Synchronizing Spatiotemporal Chaos by Introducing a Finite Flat Region in the Local Map

J. Y. CHEN[a],*, K. W. WONG[a], H. Y. ZHENG[b] and J. W. SHUAI[c]

[a]*Department of Electronic Engineering, City University of Hong Kong;*
[b]*Department of Electronic Engineering, Xiamen University, China;*
[c]*Department of Biomedical Engineering, Case Western Reserve University,
Cleveland, Ohio 44106, USA*

An approach to synchronize spatiotemporal chaos is proposed. It is achieved by introducing a finite flat region in the local map. By using this scheme, a number of orbits in both the drive and the response subsystems are forced to pass through a fixed point in every dimension. With only an arbitrary phase space variable as drive signal, synchronization of spatiotemporal chaos can be achieved rapidly in the response subsystem. This is an advantage when compared with other synchronization methods that require a linear combination of the original phase space variables.

Keywords: Chaos; Synchronization; Control; Discrete; Hyperchaos

PACS numbers: 05.45.+b

Recently, there has been a growing interest in the dynamics and the synchronization of chaos due to its potential application in various areas, such as secure communications, neural networks and pattern formation *etc*. [1–9]. The fact that coupling between the drive and the response subsystems can lead to synchronization was first discovered by Pecora and Carroll [1]. This approach has been further extended to cascade synchronization of chaos with multiple stable subsystems [10–12]. It has been shown that it is possible to synchronize chaos with a high-dimensional hyperchaotic coupled map lattice using a single scalar variable that is a linear combination of the original phase space variables [16, 17]. To enhance the security, an auxiliary chaotic subsystem is used as a common driver to synchronize two identical subsystems [18]. However, these feedback methods are effective only in a particular range of parameters. To synchronize hyperchaotic

---

* Address for correspondence: Dept. of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong. e-mail: cjyok@yahoo.com

systems, an approach utilizing convex combinations of the drive and the response subsystems has been proposed [19]. Nevertheless, an approach to achieve synchronization of high-dimensional spatiotemporal systems without the convex combinations is not yet found.

Based on the drive-and-response synchronization and the feedback methods, we propose to introduce a finite flat region in the local map of both the drive and the response subsystems. Complete synchronization can be achieved for hyperchaotic systems. A typical drive-and-response system is described by the following equations.

$$X(n + 1) = F[X(n)] \tag{1}$$

$$Y(n + 1) = F[Y(n)] + g(Y(n), X(n)) \tag{2}$$

where the vector $X(n) = (x_1(n), x_2(n), \Lambda, x_i(n))$ corresponds to the drive subsystem while $Y(n) = (y_1(n), y_2(n), \Lambda, y_i(n))$ is the vector of the response subsystem. $g(Y(n), X(n))$ is a driving term. Here $i = 1, 2, \Lambda$ M is lattice index and n is the iteration index. For hyperchaotic systems with positive conditional Lyapunov exponents, the drive-and-response scheme cannot achieve synchronization. In order to solve this problem, a finite flat region is introduced in the local map of the drive and the response subsystems in every iteration. This is achieved by the following modification.

$$x_i(n) = \begin{cases} p & p - \delta \leq x_i(n) \leq p + \delta, \\ x_i(n) & Otherwise \end{cases} \tag{3}$$

$$y_i(n) = \begin{cases} p & p - \delta \leq y_i(n) \leq p + \delta, \\ y_i(n) & Otherwise \end{cases} \tag{4}$$

In particular, p is a value chosen from the range of the chaotic attractors while δ is a sufficiently small value that controls the width of the flat region. With our method, the value of p can be selected arbitrarily within the output range of the drive and the response subsystems. However, it should have the same value in both subsystems. If the output value $x_i(n)$ or $y_i(n)$ fall between $p - \delta$ and $p + \delta$, it is set to the fixed value p. Otherwise it remains unchanged. With this modification, the two subsystems can achieve synchronization.

In this paper, the one-way coupled map lattice [20] that has been extensively investigated is discussed as an example to show the effectiveness of the proposed scheme. The systems can still achieve synchronization even when noise is induced. A typical M-dimensional one-way coupled map lattice is described by the following equations.

$$x_i(n + 1) = (1 - \varepsilon)f(x_i'(n)) + \varepsilon f(x_{i+1}'(n))$$
$$x_i'(n) = \begin{cases} p & p - \delta \leq x_i(n) \leq p + \delta, \\ x_i(n) & x_i(n) < p - \delta, x_i(n) > p + \delta, \end{cases}$$
$$x_{M+1}(n) = x_1(n) \quad i = 1, 2, \ldots, M \tag{5}$$

Where $x_i'(n)$ is the controlled result of $x_i(n)$ in each iteration and ε is the coupling strength in the range $0 < \varepsilon < 1$. Here we choose $f(x) = \alpha x(1 - x)$ with $\alpha = 4.0$, $M = 100$ and the scalar signal $x_1(n)$ is used as the driving signal to replace $y_1(n)$ in response subsystem. As the value of $x_i(n)$ is between 0 and 1, we let $p = 0.8$ and $\varepsilon = 0.25$. These settings are used throughout this paper unless specified. When the synchronization is achieved, the modification does not vanish and the system's attractors will still be affected. However, if δ is small, the effect is negligible.

Let $\delta = 0.01$, simulation results show that the synchronization occurs between the drive and the response subsystems. Figure 1 gives the plot of time *versus* the average absolute difference ρ between the two subsystems. The measure ρ is defined as

$$\rho = \frac{1}{M} \sum_{i=2}^{M} |x_i(n) - y_i(n)| \tag{6}$$

We observe from Figure 1 that with the increase of iteration time, ρ as well as the amplitude of the perturbation decreases continuously. At time $n = 27,550$, all the fields in the coupled map lattices
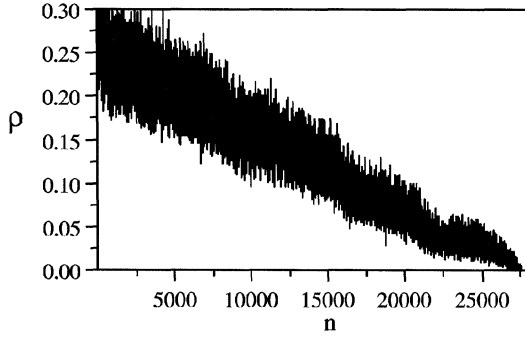
FIGURE 1 A plot of time *versus* the average absolute difference ρ with $\varepsilon = 0.25$, $p = 0.8$ and $M = 100$.

are in synchronization and the average absolute difference is smaller than 10^{-18}, which cannot be distinguished in our computer.

After the synchronization has been reached, the modification does not vanish. As a result, the dynamical structure of the modified local map will be different from the original one. In order to investigate the corresponding consequence, we calculate the percentage of the fields in the drive subsystem that are set to p.

$$k = \frac{\sum_{n=1}^{N} \sum_{i=1}^{M} (1|_{x_i(n)=p})}{NM} \tag{7}$$

Here N is the total number of iterations required to achieve full synchronization. In order to show the relation between N and k, we plot these two variables against δ in Figures 2(a) and (b), respectively, with precision $\Delta\delta = 0.0005$. When $\delta = 0$, it is a typical drive-and-response system without any finite flat region and so no synchronization is observed for positive conditional Lyapunov exponents. In this case, the percentage of the modified fields is zero and the iterations required can be considered as $N \to +\infty$. When $\delta = 0.005$, the percentage of the modified fields is $k = 1.89\%$. This means that the systems are only slightly modified and so the number of iterations to achieve synchronization is very large ($N = 86,706$). In the case of a relatively large flat region such as $\delta = 0.05$, 15.6% of the fields are affected. The corresponding number of iterations required
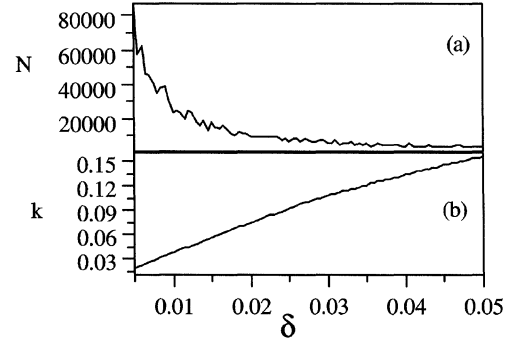

FIGURE 2 (a,b) Plots with the parameter δ *versus* (a) the number of iterations N required for synchronization, and (b) the percentage of modified fields k. The two subsystems are one-way coupled map lattices with $M = 100$ and $\varepsilon = 0.25$. The location of the finite flat region is $p = 0.8$.

is only 3,698, which is very fast for a 100-dimensional system. If δ is further increased, the synchronizing speed can be even faster. However, the effect caused by the flat region will be so apparent that the resultant systems will become considerably different from the original one.

Without the modification, the two subsystems can only achieve synchronization when $\varepsilon > 0.358$. However if the scheme is applied, synchronization can be achieved for all ε in the range $0 < \varepsilon < 1$. A typical synchronization process with $\delta = 0.01$ is shown in Figure 3. In this figure, we observed that the iteration time required for synchronization decreases rapidly with ε.

In order to show the effect caused by the finite flat region, we take $\delta = 0$, 0.02 and 0.05 and mark the location of the iterating points in the $x_i - x_{i+1}$
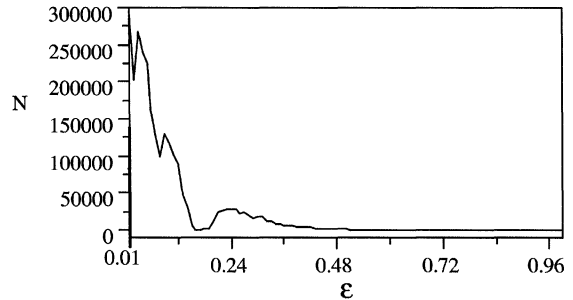

FIGURE 3 A plot of the coupling strength ε *versus* the number of iterations N required to achieve synchronization.

plane. They are plotted in Figures 4(a) – (c), respectively. In all the three cases, $i = 1$, $\varepsilon = 0.25$ and $p = 0.8$. The data in these figures are obtained from 6,000 iterations, after omitting 1,000 initial transients. In Figure 4(a), no flat region is introduced and the points are solely produced by the one-way coupled map lattices. Although there are two thin lines in Figure 4(b), the distribution of points are approximately the same as that in Figure 4(a). This shows that the effect introduced by the finite flat region is not obvious. In Figure 4(c), there are two obvious straight lines perpendicular to each other, with crossing point at (0.8, 0.8). The width of the finite flat region near the two straight lines is $2\delta (\sim 0.1)$. Comparing with Figure 4(a) that without the introduced flat region, the points in Figure 4(c) are evidently less evenly distributed. This is caused by the relatively large flat region.

In order to explore the mechanism to synchronization, a two-dimensional system is analyzed. At first, the drive and the response subsystems are independent but both have finite flat regions. The mechanism is shown in Figure 5 in which the flat regions in the dimensional space of $x_1 - x_2$ plane are marked by horizontal and vertical solid lines. Figure 5(a) represents the drive subsystem while Figure 5(b) corresponds to the response one. An orbit of the drive subsystem enters the x_1-axis flat region at time t_1 and is changed to a new start point (p, x_2). At a later time t_2, suppose that the orbit of the response subsystem is also in the flat
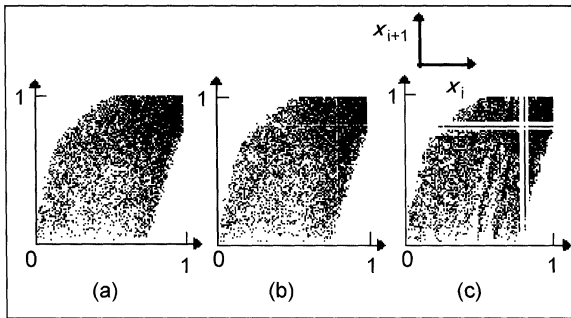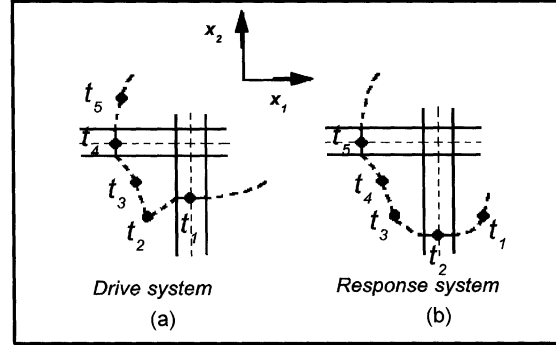

FIGURE 5 The path to synchronization in the flat region of the (a) drive subsystem and (b) the response one.

region. Although the two orbits may be different when entering the flat region, they both have the same start point p on the x_1-axis. Similarly, when the orbit enters the x_2-axis flat region, it is changed to a new start point (x_1, p). Thus, the orbit in every dimension is forced to pass the particular value p at different time. When the process continues, both the orbits of the drive and the response subsystems will become nearer and nearer. At last, they follow the same path, but with a certain amount of time delay. If a drive signal such as x_1 or x_2 is sent from the drive subsystem to the response one, this time delay vanishes rapidly and finally the two subsystems will be in synchronization. The proposed method makes the orbits of the two subsystems restart at the same value in each dimension within a limited number of iterations. This can be achieved by keeping ρ so small that cannot be distinguished by the computer. Thus the synchronization can be maintained.

The mechanism of synchronization in higher-dimensional systems is similar to that described above. As flat regions are introduced in every dimension, an increase in the number of dimensions only results in a longer iteration time to make the drive and the response subsystems run on the same orbit. Similarly, the time delay in the drive and the response subsystems can be reduced to zero by a single drive signal since the response subsystem is coupled in a cascaded manner $x(M) \to x(M-1) \to \Lambda \to x(1)$.   For

FIGURE 4 A plot of iterating points in the $x_i - x_{i+1}$ plane when (a) $\delta = 0$, (b) $\delta = 0.02$ and (c) $\delta = 0.05$. The other parameters are $M = 100$, $i = 1$ and $\varepsilon = 0.25$.

high-dimensional hyperchaotic systems, their orbits are more unstable or sensitive and hence a longer iteration time is required. This is proved by the simulation results. Let the number of dimensions is raised from 3 to 100, the number of iterations N required for synchronization increases from 832 to 25,528, as shown in Figure 6.

The robustness of the synchronization can be tested by noise perturbation. There is a study that if a small perturbation $\xi \in (-10^{-30}, 10^{-30})$ is added to the drive signal $x_1(n)$ with restriction $0 < x_1(t) + \xi < 1$, the synchronization is totally destroyed when $\varepsilon < 0.32$. In our study, we use a stronger perturbation $\xi \in (-10^{-18}, 10^{-18})$ to test the robustness of the synchronization caused by the flat regions. After the synchronization is achieved, the perturbation is added to the drive signal. Different simulations are carried out and the results are plotted in Figure 7. In Figure 7(a), the parameters chosen are $M = 4$ and $\delta = 0.01$. The corresponding graph shows that $\ln(\rho)$ approaches zero at $n = 1,500$ and thus the synchronization is destroyed. However, if δ is increased slightly to 0.02, the synchronization becomes more robust, as shown in Figure 7(b). This implies that increasing he value of δ makes systems more robust. Furthermore, if δ is kept at 0.02 and M is increased from 4 to 10 (Fig. 7(c)), the synchronization lost again. This is due to the increased number of dimensions. From the above result, it is reasonable to predict that a further increase of δ will cause the two subsystems with dimension $M = 10$ to synchronize again. The plot in Figure 7(d) has proved this.
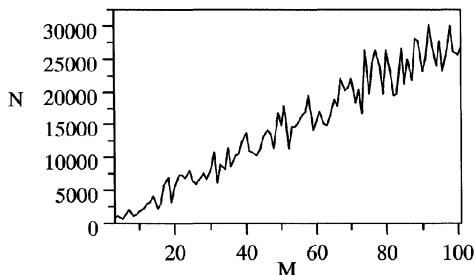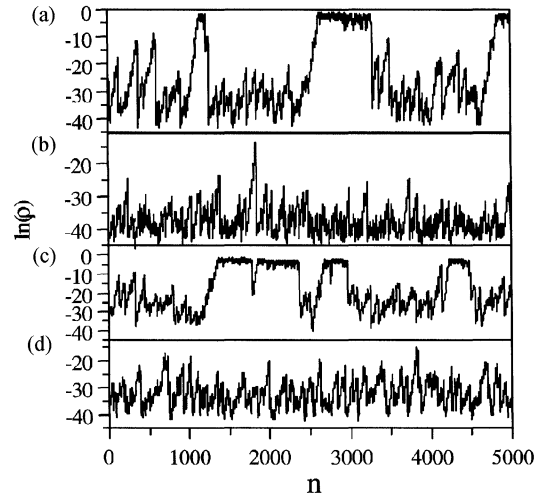

FIGURE 7 (a–d) Plots of the number of iterations *versus* the logarithm of the average absolute difference ρ under the perturbation $\xi \in (-10^{-18}, 10^{-18})$. Both the two identical subsystems are simulated with $\varepsilon = 0.25, p = 0.8$. (a) The number of dimension of two subsystems is $M = 4$ and $\delta = 0.01$. The synchronization is destroyed. (b) With $M = 4$ and $\delta = 0.02$, the synchronization is robust. (c) $M = 10$ and $\delta = 0.02$, the synchronization is destroyed. (d) $M = 10$ and $\delta = 0.03$, the synchronization is robust again.

As a result, we conclude that by introducing flat region in the local map, increasing the value of δ can strengthen the robustness of synchronization while an increase in the number of dimensions will destroy it.

As the introduction of finite flat regions only deals with the output of subsystems, the exact dynamics of the subsystems can be unknown. This characteristics is important in certain practical applications that the system dynamics are not well understood.

In conclusion, following the drive-and-response scheme proposed by Pecora and Carroll [1], we introduce finite flat regions in the local map in order to synchronize hyperchaotic systems using only a single scalar signal. The relation between the width of the flat region and the time required for synchronization is investigated. The larger the flat region the faster the synchronizing process. A tradeoff is that the dynamics of the attractors may be affected. Furthermore, a two-dimensional system is chosen as an example to analyze the

FIGURE 6 A plot of the number of dimensions in both subsystems *versus* the number of iterations N required for synchronization.

mechanism to synchronization when the proposed approach is applied. The relations between the number of iterations for synchronization, the width of the flat region and the number of dimensions in the subsystems are analyzed. Finally, the synchronization with noise perturbation is investigated. Simulations show that with appropriate width of flat regions and suitable number of dimension, the synchronization can be robust. This synchronization method is useful to models whose exact dynamics are not well understood.

References

[1] Pecora, L. M. and Carroll, T. L. (1990). Synchronization in chaotic systems. *Physical Review Letters*, **64**, 821–824.

[2] Josic, K. (1998). Invariant manifolds and synchronization of coupled dynamical systems. *Physical Review Letters*, **80**, 3053–3056.

[3] Shuai, J. W., Wong, K. W. and Cheng, L. M. (1997). Synchronization of spatiotemporal chaos with positive conditional Lyapunov exponents. *Physical Review E*, **56**, 2272–2275.

[4] Shuai, J. W. and Wong, K. W. (1998). Noise and synchronization in chaotic neural networks. *Physical Review E*, **57**, 7002–7007.

[5] Chen, J. Y., Chen, Z. X., Shuai, J. W., Lu, Y. Y. and Guo, D. H. (1998). A chaotic lattice and the circuit of its unit. *Canadian Journal of Physics*, **76**, 801–808.

[6] Lee, K. J., Kwak, Y. and Lim, T. K. (1998). Phase jumps near a phase synchronization transition in systems of two coupled chaotic oscillators. *Physical Review Letters*, **81**, 321–324.

[7] Baker, G. L., Blackburn, J. A. and Smith, H. J. T. (1998). Intermittent synchronization in a pair of coupled chaotic pendula. *Physical Review Letters*, **81**, 554–557.

[8] Chen, J. Y., Wong, K. W., Chen, Z. X., Xu, S. C. and Shuai, J. W. (2000). Phase synchronization in discrete chaotic systems. *Physical Review E*, **61**, 2559–2562.

[9] Mensour, B. and Longtin, A. (1998). Synchronization of delay-differential equations with application to private communication. *Physics Letters A*, **244**, 59–70.

[10] Cuomo, K. M. and Oppenheim, A. V. (1993). Circuit implementation of synchronized chaos with applications to communications. *Physical Review Letters*, **71**, 65–68.

[11] He, R. and Vaidya, P. G. (1992). Analysis and synthesis of synchronous periodic and chaotic systems. *Physical Review A*, **46**, 7387–7392.

[12] Chen, J. Y., Wong, K. W. and Shuai, J. W. (1999). Finding the chaotic synchronizing state with gradient descent algorithm. *Physics Letters A*, **263**, 315–322.

[13] Kittel, A., Pyragas, K. and Richter, R. (1994). Prerecorded history of a system as an experimental tool to control chaos. *Physical Review E*, **50**, 262–268.

[14] Rong He and Vaidya, P. G. (1998). Implementation of chaotic cryptography with chaotic synchronization. *Physical Review E*, **57**, 1532–1535.

[15] Xiao, J. H., Hu, G. and Qu, Z. L. (1996). Synchronization of spatiotemporal chaos and its application to multi-channel spread-spectrum communication. *Physical Review Letters*, **77**, 4162–4165.

[16] Johnson, G. A., Mar, D. J., Carroll, T. L. and Pecora, L. M. (1998). Synchronization and imposed bifurcations in the presence of large parameter mismatch. *Physical Review Letters*, **80**, 3956–3959.

[17] Peng, J. H., Ding, E. J., Ding, M. and Yang, W. (1996). Synchronizing hyperchaos with a scalar transmitted signal. *Physical Review Letters*, **76**, 904–907.

[18] Murali, K. and Lakshmanan, M. (1998). Secure communication using a compound signal from generalized synchronizable chaotic systems. *Physics Letters A*, **241**, 303–310.

[19] Gutieerez, J. M. and Iglesias, A. (1998). Synchronizing chaotic systems with positive conditional Lyapunov exponents by using convex combinations of the drive and response systems. *Physics Letters A*, **239**, 174–180.

[20] Kaneko, K. (1989). Chaotic but regular posi-nega switch among coded attractors by cluster-size variation. *Physics Review Letters*, **63**, 219–222.

[21] Zhou, C. S. and Lai, C. H. (1998). Synchronization with positive conditional Lyapunov exponents. *Physical Review E*, **58**, 5188–5191.