

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

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

Universidade de São Paulo, São Paulo, Brasil

e-mail: renatocan@lps.usp.br

Abstract - *Many communication systems based on the synchronization of chaotic systems have been proposed in the literature. However, due to the lack of robustness of chaos synchronization, in most cases even minor channel imperfections are enough to hinder communication. In this paper, we propose the use of kernel adaptive filters to decode the message and simultaneously equalize the communication channel. Preliminary results show that the proposed solution is able to perform these tasks without any knowledge of the chaotic map, which is essential to other adaptive solutions contained in the literature.*

Introdução

Sistemas de comunicação baseados em sincronização de sistemas caóticos têm sido amplamente investigados na literatura (ver, e.g., [1–7] e suas referências). Um sistema caótico gera, de forma determinística, trajetórias no espaço de estados que são aperiódicas, limitadas e que apresentam dependência sensível às condições iniciais [8]. Portanto, sinais caóticos podem ser utilizados como portadoras de banda larga para sinais de informação com o potencial de apresentar um alto nível de privacidade na transmissão dos dados [4, 5].

Uma das maiores desvantagens dos sistemas de comunicação baseados em caos é a pouca robustez do sincronismo caótico em relação ao nível de ruído e à interferência intersimbólica (ISI) introduzidos pelo canal. Mesmo um pequeno nível de ruído ou distorções simples podem ser suficientes para impedir a comunicação [7, 9, 10]. A fim de mitigar a ISI introduzida pelo canal, esquemas de equalização aplicados a sistemas de comunicação baseados em caos foram propostos na literatura, utilizando-se diferentes abordagens de codificação da mensagem (veja, por exemplo, [11–16] e referências contidas). Dentre essas referências, apenas [7] considera a equalização aplicada no domínio do tempo discreto para a modulação caótica que realimenta a sequência transmitida no gerador de sinais caóticos (GSC). No entanto, a recuperação da mensagem proposta em [7] depende do bom desempenho do equalizador, que por sua vez, utiliza

o mapa caótico, pois o mesmo GSC do transmissor é utilizado no receptor.

Como alternativa aos equalizadores adaptativos de [7], propõe-se neste trabalho utilizar algoritmos adaptativos baseados em núcleo (*kernel adaptive filters* – KAFs) no receptor. KAFs são capazes de resolver problemas não lineares, projetando-se implicitamente as amostras de entrada em um espaço de dimensão maior, onde os problemas podem ser resolvidos linearmente [17]. Dessa forma, ao se utilizar um KAF em um sistema de comunicação baseado em caos, deixa de ser necessário considerar o GSC no receptor.

O artigo está organizado da seguinte forma: descrevem-se em seguida o sistema de comunicação baseado em caos e o algoritmo adaptativo baseado em *kernel* utilizado. Resultados de simulação e as conclusões do trabalho são então apresentados.

Formulação do Problema

Na Figura 1, é mostrado o sistema de comunicação considerado. Como GSC, utiliza-se o mapa de Hénon [18], cuja equação é dada por

$$\mathbf{x}(n) = \mathbf{A}\mathbf{x}(n-1) + \mathbf{b} + \mathbf{f}(s(n)), \quad (1)$$

sendo $\mathbf{x}(n) = [x_1(n) \ x_2(n)]^T$,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ \beta & 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{f}(s(n)) = \begin{bmatrix} -\alpha s^2(n) \\ 0 \end{bmatrix},$$

α e β constantes reais e $(\cdot)^T$ a operação de transposição. A mensagem binária $m(n) \in \{-1, +1\}$ é codificada utilizando-se $x_1(n)$, i.e.,

$$s(n) = \eta_1 x_1(n) - \eta_2 [m(n) + 1] \text{sign}[\eta_1 x_1(n)], \quad (2)$$

em que $\{\eta_1, \eta_2\} \subset [0, 1]$ e $\text{sign}[\cdot]$ é a função sinal. Esta codificação foi proposta em [7] para diminuir o distúrbio causado pela mensagem em $x_1(n)$ e aumentar o espaço dos parâmetros $\{\eta_1, \eta_2\}$ onde o sinal gerado é caótico. Neste caso, se $m(n) = -1$, $s(n) = \eta_1 x_1(n)$ e se $m(n) = 1$ uma constante com o sinal oposto a $\eta_1 x_1(n)$ é adicionado a ele.

Em seguida, o sinal $s(n)$ é realimentado e transmitido por um canal de comunicação, cujo

modelo é constituído por uma função de transferência $H(z)$ e ruído gaussiano branco e aditivo. A fim de se recuperar a mensagem transmitida, utiliza-se no receptor um filtro adaptativo baseado em *kernel* com entrada $\mathbf{r}(n) = [r(n) r(n-1) \dots r(n-M+1)]^T$ em que $M \geq 1$ e saída $y(n)$, que representa uma estimativa da mensagem transmitida com um atraso de Δ amostras. Assume-se que existe uma sequência de treinamento de modo que $m(n-\Delta)$ faz papel do sinal desejado $d(n)$ e o erro $e(n) = m(n-\Delta) - y(n)$ é utilizado para adaptar o filtro.

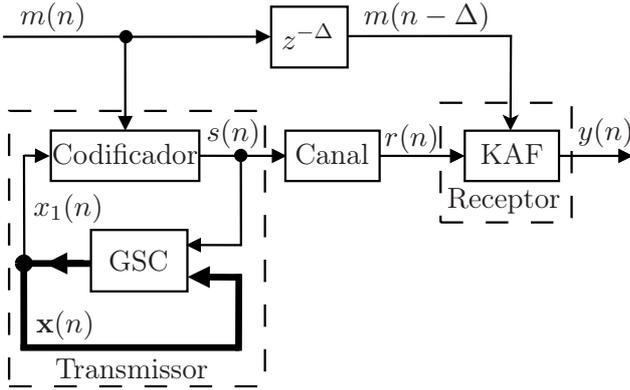


Figura 1 – Sistema de comunicação caótico com um equalizador adaptativo.

Filtragem Adaptativa baseada em *kernel*

A versão do algoritmo LMS (*least-mean-square*) com *kernel*, denotada como KLMS [19], mapeia o vetor coluna de entrada $\mathbf{r}(n) \in \mathbb{U} \subset \mathbb{R}^M$ em um espaço de características de dimensão alta \mathbb{F} como $\varphi(\mathbf{r}(n))$, usando um *kernel* de Mercer, que por sua vez, é uma função contínua, simétrica e positiva-definida $\kappa: \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{R}$, de modo que vale [17]

$$\kappa(\mathbf{r}, \mathbf{r}') = \varphi(\mathbf{r})^T \varphi(\mathbf{r}'). \quad (3)$$

No espaço \mathbb{F} , o algoritmo LMS é usado para atualizar o vetor coluna dos pesos $\mathbf{\Omega}(n-1)$ a fim de estimar o sinal desejado $d(n) \in \mathbb{R}$, que depende não linearmente da entrada $\mathbf{r}(n)$, i.e.,

$$\mathbf{\Omega}(0) = \mathbf{0}, \quad (4)$$

$$e(n) = d(n) - \varphi(\mathbf{r}(n))^T \mathbf{\Omega}(n-1), \quad (5)$$

$$\mathbf{\Omega}(n) = \mathbf{\Omega}(n-1) + \mu e(n) \varphi(\mathbf{r}(n)), \quad (6)$$

em que μ é um passo de adaptação. Usando manipulações algébricas, podemos escrever

$$\mathbf{\Omega}(n) = \sum_{i=1}^n \mu e(i) \varphi(\mathbf{r}(i)), \quad (7)$$

que nos permite reescrever a estimativa de $d(n)$ como

$$y(n) = \varphi(\mathbf{r}(n))^T \mathbf{\Omega}(n-1) = \sum_{i=1}^{n-1} \mu e(i) \kappa(\mathbf{r}(n), \mathbf{r}(i)). \quad (8)$$

Usando (8) e definindo $a(i) \triangleq \mu e(i)$, o algoritmo KLMS pode ser resumido como

$$a(1) = \mu d(1), \quad (9)$$

$$y(n) = \sum_{i=1}^{n-1} a(i) \kappa(\mathbf{r}(n), \mathbf{r}(i)), \quad (10)$$

$$e(n) = d(n) - y(n), \quad (11)$$

$$a(n) = \mu e(n). \quad (12)$$

Um dos *kernels* mais utilizados é o gaussiano, definido como

$$\kappa(\mathbf{r}, \mathbf{r}') = \exp\left(-\frac{\|\mathbf{r} - \mathbf{r}'\|^2}{2\sigma^2}\right), \quad (13)$$

em que $\sigma > 0$ é a largura do *kernel* [17]. Como é o mais utilizado, apenas esse *kernel* será considerado neste trabalho.

Um dos principais desafios da filtragem adaptativa baseada em *kernel* é reduzir o tamanho do dicionário a fim de diminuir o custo computacional e a memória. O dicionário é definido como o conjunto de vetores de suporte $\mathcal{C}(n) = \{\mathbf{r}(c_j)\}_{j=1}^{N_c(n)}$, em que $\mathbf{r}(c_j)$ é o seu j -ésimo elemento e $N_c(n)$ é a sua cardinalidade, que pode variar de um instante de tempo a outro. O índice $c_j \in \{1, 2, \dots, n-1\}$ é usado para distinguir os elementos do dicionário $\mathbf{r}(c_1), \dots, \mathbf{r}(c_{N_c(n)})$ do vetor de entrada $\mathbf{r}(n)$. Sem nenhuma restrição de dicionário, podemos observar de (8) que para estimar $d(n)$, devemos usar todos os vetores de entrada desde a inicialização do algoritmo até o instante de tempo imediatamente anterior, i.e, $\mathcal{C}(n) = \{\mathbf{r}(i)\}_{i=1}^{n-1}$ e, neste caso, o tamanho do dicionário $N_c(n) = n-1$ cresce linearmente com as iterações. Para evitar esse crescimento linear, diferentes técnicas de esparsificação para atualizar o dicionário foram propostas na literatura, incluindo apenas dados que contenham informação, como o critério da novidade [20] e o critério da coerência [21].

Uma alternativa a essas técnicas é a utilização do algoritmo KLMS com dicionário reduzido, proposto em [22] e denotado por QKLMS (*quantized KLMS*). Para decidir se o vetor de entrada deve ou não ser acrescentado ao dicionário, define-se

$$\text{dis}(\mathbf{r}(n), \mathcal{C}(n)) = \min_{1 \leq j \leq N_c(n)} \|\mathbf{r}(n) - \mathbf{r}(c_j)\|,$$

em que $\|\cdot\|$ denota a norma euclidiana. Se $\text{dis}(\mathbf{r}(n), \mathcal{C}(n)) \leq \varepsilon$, sendo ε um limiar, mantém-se o dicionário inalterado e atualiza-se o coeficiente do vetor mais próximo como $a(j^*) \leftarrow a(j^*) + \mu e(n)$, em que $j^* = \arg \min_{1 \leq j \leq N_c(n)} \|\mathbf{r}(n) - \mathbf{r}(c_j)\|$. Caso contrário, acrescenta-se o vetor de entrada ao dicionário e seu respectivo coeficiente deve ser guardado na memória, i.e., $\mathcal{C}(n+1) = \{\mathcal{C}(n), \mathbf{r}(n)\}$ e $a(N_c(n)+1) = \mu e(n)$. Esse procedimento leva a um algoritmo similar ao KLMS esparsificado com o critério da novidade. Assim, uma precisão melhor e um dicionário mais compacto podem ser encontrados [22]. Devido às suas vantagens inerentes, vamos considerar apenas o QKLMS neste trabalho.

Resultados

A fim de verificar o funcionamento do esquema proposto foi considerado um sistema de comunicação utilizando o mapa de Hénon com parâmetros $\alpha = 1,4$ e $\beta = 0,3$. O vetor de estados foi inicializado com $\mathbf{x}(0) = \mathbf{0}$ e foi considerada a transmissão de uma sequência binária. O sinal $s(n)$ foi gerado considerando $\eta_1 = 0,9$ e $\eta_2 = 0,3$, o que assegura que ele seja caótico, i.e., apresenta dependência sensível às condições iniciais [7]. Inicialmente, considerou-se o KAF para a decodificação da mensagem. Para tanto, foi considerado um canal de comunicação ideal, com $r(n) = s(n)$ e a sequência de treinamento $d(n) = m(n - \Delta)$ com o atraso $\Delta = 0$. Na Figura 2, são mostrados: (a) a saída $y(n)$ do KAF e (b) os erros obtidos após o decisor. Pode-se notar que o algoritmo QKLMS é capaz de decodificar corretamente a mensagem a partir do sinal caótico $s(n)$, sem o conhecimento de parâmetros do transmissor, como o mapa utilizado e a função de codificação.

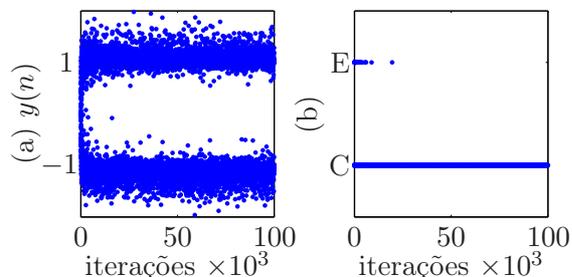


Figura 2 – (a) Saída do QKLMS para decodificação da mensagem. (b) Erros (E) e acertos (C) na mensagem recuperada. Receptor QKLMS com $M = 3$, $\mu = 0,5$, $\sigma = 0,1$ e $\varepsilon = 0,1$.

Em seguida, para verificar o funcionamento do sistema para a equalização de um canal de co-

municação, foi considerada a transmissão do sinal caótico $s(n)$ pelo canal

$$H(z) = 0,25 + z^{-1} + 0,25z^{-2}, \quad (14)$$

na ausência de ruído. No receptor, foi utilizado o algoritmo QKLMS considerando a sequência de treinamento $d(n) = m(n - \Delta)$ com o atraso $\Delta = 1$ amostra. Na Figura 3, são mostrados: (a) a mensagem recuperada $y(n)$ e (c) os erros obtidos após o decisor. Para efeito de comparação, também são mostrados: (b) a saída e (d) e os erros obtidos após o decisor, utilizando o equalizador proposto em [7].

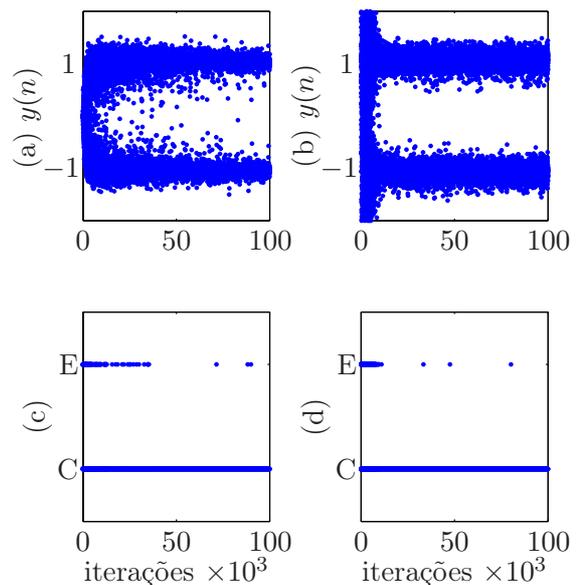


Figura 3 – (a) Saída do QKLMS ($M = 5$, $\mu = 0,5$, $\sigma = 0,1$ e $\varepsilon = 0,05$) e (c) erros (E) e acertos (C) na mensagem recuperada. (b) Saída do esquema de equalização de [7] ($M = 5$, $\tilde{\mu} = 0,01$, $\delta = 10^{-5}$ e $\varepsilon = 0,1$) e (d) erros e acertos na mensagem recuperada. Sinal $s(n)$ transmitido pelo canal $H(z) = 0,25 + z^{-1} + 0,25z^{-2}$ na ausência de ruído.

Pode-se notar que o algoritmo QKLMS é capaz de equalizar o canal de comunicação e simultaneamente decodificar a mensagem transmitida com o sinal caótico. Em comparação ao equalizador utilizado em [7], é possível notar que o QKLMS apresenta uma velocidade de convergência menor e um maior custo computacional. Isso se deve à necessidade da utilização do dicionário com $N_c(n)$ vetores para o cálculo da saída $y(n)$. No exemplo mostrado na Figura 3, o QKLMS finalizou com um dicionário com 25676 vetores e o tempo de execução foi cerca de 400 vezes maior do que o do equalizador de [7]. Em contrapartida, ele necessita apenas de um sinal de treinamento para realizar a equalização do canal e a decodificação da mensagem, não sendo necessária nenhuma informação à respeito do transmissor como o mapa utilizado no GSC e a função de codificação.

Conclusões

Neste trabalho, a mensagem transmitida por um sistema de comunicação que utiliza sinais caóticos foi recuperada de forma supervisionada utilizando-se um filtro adaptativo baseado em *kernel* no receptor. Resultados preliminares de simulação mostram que esse filtro é capaz de recuperar a mensagem transmitida sem o conhecimento do mapa caótico e da função de codificação, às custas de uma complexidade computacional mais elevada. Para um canal ideal, verificou-se que o filtro também pode ser usado para decodificação da mensagem, fazendo o papel do “escravo” no sistema de comunicação de Wu e Chua. Isso mostra que o alto nível de privacidade na transmissão dos dados nem sempre acontece no caso de sistemas de comunicação baseado em caos.

Agradecimentos

Ao CNPq pelo apoio financeiro (304275/2014-0, 311575/2013-7 e 309275/2016-4).

Referências

- [1] L. M. Pecora and T. L. Carroll, “Synchronization in chaotic systems,” *Phys. Rev. Lett.*, vol. 64, pp. 821–824, Feb. 1990.
- [2] A. V. Oppenheim *et al.*, “Signal processing in the context of chaotic signals,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, vol. 4, pp. 117–120, 1992.
- [3] K. M. Cuomo and A. V. Oppenheim, “Chaotic signals and systems for communications,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, vol. 3, pp. 137–140, Apr. 1993.
- [4] M. Feki *et al.*, “Secure digital communication using discrete-time chaos synchronization,” *Chaos, Solitons & Fractals*, vol. 18, pp. 881–890, Nov. 2003.
- [5] A. Argyris *et al.*, “Chaos-based communications at high bit rates using commercial fibre-optic links,” *Nature*, vol. 438, no. 7066, pp. 343–346, 2005.
- [6] J. Grzybowski, M. Eisencraft, and E. Macau, “Chaos-based communication systems: Current trends and challenges,” in *Applications of Chaos and Nonlinear Dynamics in Engineering - Vol. 1* (S. Banerjee, M. Mitra, and L. Rondoni, eds.), vol. 71 of *Understanding Complex Systems*, pp. 203–230, New York: Springer-Verlag, 2011.
- [7] R. Candido, *A questão da equalização em sistemas de comunicação que utilizam sinais caóticos*. Tese de Doutorado, Escola Politécnica da Universidade de São Paulo, São Paulo, 2014.
- [8] K. T. Alligood, T. Sauer, and J. A. Yorke, *Chaos: An Introduction to Dynamical Systems*. Textbooks in Mathematical Sciences, New York: Springer-Verlag, 1997.
- [9] C. Williams, “Chaotic communications over radio channels,” *IEEE Trans. Circuits Syst. I*, vol. 48, pp. 1394–1404, Dec. 2001.
- [10] M. Eisencraft, R. D. Fanganiello, and L. H. A. Monteiro, “Chaotic synchronization in discrete-time systems connected by bandlimited channels,” *IEEE Commun. Lett.*, vol. 15, pp. 671–673, Jun. 2011.
- [11] H. Leung, “System identification using chaos with application to equalization of a chaotic modulation system,” *IEEE Trans. Circuits Syst. I*, vol. 45, pp. 314–320, Mar. 1998.
- [12] Z. Zhu and H. Leung, “Adaptive blind equalization for chaotic communication systems using extended-Kalman filter,” *IEEE Trans. Circuits Syst. I*, vol. 48, no. 8, pp. 979–989, 2001.
- [13] J. Feng, C. K. Tse, and F. C. M. Lau, “Reconstruction of chaotic signals with application to channel equalization in chaos-based communication systems,” *International J. Commun. Syst.*, vol. 17, pp. 217–232, 2004.
- [14] M. Ciftci and D. Williams, “Iterative equalization for chaotic communications systems,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, vol. 4, pp. 165–68, 2005.
- [15] C. Vural and G. Çetinel, “Blind equalization of single-input single-output fir channels for chaotic communication systems,” *Digital Signal Process.*, vol. 20, no. 1, pp. 201–211, 2010.
- [16] R. Candido, M. Eisencraft, and M. T. M. Silva, “Channel equalization for chaotic communications systems,” in *Chaotic Signals in Digital Commun.* (M. Eisencraft, R. Attux, and R. Suyama, eds.), CRC Press, Inc., 2013.
- [17] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*. Wiley, 2010.
- [18] M. Hénon, “A two-dimensional mapping with a strange attractor,” *Commun. in Mathematical Physics*, vol. 50, pp. 69–77, 1976.
- [19] W. Liu, P. P. Pokharel, and J. C. Príncipe, “The kernel least-mean-square algorithm,” *IEEE Trans. Signal Process.*, vol. 56, pp. 543–554, Feb. 2008.
- [20] J. Platt, “A resource-allocating network for function interpolation,” *Neural Comput.*, vol. 2, pp. 213–225, 1991.
- [21] C. Richard, J. C. M. Bermudez, and P. Honeine, “Online prediction of time series data with kernels,” *IEEE Trans. Signal Process.*, vol. 57, pp. 1058–1067, Mar. 2009.
- [22] B. Chen *et al.*, “Quantized kernel least mean square algorithm,” *IEEE Trans. on Neural Netw. and Learning Syst.*, vol. 23, pp. 22–32, Jan. 2012.