Equalização em sistemas de comunicação que utilizam sinais caóticos com algoritmos adaptativos baseados em kernel

Renato Candido, Magno T. M. Silva, Marcio Eisencraft

Escola Politécnica - Universidade de São Paulo

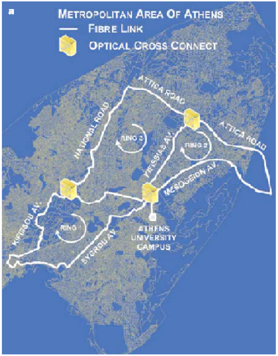
Novembro, 2017

Sumário

1. Introdução

- Sinais caóticos apresentam várias características interessantes para telecomunicações → Candidatos para sistemas de espalhamento espectral
- Pecora e Carroll, 1990 → Sincronismo caótico
- Várias ideias de sistemas na literatura

- Em comunicações ópticas: utilização de propriedades não lineares dos lasers para gerar caos
  - Argyris et al., 2005: sistema baseado em caos em canal comercial de fibra óptica



1. Sistema de comunicação de Wu e Chua

- Maneira simples de usar sinais caóticos para comunicações
- Verificação direta da convergência do erro de sincronismo para zero

Considerando

$$\begin{aligned} \text{Mestre:} \quad & \mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{b} + \mathbf{f}(x_i(n)) \\ \text{Escravo:} \quad & \widehat{\mathbf{x}}(n+1) = \mathbf{A}\widehat{\mathbf{x}}(n) + \mathbf{b} + \mathbf{f}(x_i(n)) \end{aligned}$$

Sendo:

- $\mathbf{x}(n)$ e $\widehat{\mathbf{x}}(n)$ vetores coluna de tamanho $K \times 1$
- $\mathbf{A}$ uma matriz quadrada e $\mathbf{b}$ um vetor coluna, constantes
- $\mathbf{f}(\cdot)$: $\mathbb{R} \to \mathbb{R}^K$ geralmente não linear, dependendo apenas de um componente de $\mathbf{x}(n)$, tendo a forma
$\mathbf{f}(x_i(n)) = [\underbrace{0\ 0\ \cdots\ 0}_{i-1 \text{ zeros}}\ f\left(x_i(n)\right)\ \underbrace{0\ 0\ \cdots\ 0}_{K-i \text{ zeros}}]^T$

1. Sistema de comunicação de Wu e Chua (2)

$$\begin{aligned} \text{Mestre:} \quad & \mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{b} + \mathbf{f}(x_i(n)) \\ \text{Escravo:} \quad & \widehat{\mathbf{x}}(n+1) = \mathbf{A}\widehat{\mathbf{x}}(n) + \mathbf{b} + \mathbf{f}(x_i(n)) \end{aligned}$$

Erro de sincronismo:

$$\begin{aligned} \mathbf{e}(n) &\triangleq \widehat{\mathbf{x}}(n) - \mathbf{x}(n) \\ \mathbf{e}(n+1) &= \mathbf{A}\mathbf{e}(n) \end{aligned}$$

Se os autovalores λ_i de $\mathbf{A}$ satisfizerem $|\lambda_i| < 1, 1 \leq i \leq K$:

$$\mathbf{e}(n) \to \mathbf{0} \Rightarrow \text{Sincronismo caótico}$$

1. Sistema de comunicação de Wu e Chua (3)

$$\text{Mestre:} \quad \mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{b} + \mathbf{f}(x_i(n))$$
$$\text{Escravo:} \quad \widehat{\mathbf{x}}(n+1) = \mathbf{A}\widehat{\mathbf{x}}(n) + \mathbf{b} + \mathbf{f}(x_i(n))$$

Sistema de transmissão de informações de Wu e Chua

- Codifica-se $m(n)$ usando o i-ésimo componente de $\mathbf{x}(n)$:

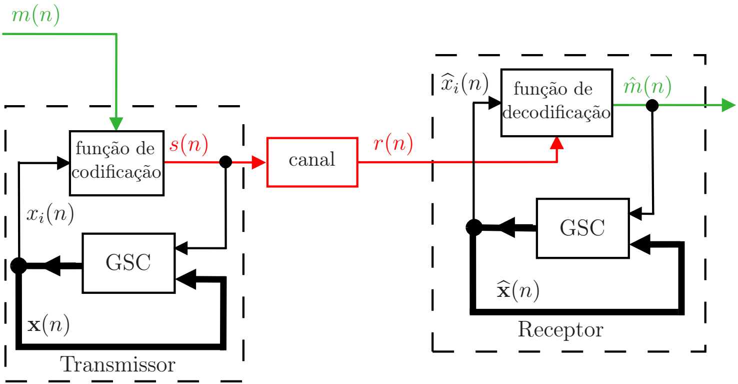
$$s(n) = c(x_i(n), m(n))$$

- De forma que se possa recuperar

$$m(n) = c^{-1}\left(\widehat{x}_i(n),\, s(n)\right)$$

$$\text{Transmissor:} \quad \mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{b} + \mathbf{f}(s(n))$$
$$\text{Receptor:} \quad \widehat{\mathbf{x}}(n+1) = \mathbf{A}\widehat{\mathbf{x}}(n) + \mathbf{b} + \mathbf{f}(s(n))$$

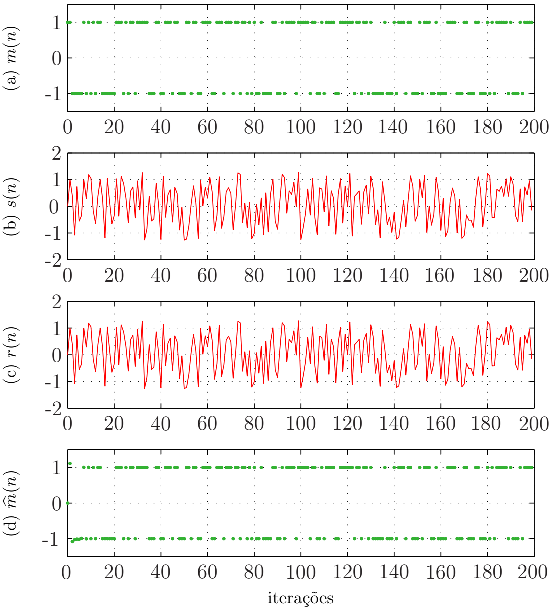
1. Sistema de comunicação de Wu e Chua (4)



$$\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{b} + \mathbf{f}(s(n))$$
$$\widehat{\mathbf{x}}(n+1) = \mathbf{A}\widehat{\mathbf{x}}(n) + \mathbf{b} + \mathbf{f}(r(n))$$

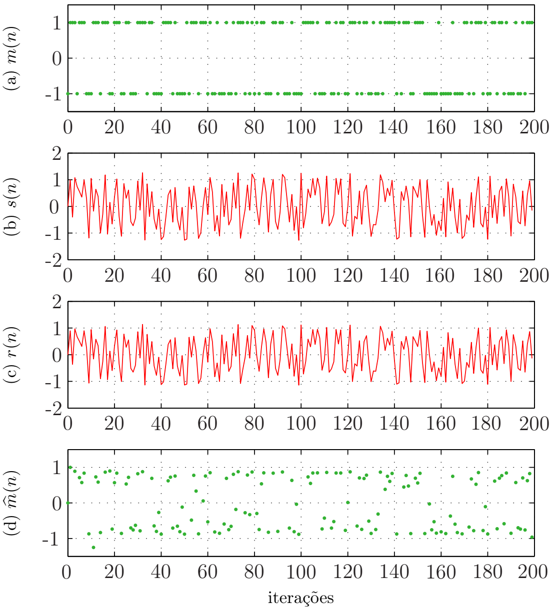
1. Sistema de comunicação de Wu e Chua (5)

- Mapa de Hénon no GSC
- Canal ideal
- $s(n) = m(n)x_1(n)$ e $\widehat{m}(n) = r(n)/\widehat{x}_1(n)$



1. Sistema de comunicação de Wu e Chua (6)

- Canal $H(z) = 0{,}9$
  $r(n) = 0{,}9s(n)$
- Sincronismo caótico sensível a canal não ideal



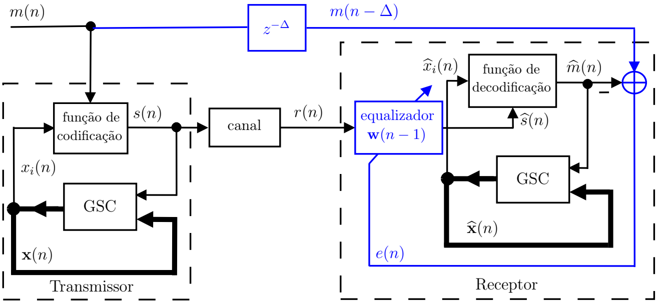
1. SCBC com equalizador

Comp. Appl. Math.
DOI 10.1007/s40314-017-0519-9

CrossMark

A new encoding and switching scheme for chaos-based communication

Renato Candido[1] · Magno T. M. Silva[1] · Marcio Eisencraft[1]

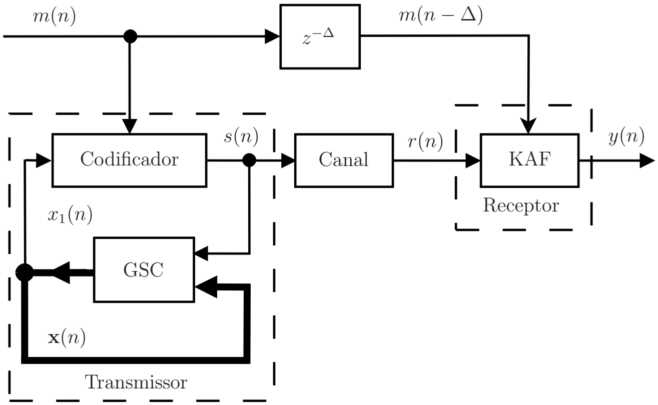


$$\mathbf{r}(n) = [r(n), r(n-1), \cdots r(n-M+1)]^T$$

$$s(n) = \gamma_1\, x_1(n) - \gamma_2\,[m(n)+1]\,\text{sign}[\gamma_1\, x_1(n)] \qquad \widehat{s}(n) = \mathbf{r}^T(n)\mathbf{w}(n-1)$$

$$\widehat{m}(n) = c^{-1}(\widehat{x}_i(n), \widehat{s}(n)) = \frac{\gamma_1 \widehat{x}_1(n) - \widehat{s}(n)}{\gamma_2 \text{sign}[\gamma_1 \widehat{x}_1(n)]} - 1 \qquad e(n) = m(n-\Delta) - \widehat{m}(n) \qquad \text{cNLMS}_+$$

1. Ideia deste trabalho

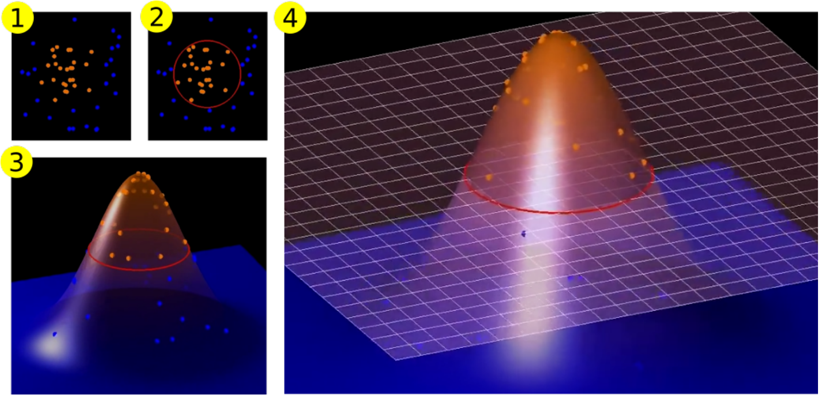


- Tratar o modulador caótico e o canal de comunicação como um sistema não linear
- Usar um filtro adaptativo não linear para obter $m(n)$ a partir de $r(n)$

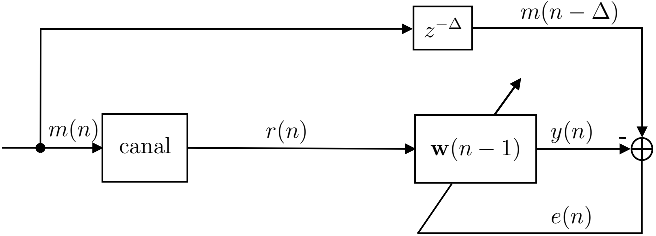
Sumário

2. Métodos baseados em *kernel*

- Transformação dos dados de entrada para um espaço de dimensão maior



2. Algoritmo LMS



$$\mathbf{r}(n) = [r(n), r(n-1), \cdots r(n-M+1)]^T$$

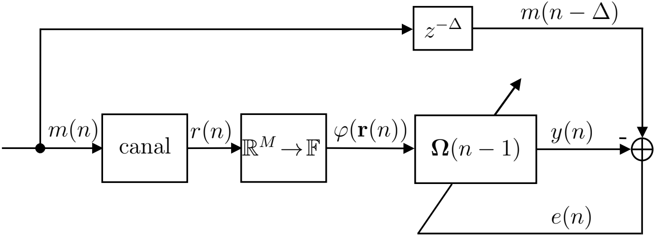
$$\mathbf{w}(n) = [w_0(n), w_1(n), \cdots w_{M-1}(n)]^T$$

$$y(n) = \mathbf{r}(n)^T \mathbf{w}(n-1) \quad e(n) = m(n-\Delta) - y(n)$$

Para minimizar $J = e^2(n)$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu e(n)\mathbf{r}(n)$$

2. Algoritmo KLMS



$$\mathbf{r}(n) = [r(n), r(n-1), \cdots r(n-M+1)]^T$$

$$y(n) = \varphi(\mathbf{r}(n))^T \mathbf{\Omega}(n-1) \quad e(n) = m(n-\Delta) - y(n)$$

$\mathbf{\Omega}(n) = \mathbf{\Omega}(n-1) + \mu e(n)\varphi(\mathbf{r}(n))$

Fazendo $\mathbf{\Omega}(0) = \mathbf{0}$ e reescrevendo

$$y(n) = \sum_{i=1}^{n-1} \mu e(i)\kappa(\mathbf{r}(n), \mathbf{r}(i))$$

Necessário dicionário $\{\mathbf{r}(i), e(i)\}_{i=1}^{n-1}$

Para limitar o tamanho do dicionário:

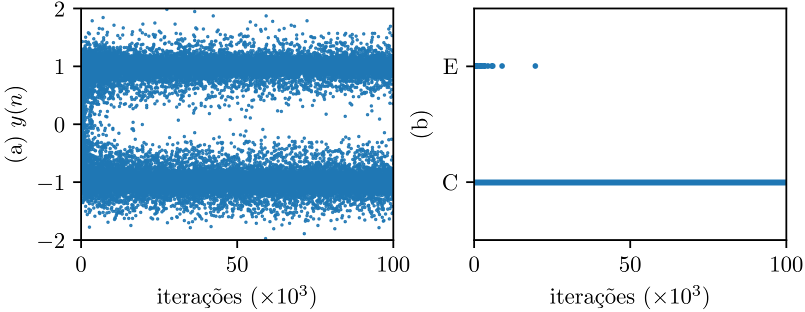
Técnicas de esparsificação

$\Rightarrow$ QKLMS

Sumário

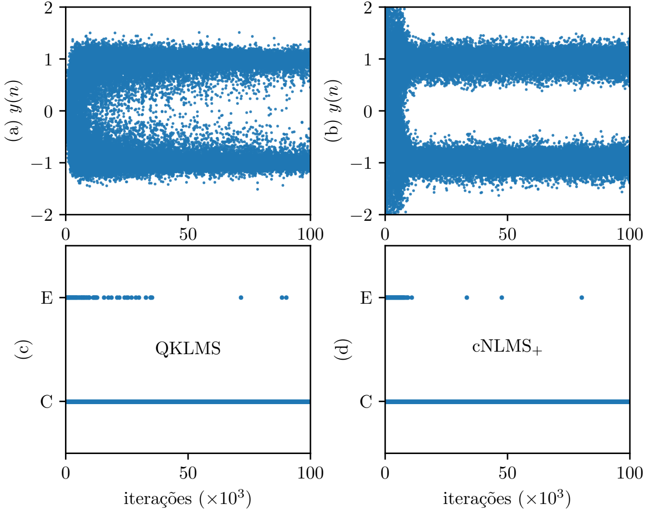
3. Decodificação da mensagem

- Canal ideal: KAF apenas para decodificação da mensagem
- $M = 3$, *kernel* Gaussiano com $\sigma = 0{,}1$, $\mu = 0{,}5$



3. Equalização e decodificação da mensagem

- Canal $H(z) = 0{,}25 + z^{-1} + 0{,}25z^{-2}$, atraso $\Delta = 1$
- $M = 5$, *kernel* Gaussiano com $\sigma = 0{,}1$, $\mu = 0{,}5$



Sumário

5. Conclusões preliminares

- O algoritmo QKLMS é capaz de equalizar o canal e decodificar a mensagem para o SCBC
  - Não é necessária nenhuma informação sobre o transmissor
  - Pode ser uma abordagem interessante para estudo quanto à privacidade do sistema

5. Conclusões preliminares

- O algoritmo QKLMS é capaz de equalizar o canal e decodificar a mensagem para o SCBC
  - Não é necessária nenhuma informação sobre o transmissor
  - Pode ser uma abordagem interessante para estudo quanto à privacidade do sistema
- O QKLMS apresenta velocidade de convergência menor que a do cNLMS$_+$

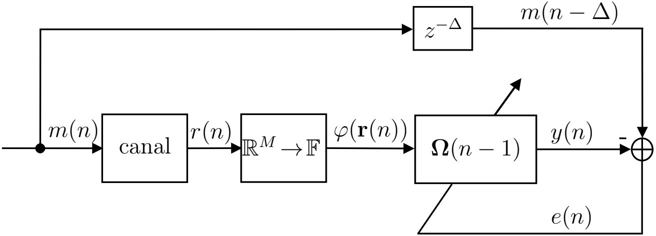
5. Conclusões preliminares

- O algoritmo QKLMS é capaz de equalizar o canal e decodificar a mensagem para o SCBC
  - Não é necessária nenhuma informação sobre o transmissor
  - Pode ser uma abordagem interessante para estudo quanto à privacidade do sistema
- O QKLMS apresenta velocidade de convergência menor que a do cNLMS_+
- O custo computacional do QKLMS é mais alto:
  - Necessidade do dicionário
  - No exemplo de equalização, o dicionário ficou com 25676 elementos
  - O tempo de execução foi 400 vezes maior que o do cNLMS_+

Obrigado!

Renato Candido - renatocan@lps.usp.br

Algoritmo KLMS



$$\mathbf{r}(n) = [r(n), r(n-1), \cdots r(n-M+1)]^T$$

$$y(n) = \varphi(\mathbf{r}(n))^T \mathbf{\Omega}(n-1) \quad e(n) = m(n-\Delta) - y(n)$$

$$\mathbf{\Omega}(n) = \mathbf{\Omega}(n-1) + \mu e(n) \varphi(\mathbf{r}(n))$$

Fazendo $\mathbf{\Omega}(0) = \mathbf{0}$

$$\underbrace{\varphi(\mathbf{r}(n))^T \mathbf{\Omega}(n-1)}_{y(n)} = \sum_{i=1}^{n-1} \mu e(i) \varphi(\mathbf{r}(n))^T \varphi(\mathbf{r}(i))$$

Truque de *kernel*: $\varphi(\mathbf{r}(n))^T \varphi(\mathbf{r}(i)) = \kappa(\mathbf{r}(n), \mathbf{r}(i))$

$$y(n) = \sum_{i=1}^{n-1} \mu e(i) \kappa(\mathbf{r}(n), \mathbf{r}(i))$$

Necessário dicionário $\{\mathbf{r}(i), e(i)\}_{i=1}^{n-1}$

Para limitar o tamanho do dicionário:

Técnicas de esparsificação

$\Rightarrow$ QKLMS

Kernel Gaussiano

Um dos *kernels* mais utilizados é o *kernel* Gaussiano:

$$\kappa(\mathbf{r}, \mathbf{r}') = \exp\left(-\frac{\|\mathbf{r} - \mathbf{r}'\|^2}{2\sigma^2}\right)$$

O *kernel* Gaussiano representa uma transformação para um espaço de dimensão infinita:

Considerando a constante igual a 1 e o caso escalar:

$$\exp(-(r - r')^2) = \exp(-r^2)\exp(-r'^2)\underbrace{\exp(2rr')}_{\sum_{k=0}^{\infty}\frac{2^k r^k r'^k}{k!}}$$

$$= \exp(-r^2)\exp(-r'^2)\sum_{k=0}^{\infty}\frac{\sqrt{2^k}r^k}{\sqrt{k!}}\frac{\sqrt{2^k}r'^k}{\sqrt{k!}}$$

$$= \varphi(r)^T\varphi(r')$$

$$\varphi(r) = \exp(-r^2)\begin{bmatrix}\frac{\sqrt{2^0}r^0}{\sqrt{0!}} \\ \frac{\sqrt{2^1}r^1}{\sqrt{1!}} \\ \vdots\end{bmatrix}$$

$$\varphi(r') = \exp(-r'^2)\begin{bmatrix}\frac{\sqrt{2^0}r'^0}{\sqrt{0!}} \\ \frac{\sqrt{2^1}r'^1}{\sqrt{1!}} \\ \vdots\end{bmatrix}$$