

Fichiers de paramétrisation

Paramétrisation des cibles

Si les variables suivantes sont fixées avant *cibleMGXXX.m* ou *cibleLorenz.m*, celles-ci remplacent les valeurs par défaut présentes dans le script correspondant : *CI*, *h*, *T_tot*.

D'autre part, *cibleMGXXX.m* proposent aussi la modification des variables *tau* et *ChangeScaleMG* qui leur sont propres tandis que *cibleLorenz.m* propose en plus *ChangeScaleLorenz* comme variable propre à ce système.

Paramétrisation des réservoirs

Si les variables suivantes sont fixées avant *GenResXXX.m*, celles-ci remplacent les valeurs par défaut présentes dans le script correspondant : *N*, *rho*, *gainIn*, *gainFb*, *delta*, *C*, *a*, *LvlNoise*.

Prédiction

L'équation d'évolution du réservoir utilisée est contenue dans le fichier *majRes.m* et est donnée par l'équation 21 de « The "echo state" approach to analysing and training recurrent neural networks » (Jaeger 2010) :

$$x(n + 1) = (1 - \delta C a)x(n) + \delta C f(W_{in} u(n + 1) + Wx(n) + W_{back} y(n))$$

Pour lancer un entraînement, faites appel au fichier *mainPredict.m*. Ce programme vous invite à choisir entre 3 systèmes pour ensuite faire appel à plusieurs sous-routines¹, dans l'ordre :

- (1) La fonction cible pour l'apprentissage forcé est construite dans les fichiers *cibleXXX.m*.
- (2) Les paramètres de l'équation d'évolution et constituant le réservoir sont définis dans les fichiers *genResXXX.m*
- (3) Le nombre de points rejeté, d'entraînement ou pour l'évolution libre sont posés dans les fichiers *train(Version)XXX.m* où « Version » n'est pas toujours présent mais peut apparaître avec les valeurs « Simple » et « Avance »².
- (4) L'évaluation des erreurs pendant l'entraînement et l'évolution libre sont calculées dans le fichier *calcErreursTrain.m*.
- (5) Pour tous les systèmes XXX, un test du caractère chaotique peut être fait via l'activation de *testChaos.m*.
- (6) Pour MG, le fichier *testAttracteursMG.m* permet de comparer les attracteurs cible et prédit.

Verrouillage

Les routines de verrouillage sont implémentées pour MG et pour Lorenz. Le fichier *mainLock.m* invite l'utilisateur à choisir parmi 6 possibilités :

- (1) Verrouillage d'un MG sur un autre (*MGLockMG.m*) ;
- (2) Verrouillage d'un MG sur un réservoir entraîné pour imiter un MG (*MGLockRC.m*) ;
- (3) Verrouillage d'un réservoir entraîné pour imiter un MG sur un MG (*RClockMG.m*) ;
- (4) Verrouillage d'un réservoir entraîné pour imiter un MG sur un autre RC (*RClockRC.m*).
- (5) Verrouillage d'un Lorenz sur un autre (*LoLockLo.m*) ;
- (6) Verrouillage d'un réservoir entraîné pour imiter un Lorenz sur un Lorenz (*RClockLo.m*) ;

La variable $q \in [0 ; 1]$ conditionnant la proportion de verrouillage et l'amplitude du bruit sont définis séparément dans ces fichiers. L'équation pour le verrouillage était :

$$\tilde{S}_2(t + 1) = (1 - q)f[t, \tilde{S}_2(t)] + qS_1(t + 1) + v(t)$$

À noter que les fichiers peuvent être lancés directement (même après modification de q par exemple) sans passer par le programme principal si ce dernier a déjà tourné une fois pour ce cas avant.

¹ Certains (comme le MG) permettent de choisir (en jonglant avec les lignes commentées ou non) différentes routines prédéfinies.

² La méthode « Avance » est obsolète et doit être mise à jour avant d'être utilisable.

Cryptographie

L'interception de messages transmis par un algorithme basé sur le chaos se fait via l'appel de [mainCrypto.m](#). Ce fichier propose deux méthodes de communication : la superposition (avec MG [superposition.m](#) ou Lorenz [superpositionLorenz.m](#)) et le mélange non-linéaire (avec MG [melange.m](#)).

Ci-dessous, les paramètres pouvant être modifiés dans ces fichiers :

- ▷ Superposition : `h, bitRepete, nbrBitLock, nbrBit, nbrBitTrain, A, A_eps ;`
- ▷ Mélange : `nbrSinus, omega, h, B, fc_ex, fm_ex, fc_msg, fm_msg, inputFactor, tau, A_filtre, A_eps.`

En outre, ces deux routines font appel aux programmes précédents : leurs paramètres peuvent alors également être modifiés.

Finalement, le fichier [LorenzSynchro.m](#) implémente la méthode de Cuomo³ en synchronisant deux Lorenz.

³ K. M. Cuomo et al., *Synchronization of Lorenz-Based Chaotic Circuits with Applications to Communications*, IEEE Trans. Circ. Syst., 40:626-633, 1993b.