chaotic-map truly random number generators. *IEEE Transactions on ciucuits and systems -II: Express Briefs*, 60(7):446 – 450, July 2013.

[17] H. Poincare. *The Value of Science: Essential Writings of Henri Poincare*. Modern Library Science, October 2001.

[18] J. Gleick. *Chaos: Making a New Science*. Penguin Books, 1987.

[19] R.M. May. Biological populations with nonoverlapping generations: Stable points, stable cycles and chaos. *Science*, 186(4164):645–647, November 1974.

[20] J. Wisdom and J. Stanton. The chaotic rotation of hyperion. *Physics Letters A*, 58(2):137–152, May 1984.

[21] A. Babloyantz and J.M. Salazar. Evidence of chaotic dynamics of brain activity during the sleep cycle. *Physics Letters A*, 111(3):152–156, September 1985.

[22] M. Kennedy, R. Rovatti, and G. Setti. *Chaotic Electronics In Telecommunications*. CRC Press, FL (USA), June 2000.

[23] M.P. Kennedy and L.O. Chua. Van der pol and chaos. *IEEE Trans. on Circuits and Systems,*, 33(10):974–980, October 1986.

[24] L.O. Chua. The genesis of chua's circuit. *Archiv fur Elektronik und Ubertragungstechnik*, 46(4):250–257, 1992.

[25] L.M. Pecora and T. L. Carroll. Synchronization in chaotic systems. *Physical Review Letters*, 64(8):821–825, February 1990.

[26] E. Ott, C. Grebogi, and J.A. Yorke. Controlling chaos. *Physical Review Letters*, 64:1196–1199, March 1990.

[27] G.S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Transactions of the American Institute of Electrical Engineers*, pages 295–301, 1926.

[28] D. Chaves, C. Souza, and C. Pimentel. A new map for chaotic communication. *International telecommunication Symposium (ITS 2014)*, pages 1 – 5, August 2014.

[29] L.A. Aguirre. *Introdução à Identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. UFMG, 2007.

[30] P. Glendinning. *Stability, Instability and Chaos: An Introduction to the Theory of Nonlinear Differential Equations.* Cambridge Texts in Applied Mathematics, December 1994.

[31] D.O. Pederson and K. Mayaram. *Analog Integrated Circuits for Communication: Principles, Simulation, and Design*. Kluwer Academic Publishers, 1991.

[32] P. Dudek and V.D. Juncu. Compact discrete-time chaos generator circuit. *Electronics Letters*, 39(20):1431–1432, October 2003.

[33] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. Statistical test suite for random and pseudo random number generators for cryptographic applications. *Special Publication 800-22 Revision 1a, National Institute of Standards and Technology*, April 2010.

[34] G. Marsaglia. Diehard statistical tests. 1995.

[35] D.E. Knuth. *The Art of Computer Programming: Seminumerical algorithms*. Addison-Wesley series in computer science and information processing. Addison-Wesley, 1981.

[36] W.G. Solomon and G Guang. *Signal Design for Good Correlation: For Wireless Communication, Cryptography, and Radar*. Cambridge University Press, July 2004.

[37] C.M. Lua and C.K Tse. *Chaos-Based Digital Communication Systems Operating Principles, Analysis Methods, and Performance Evaluation*. Signals and Communication Technology. Springer; 2003 edition, June 2003.

[38] C. Paar and J. Pelzl. *Understanding Cryptography, A Textbook for Students and Practitioners*. Springer, 2010.

[39] J.L. Massey. *Cryptography: Fundamentals and Applications. Copies of transparencies*. Advances Technology Seminars, 1997.