

PROJET
D'ENSEIGNEMENT D'APPROFONDISSEMENT
SYSTEMES AVANCES

Titre : Algorithme d'interpolation de mesures expérimentales.
Optimisation de la mesure et traitement par deep-learning

Présenté par messieurs

Hugo Sarazin

et

Jean-François Viardot,

Etudiants en maîtrise des risques industriels

Sous la direction de Monsieur Antoine Ferreira,
Maître de conférences à l'INSA Centre Val de Loire

SOMMAIRE

INTRODUCTION	3
L'EXPERIENCE	4
LE CODE	8
CONCLUSION.....	9

INTRODUCTION

Les outils mathématiques qu'offrent Newton ou encore Lagrange au regard des algorithmes d'interpolation semblent aujourd'hui presque dépréciés. Face à la puissance des algorithmes d'intelligence artificielle de machine learning et de deep learning, les mathématiques standards ont des airs de lointains souvenirs.

Lagrange nous offre la possibilité de déterminer une fonction polynomiale passant par l'ensemble des points de mesures renseignées.

Le polynome a donc la forme :

$$P_n(x) = \sum_{k=0}^n l_k(x) f(x_k)$$

avec l_k un polynome de degré n .

Un outil semble-t-il fiable, bien qu'il demeure difficile de déterminer l_{k+1} . L'algorithme d'interpolation de Newton intervient alors, proposant une forme de polynome :

$$e_k(x) = \prod_{i=0}^{k-1} (x - x_i)$$

Avec $e_0 = 1$

Le polynome de Newton est alors :

$$P_n(x) = \sum_{k=0}^n \alpha_k e_k(x_0)$$

Il est donc nécessaire d'évaluer l'ensemble des $e_k(x_i)$.

On obtient un calcul de différences divisées d'ordre i proposant que :

$$f(x_0, \dots, x_i) = \alpha_i = \frac{f(x_1, \dots, x_i) - f(x_0, \dots, x_{i-1})}{x_i - x_0}$$

Cette méthode étant longue et, d'après nous, dépréciée, nous utiliserons des algorithmes d'intelligence artificielle que nous jugeons plus performant, car permettant de mettre en symbiose une plus grande quantité de données.

L'EXPERIENCE

L'expérience réalisée en laboratoire est la suivante :

Un ensemble de bobines de dimension connue est installée sur un banc d'essai.

Les bobines sont reliées à un circuit d'alimentation connu.

On mesure alors en un maximum de point de l'espace le champ magnétique induit.

On fait varier les paramètres d'entrée: position.

Puis, on réalise à nouveau une série de mesure.

L'ensemble des mesures effectuées dans les différentes conditions de position et de courant forme un modèle mathématique direct donnant la fonction d'entrée-sortie de notre système.

Le but est de pouvoir connaître les conditions paramétriques nécessaire à induire le champ et gradient de champ magnétique induit souhaités aux points de l'espace choisis.

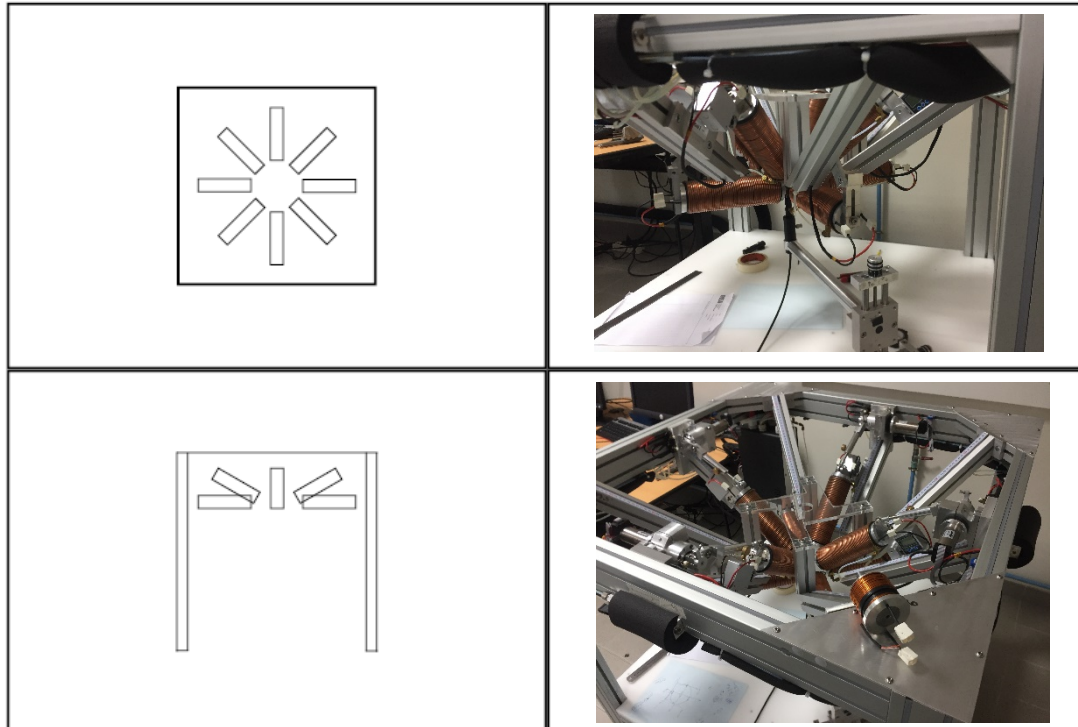


Figure 1 : schéma et photographie du dispositif expérimental

Nous avons jugé après entretien avec M. Ruipeng Chen que sa méthodologie de mesure expérimentale n'était pas optimisée : Nous avons remarqué que le contexte de sa mesure pouvait l'amener à échouer un certain nombre de mesures dans des proportions suffisantes pour fausser le modèle du système.

Remarque : A considérer que la position du capteur soit suffisamment précise dans le cadre de la méthode de mesure actuelle, on considère surtout que la mesure manuelle est laborieuse et fastidieuse si l'on souhaite obtenir un grand nombre de mesures.

Nous proposons d'établir un processus entièrement automatisé permettant de valider avec une plus grande précision la position et l'orientation du capteur.

Les bobines sur le banc d'essai sont déplaçables grâce à des moteurs que nous n'omettrons pas d'asservir à notre système. Il est nécessaire que l'automate ait l'ensemble du système sous son contrôle.

Nous avons donc réalisé un robot **SCARA (modélisation SW jointe)** permettant d'évaluer avec précision la position du capteur à effet hall. Il permet également de se déplacer plus rapidement qu'une personne chargée de faire manuellement les mesures. On pourra réaliser plus de mesures dans un temps moindre. Nous lui associons un poignet sphérique afin de maîtriser l'orientation du capteur.

On considèrera que l'intervalle de distance pour les mesures établies par M. Ruipeng Chen est suffisamment précis. Nous utiliserons le même : **tous les 4mm dans toutes les directions.**

Le capteur est un capteur à effet hall circonscrit dans un cube de 40mm de côté.

On placera le point central du poignet du robot au centre de ce cube.

On réduira l'orientation des différentes bobines à l'orientation du plateau qui les relie.

On établit pour notre robot S.C.A.R.A la position du poignet en (x,y,z) et les rotations des points A_1 et A_2 (respectivement θ_1 et θ_2) telles que :

$$\theta_1 = 2 \tan^{-1} \left(\frac{-a_1 y \pm \sqrt{a_1^2 (x^2 + y^2) - (a_1^2 - a_2^2 + x^2 + y^2)^2}}{a_1^2 - a_2^2 + x^2 + y^2 - 2a_1 x} \right)$$

$$\theta_2 = \cos^{-1} \left(\frac{x - a_1 \cos \theta_1}{a_2} \right) - \theta_1$$

$$q_3 = -z$$

On positionne le plateau à sa position la plus basse et horizontale.

Un capteur de position (laser) détecte lorsque l'axe vertical du robot est dans le coin préalablement choisi comme point pré-initial.

Une fois ce point calibré, l'axe se déplace afin d'atteindre le point central du banc. (déplacement de $\frac{\sqrt{2}}{2}$ selon x et $\frac{\sqrt{2}}{2}$ selon y. L'axe vertical monte jusqu'en butée. D'après la hauteur estimée du plateau (1 mètre), la distance entre le capteur et le plateau est de 10cm.

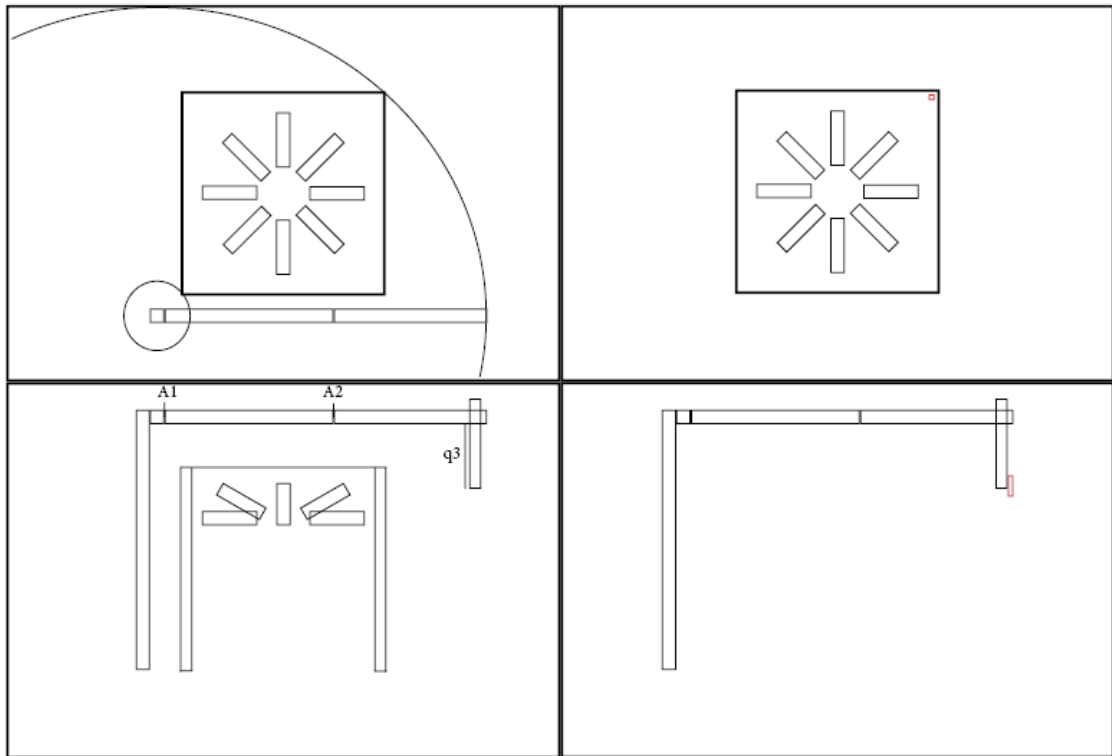


Figure 2 : schéma du robot SCARA, de son domaine et de son capteur d'initialisation

Le robot va se déplacer afin de collecter les données de champ magnétique dans tout l'espace. Les mesures seront effectuées successivement dans les 3 dimensions de l'espace par pas de 4mm.

Une fois l'ensemble des mesures effectuées, le robot revient à sa position initiale.

1ère étape : Le plateau se déplace de 5mm vers le haut. Le robot effectue une nouvelle série de mesures. Le plateau se déplace à nouveau jusqu'à effectuer 10 fois la série de mesures.

2ème étape : un côté du plateau s'abaisse de 5mm vers le bas. Le robot effectue une nouvelle série de mesures. Ce côté du plateau se déplace à nouveau jusqu'à effectuer 10 fois la série de mesures.

3ème étape : un côté adjacent au précédent s'abaisse de 5mm vers le bas. Le robot effectue une nouvelle série de mesures. Ce côté du plateau se déplace à nouveau jusqu'à effectuer 9 fois la série de mesures.

Chaque série de mesures se compose de : $9 \times 9 \times 6$ mesures soit 486 mesures.

Le plateau se déplace 29 fois et l'ordinateur effectue 30 séries de mesures soit : 14 580 mesures.

Pour finir, le système répètera l'expérience avec le capteur orienté dans les deux autres directions de l'espace à trois dimensions : 43 740 mesures seront donc effectuées.

Le Robot SCARA est capable de se déplacer à une vitesse de 7 m/s avec une précision de 0,01 mm . Selon le chemin qu'il doit parcourir, la vitesse n'est pas la limite et ne déterminera pas la vitesse d'échantillonnage. On prendra comme vitesse d'échantillonnage 1 mesure par seconde. Le robot se déplacera de 4mm sur l'intervalle de 0 à 0,5s, puis s'arrêtera. Le capteur réalisera chaque mesure sur l'intervalle de 0,5 à 1s. Cette méthode séquencée entre mesure et déplacement est nécessaire afin d'éviter de mesurer les perturbations du champ magnétique induit par le déplacement du capteur.

A cette vitesse, l'ensemble des mesures sera effectué en 12 heures et 9 minutes.

Les données sont entrées dans le module Python TensorFlow avec comme structure de donnée :

$43740 \times 8 : Z = [x \ y \ z \ r1 \ r2 \ t1 \ t2 \ V]$

$[x \ y \ z]$ représentent la position du capteur dans l'espace de dimension 3

$[r1 \ r2]$ représentent l'orientation du capteur dans l'espace de dimension 3

$[t1 \ t2]$ représentent la position/orientation des bobines

V est la donnée de mesure du capteur

Chaque paramètre est un float codé sur 4 bits. Les données sont injectées 100 fois chacune dans tensorflow afin d'avoir une plus grande précision.

La dataset fera donc environ 18Mo

LE CODE

```
1 import pandas as pd
2 import numpy as np
3
4 np.set_printoptions(precision=3, suppress=True)
5
6 import tensorflow as tf
7 from tensorflow.keras import layers
8 from tensorflow.keras.layers.experimental import preprocessing
9
10 #abalone_train = pd.read_csv("data.csv",names=["x", "y", "z", "r1", "r2","t1", "t2", "V"])
11
12
13 abalone_train.head()
14
15 abalone_features=abalone_train.copy()
16 abalone_labels = abalone_features.pop('V')
17
18 abalone_features=np.array(abalone_features)
19 abalone_features
20
21 abalone_model = tf.keras.Sequential([
22     layers.Dense(64),
23     layers.Dense(1)
24 ])
25
26 abalone_model.compile(loss = tf.losses.MeanSquaredError(),
27                       optimizer = tf.optimizers.Adam())
28
29
30 abalone_model.fit(abalone_features, abalone_labels, epochs=100)
31
32 normalize = preprocessing.Normalization()
33 normalize.adapt(abalone_features)
34
35 norm_abalone_model = tf.keras.Sequential([
36     normalize,
37     layers.Dense(64),
38     layers.Dense(1)
39 ])
40
41 norm_abalone_model.compile(loss = tf.losses.MeanSquaredError(),
42                            optimizer = tf.optimizers.Adam())
43
44 norm_abalone_model.fit(abalone_features, abalone_labels, epochs=100)
45
```

Figure 3 : code python TensorFlow

Les données sont ainsi stockées dans un fichier .csv, facilement importable dans Excel ou autre afin de les visualiser

Le Loss est calculé à partir des moyennes carrées.

CONCLUSION

Nous avons travaillé avec une approche quasi-industrielle dans le cadre de cette étude. Nous n'avons pas cherché à faire évoluer la technique sur le plan mathématique mais avons optimisé l'expérience afin d'obtenir un rendement de sortie.

En effet, plutôt que de s'acharner sur des algorithmes mathématiques déjà établis, nous avons décidé de nous servir des méthodes de Deep Learning pour établir un modèle convaincant à partir de données fournies. De cette manière, en établissant une intelligence artificielle basée sur l'apprentissage d'un réseau de neurones, on peut obtenir plus efficacement et sans calcul et/ou algorithme complexe le modèle souhaité.

Nous avons donc pu nous concentrer en plus sur l'optimisation de la mesure en mettant en place un système qui permet des relevés automatisés des valeurs de champs

La finalité est a posteriori de pouvoir donner à TensorFlow un ensemble de paramètres tels que la position de l'espace, le champs magnétique et l'orientation du champ magnétique souhaité. L'algorithme donnera à l'utilisateur (voire en commande directe sur le système) les valeurs de position des bobines afin d'obtenir la sortie souhaitée.