Hackaton AI4IA - Equipe MIA

A. Gardille, F. Najar, A. Forveille, R. Dakhmouche

November 10, 2021

Une explication de l'évolution des stratégies est faite en détails dans le notebook scientifique, on présente ici la stratégie choisie parmi celles testées.

1 Modelisation

On suppose que chacun des outputs correspond à la réponse d'un système physique ou numérique à une entrée, l'input.

On ne connait pas la forme de la réponse de ces systèmes. Cependant, on suppose que localement vis à vis de l'input, ces systèmes se comportent comme des systèmes linéaires invariant du temps (LTI), c'est à dire qui vérifient une équation de la forme:

$$a_N \frac{d^N y(t)}{dt^N} + \cdots + a_1 \frac{dy(t)}{dt} + a_0 y(t) = b_N \frac{d^N x(t)}{dt^N} + \cdots + b_1 \frac{dx(t)}{dt} + b_0 x(t),$$

où $x(t)$ désigne l'entrée du système, $y(t)$ sa sortie et N l'ordre du système.

Pour analyser ces systèmes, une technique couramment utilisée consiste à passer dans le domaine de Laplace, on note $\mathcal{L}$ la transformation de Laplace. Notons $X(p) = \mathcal{L}[x(t)]$ et $Y(p) = \mathcal{L}[y(t)]$ les transformées de Laplace de l'entrée et de la sortie. En utilisant le fait que la transformée de Laplace est linéaire et que $\mathcal{L}[\frac{d^k e(t)}{dt^k}] = p^k E(p)$, il est possible de décrire la système par la fonction suivante :

$$H(p) := \frac{Y(p)}{X(p)} = \frac{b_N s^N + b_{N-1} p^{N-1} + \cdots + b_1 p + b_0}{a_N p^N + a_{N-1} p^{N-1} + \cdots + a_1 p + a_0}.$$

H est la *fonction de transfert* du système.
Ainsi, un système LTI est entièrement caractérisé par ses pôles et ses zéros que l'on note $(b_1, ..., b_N, a_1, ..., a_N)$ où N est l'ordre du système.
Si $R_j : \mathbf{I} \longmapsto \mathbf{O}$ est la fonction qui représente la réponse du système j où $1 \leq j \leq 5$, $\mathbf{I}$ est l'espace des inputs et $\mathbf{O}$ est celui des outputs. Alors on suppose qu'il existe une norme N sur $\mathbf{I}$, appelée *norme de similitude*, telle que sur toute boule de centre un des inputs au sens de la norme de similitude, R se comporte comme la réponse d'un système LTI.

Ainsi, pour chacun des 7 inputs, on calcule une fonction de transfert (en calculant ses pôles et ses zeros) qui permet d'approcher la réponse de nos 5 systèmes inconnus.
Pour un input I_i et un output $O_{i,j} =: R_j(I_i)$, on calcule les coefficients $(b_1^{i,j}, ..., a_N^{i,j})$ en minimisant l'erreur entre $O_{i,j}$ et la réponse d'un système LTI définie par ces coefficients. On utilise en pratique le module signal de scipy pour manipuler les fonctions de transferts et calculer les sorties des systèmes linéaires utilisés.

On vérifie en pratique que la réponse du système LTI défini par les coefficients $(b_1^{i,j}, ..., a_N^{i,j})$ approche bien $O_{i,j}$ mais approche mal certains des autres outputs $(O_{k,j})_{k\neq i}$ du système j.
On note $\hat{R}_{i,j}$ la fonction réponse système linéaire de coefficients $(b_1^{i,j}, ..., a_N^{i,j})$, $\hat{R}_{i,j}(I_i)$ est un estimateur de $O_{i,j}$

2 Approximation de la norme de similitude

Les résultats numériques précedent renforce l'hypothèse de l'existence d'une norme de similitude comme définie plus tôt. Pour approcher cette norme on associe à chacun des inputs de l'espace I des paramètres $P = (p_1, ..., p_r)^T$ qui caractérisent le signal :

- p_1 = fréquence des oscillations;
- p_2 = constante de Lipschitz;
- p_3 = nombre de changement de signes;
- p_4 = proportion en temps où le signal est nul;
- p_5 = aire.

On définit alors un estimateur de la norme de similitude entre un signal I_1 et un signal I_2 de paramètres $P_1 = \left(p_1^{(1)}, ..., p_r^{(1)}\right)^T$ et $P_2 = \left(p_1^{(2)}, ..., p_r^{(2)}\right)^T$ par

$$N_{sim}(I_1, I_2) = \beta \left(\|p_1^{(1)} - p_1^{(2)}\|, ..., \|p_r^{(1)} - p_r^{(2)}\|\right)^T,$$

où $\beta = (\beta_1, ..., \beta_r)$ est inconnu et à optimiser.

3 Prédiction d'un output

Soit j un système, on calcule un estimateur de $O_{i,j}$, noté $\hat{O}_{i,j}$, avec une moyenne pondérée par la valeur de la norme de similitude, plus un signal est proche de I_i du point de vue de N_{sim}, plus son apport dans le calcul de $O_{i,j}$ sera grand :

$$\hat{O}_{i,j} = \frac{\sum_{k \neq i} \alpha_k \hat{R}_{k,j}(I_i)}{\sum_{k \neq i} \alpha_k},$$

où $\alpha_k = 1/N_{sim}(I_k, I_i)^2$.

4 Optimisation de N_{sim}

On veut optimiser la valeur de β pour minimiser l'erreur quadratique moyenne prise sur tous les outputs prédit du dataset d'entrainement. On minimise alors la fonction :

$$E : \beta \longmapsto \sum_{i,j} RMSE(\hat{O}_{i,j}, O_{i,j}).$$

On obtient un β avec seulement deux coefficients non nuls, β_2 et β_3 qui correspondent aux paramètres :

- p_2 = constante de Lipschitz;
- p_3 = nombre de changement de signes.