

## Paramétrisation des cibles

---

Si les variables suivantes sont fixées avant *cibleMGXXX.m* ou *cibleLorenz.m*, celles-ci remplacent les valeurs par défaut présentent dans le script correspondant : Cl, 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 présentent 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 aux fichiers *mainPredictXXX.m*. Ce programme fait appel à plusieurs sous-routines<sup>1</sup>, 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 »<sup>2</sup>.
- (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*.  
Pour MG, le fichier *testAttracteursMG.m* permet de comparer les attracteurs cible et prédit.

## Verrouillage

---

Les routines de verrouillage sont implémentées pour MG uniquement. Le fichier *mainLock.m* invite l'utilisateur à choisir parmi 4 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 (*MGLockRC.m*) ;
- (4) Verrouillage d'un réservoir entraîné pour imiter un MG sur un autre RC (*RCLockRC.m*).

La variable  $q \in [0 ; 1]$  conditionnant la proportion de verrouillage est définie séparément dans ces fichiers.

À noter que *MGLockMG.m* peut être lancé directement sans passer par le programme principal. Pour les autres cas, le fichier correspondant peut être lancé 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.

---

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

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

## Cryptographie

---

L'interception de message 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 ([superposition.m](#)) et le mélange non-linéaire ([melange.m](#)).

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

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

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