

TP 1 – UTILISER DIFFERENTS ALGORITHMES D'IA

ELEMENTS DE CORRIGE

CAS 1 : GENERER DU CODE POUR PROGRAMMER UNE CARTE ELECTRONIQUE

1.1 AFFICHER UN MESSAGE SUR L'ECRAN DU WIO TERMINAL

On peut par exemple utiliser le prompt suivant :

Peux-tu me générer un code pour un wio terminal programmé avec arduino IDE, pour afficher le texte suivant en blanc sur fond bleu sombre : "bonjour" et à la ligne suivante "bienvenue dans ce TP d'IA". Merci de centrer le texte sur l'écran et de tourner l'écran pour que le connecteur USB du Wio terminal soit vers le bas. Je voudrais aussi afficher un smiley jaune souriant en bas au centre de l'écran.



Un exemple de code obtenu est donné dans le fichier **1-1_Affichage_message_Wio.ino**.

A retenir dans ce prompt visant à générer du code :

- On spécifie bien quelle carte on souhaite programmer, et via quelle interface de programmation.
- On indique avec précision le détail des éléments que l'on souhaite afficher (couleurs, texte centré...) et bien sûr le contenu du texte.
- On n'hésite pas à procéder par itération : par exemple on teste une première fois sans indiquer l'orientation du Wio Terminal, on remarque que le texte est affiché à l'envers par rapport à ce qu'on souhaite, puis on envoie un nouveau prompt à la suite pour indiquer dans quel sens on souhaite tenir le terminal.

1.2 CONNECTER UN CAPTEUR ET AFFICHER LA MESURE SUR L'ECRAN

On peut par exemple utiliser le prompt suivant, rédigé en anglais car les LLM sont actuellement plus performants avec cette langue (la plupart des LLM actuels sont entraînés sur des bases de données en anglais. Lorsqu'on leur pose une question en français, ils commencent par la traduire en anglais) :



I want to program a Wio terminal using Arduino IDE. I wire a Seeed ultrasonic sensor to the right Grove connector of the wio terminal (USB connector turned down). The sensor has the following pins : GND, VCC, NC and SIG. I want to use the Arduino Library "Grove ultrasonic ranger" by Seeed which is called in the program by ""#include <Ultrasonic.h>". The readings are made with the function "ultrasonic.MeasureInCentimeters()". The measure will be displayed in meters in large white text at the center of the screen of the terminal. The background is dark blue.

Et on peut ajouter :

Add a small text at the top of the screen "distance mesurée" and a really small text at the bottom of the screen "programme créé par une IA".

Un exemple de code obtenu est donné dans le fichier **1-2_Mesures_capteur_ultrasons.ino**.

A retenir dans ce prompt visant à générer du code :

- On peut repartir de l'exemple précédent, dans lequel la mesure avec le capteur à ultrasons est déjà opérationnelle.
- On est ici confronté au grand nombre de modèles de capteurs à ultrasons légèrement différents existant sur le marché, et au grand nombre de bibliothèques associées. Il faut arriver à expliquer précisément au LLM quel modèle de capteur et quelle bibliothèque on veut utiliser.

1.3 AFFICHER LES MESURES AU COURS DU TEMPS SUR UN GRAPHIQUE

On peut par exemple utiliser le prompt suivant :

Can you change the previous code to display a chart showing the evolution of the measured distance over time ? Please use the Seeed_Arduino_Linechart library, starting from the complete code given at the end of the webpage <https://wiki.seeedstudio.com/Wio-Terminal-LCD-Linecharts/>. The measurements are plotted in meters, and the time range of the chart is 30s. A measurement value is added every second. Please add a small title at the top of the screen "Mesure capteur a ultrasons" and display the value of the measurement in meters in small characters at the bottom of the screen...

Un exemple de code obtenu est donné dans le fichier **1-3_Graphique_mesures.ino**.

A retenir dans ce prompt visant à générer du code :

- On impose l'utilisation de la bibliothèque Seeed_Arduino_Linechart qui n'est pas très largement utilisée sur internet, donc mal connue des LLM. Une solution pour l'utiliser consiste à donner au LLM l'exemple de code disponible au bas de la page de présentation de la bibliothèque, et de lui demander de l'adapter pour la mesure avec le capteur à ultrasons. On donne donc au LLM un exemple à modifier pour l'adapter à la demande formulée dans le prompt.
- Il faut bien renseigner les différentes caractéristiques du graphique que l'on souhaite obtenir (unités, intervalle temporel à tracer, période d'échantillonnage). Sans ces informations, le LLM pose des valeurs arbitraires qui ne correspondent pas forcément à ce qu'on souhaite.

Q1. *Que retenez-vous de la manière de communiquer des prompts à l'algorithmes d'IA introduite dans cette partie du TP ? Avez-vous rencontré des points bloquants ? A quoi sont-ils dus ?*

On note dans cette partie du TP que même en l'absence de toute connaissance en codage, il est possible grâce à un LLM de programmer une carte électronique pour un usage bien défini.

Au-delà des lignes directrices données dans l'Annexe 2 « Prompt Engineering », on remarque que la démarche est très souvent itérative pour parvenir aux résultats souhaités. En général, les itérations sont induites par un manque de précisions dans la demande initiale. Imaginer les informations relatives au choix qu'il serait nécessaire d'effectuer si on programmait la carte de manière usuelle peut aider à communiquer le maximum d'informations au LLM.

Les difficultés rencontrées sont principalement dues à la variété d'architectures matérielles (ici de capteurs à ultrasons) relativement similaires existant sur le marché. Le LLM, entraîné à partir d'un vaste ensemble de données sur les différents modèles, peut regrouper des lignes de code provenant de différents exemples présents sur internet, constituant ainsi un code qui ne fonctionne pas du tout ou qui ne fonctionne pas avec le type de capteur utilisé. Il peut alors être utile de décrire précisément l'architecture matérielle et/ou d'imposer l'utilisation d'une bibliothèque.

Pour finir, utiliser une bibliothèque peu connue nécessite parfois de communiquer directement un exemple de code au LLM. Il lui reste alors à le modifier, mais on parvient plus rapidement à un résultat pertinent en lui communiquant l'exemple.

CAS 2 : RESOUDRE UN PROBLEME DE REGRESSION AVEC EXCEL

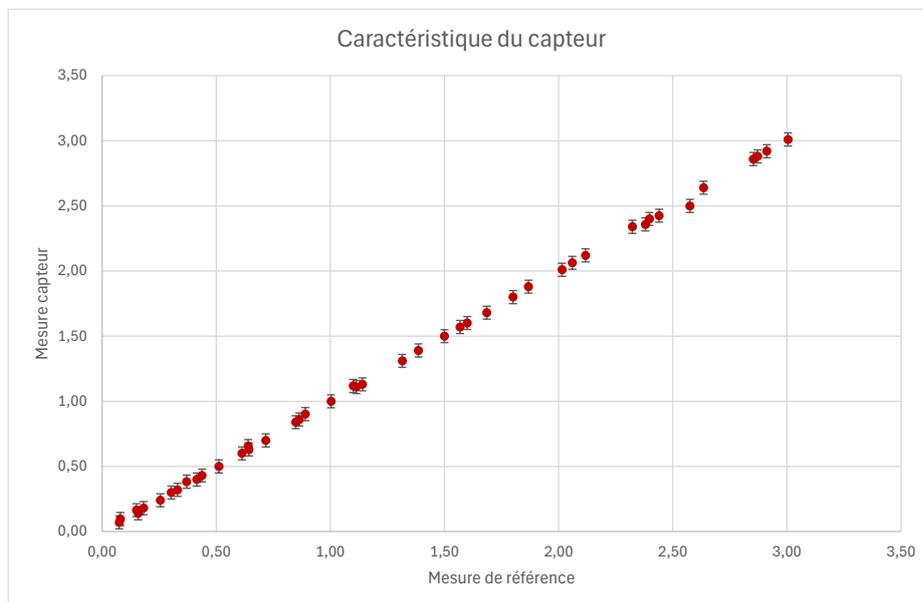
2.1 ACQUERIR DES MESURES POUR LA CARACTERISATION DU CAPTEUR.

Nous allons ici utiliser le capteur à ultrasons et l'interface de mesure associée créé dans la partie 1.2.



Protocole expérimental proposé :

- On place le Wio terminal ainsi que le capteur à ultrasons sur un support stable, fixe, et avec un champ dégagé. On y téléverse le code créé dans la partie 1.2 pour permettre une lecture aisée des valeurs de distance mesurées par le capteur.
- On utilise un objet plan, suffisamment grand pour être détecté par le capteur à ultrasons – un carton par exemple. L'objet plan devra être perpendiculaire à la ligne de visée du capteur.
- On utilise un mètre ruban comme mesure de référence. La précision de mesure avec le mètre est à évaluer, mais elle est normalement de l'ordre de quelques millimètres.
- On effectue ensuite pour différentes positions de l'objet les mesures de distance avec le mètre ruban et avec le capteur à ultrasons + Wio terminal, et on compile les résultats dans le tableau proposé « Caractérisation capteur ultrasons.xlsx ». Dans l'idéal, il faudrait une vingtaine de mesures pour caractériser correctement le capteur.
- On obtient la courbe de caractérisation du capteur $\text{Mesure capteur} = f(\text{Mesure de référence})$.
- On complète les autres informations de la fiche de caractérisation capteur afin d'obtenir un rapport complet de l'essai.



Les résultats (valeurs de paramètres) qui suivent dépendent bien évidemment des mesures effectuées.

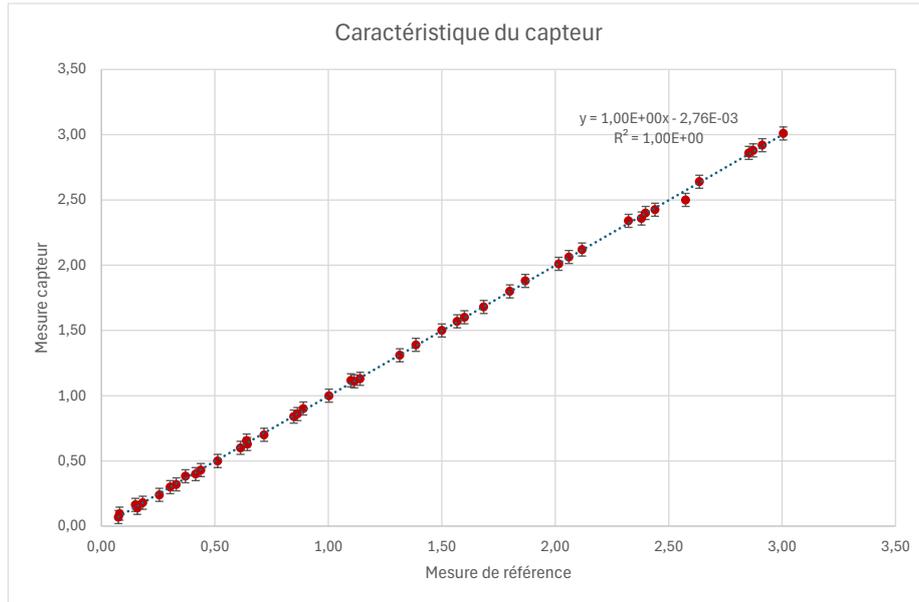
2.2 EFFECTUER UNE REGRESSION LINEAIRE AVEC UNE FONCTION INTEGREE A EXCEL

Avec la fonction de régression linéaire intégrée à Excel, on obtient la courbe modèle donnée sur la page suivante. On obtient l'équation $y = 1,00 \times x - 2,76.10^{-3}$ avec un coefficient $R^2 = 1,00$.

Le coefficient $R^2 = 1,00$ montre une excellente corrélation entre le modèle et l'expérience.

L'équation de la fonction modèle montre un coefficient directeur égal à 1,00, qui montre un bon étalonnage du capteur pour sa partie gain. En effet, cette courbe devrait dans l'idéal avoir une équation $y = x$.

L'ordonnée à l'origine égale à $-2,76 \cdot 10^{-3} m$ montre un offset de 2,76mm au niveau de la mesure. Il peut être dû à la référence qui a été prise pour la mesure de position avec le mètre, qui n'est pas la même que celle utilisée par le capteur à ultrasons. Grâce à l'obtention de cette courbe d'étalonnage, on peut corriger cet offset.



2.3 PROGRAMMER UNE REGRESSION LINEAIRE AVEC EXCEL

En suivant les consignes du sujet, on construit la colonne F permettant de calculer les carrés des écarts modèle / expérience ainsi que leur somme. Ci-dessous et dans le corrigé, ce calcul est effectué en 2 étapes (colonnes E et F) :

- Calcul de la valeur prédite par le modèle : $y = a \times x + b$
- Calcul du carré des écarts modèle / expérience
- Calcul de la somme des carrés des écarts

Tableau et courbe de caractérisation :

	B	C	E	F
	Mesure de référence	Mesure capteur	Régression linéaire manuelle	(Ecart mod/exp) ²
	m	m		
25	0,08	0,07	=a*B25+b	0,0000
	0,08	0,10	0,079886015	0,0003
	0,15	0,16	0,149786278	0,0002
	0,16	0,14	0,15777488	0,0003
	0,18	0,18	0,181740684	0,0000
		
			Somme écarts² :	0,0102

a et b sont des cellules nommées (voir sujet)
 $= (E25 - C25)^2$

En utilisant le solveur de Excel comme proposé dans le sujet, on parvient à minimiser la somme des carrés des écarts. Le solveur nous permet d'obtenir les valeurs suivantes des paramètres $\{a, b\}$ optimaux :

Régression linéaire manuelle (sans la fonction courbe de tendance) :
 équation $y = ax + b$

a	1,00E+00 sans unité
b	-2,76E-03 m

Voici la configuration de la fenêtre solveur à utiliser pour obtenir les valeurs précédentes avec les données expérimentales du corrigé :

Objectif à définir :

À : Max Min Valeur :

Cellules variables :

Contraintes :

Rendre les variables sans contrainte non négatives

Penser à décocher « Rendre les variables sans contraintes non négatives » car rien ne contraint les valeurs des paramètres de cette régression. On note que pour des problèmes plus complexes, on pourrait ajouter des contraintes (dépendant éventuellement d'autres cellules) au solveur.

Q2. Commenter les deux démarches de régression mises en œuvre dans cette partie du TP. Observez-vous des différences entre les résultats obtenus ? Quelles autres possibilités offre la seconde méthode de régression utilisée ?

On remarque que les deux démarches de régression linéaire ont visiblement mené à des coefficients strictement identiques. En pratique, et si on extrayait ces coefficients avec un grand nombre de décimales, il existerait une différence très légère entre les valeurs obtenues, dépendant de la tolérance imposée à chacun des algorithmes d'identification (modifiable dans les paramètres avancés du solveur). Ici, et au vu de la faible dispersion / complexité des données expérimentales, on converge nécessairement vers la même solution mais ce ne sera pas forcément le cas selon le problème d'identification posé (il peut exister plusieurs minimums locaux de la fonction coût).

La seconde méthode permet, en plus de la première :

- De ne pas être limité par les fonctions disponibles dans l'option « courbe de tendance » de Excel. On peut alors utiliser un grand nombre de modèles mathématiques, avec éventuellement de nombreux paramètres.
- De choisir l'algorithme d'identification, ainsi que de jouer sur ses paramètres (options de convergence / tolérance, option multistart pour éviter les minimums locaux de la fonction coût...)
- D'ajouter des contraintes supplémentaires aux paramètres intervenant dans le problème de régression.

Cette seconde implémentation d'un problème de régression est plus longue à mettre en œuvre. Elle offre cependant des possibilités plus larges.

CAS 3 : GENERER DES MACROS EXCEL VIA DU CODE VBA



3.1 AUTOMATISER LA GENERATION D'UN RAPPORT

Objectif : Générer un rapport au format pdf, obtenu à partir d'une certaine plage de données du tableur élaboré précédemment. La macro pourra dialoguer avec l'utilisateur (imaginer des contraintes que vous pourrez ajouter dans le prompt). On intégrera ensuite un bouton dans Excel, de manière à pouvoir exécuter directement la tâche.

On peut par exemple utiliser le prompt suivant pour obtenir la macro :

J'ai un document Excel avec du contenu (données et courbe) dans la plage A1:L46. J'aimerais une macro VBA pour générer à partir de cette sélection un document au format pdf en conservant la mise en forme et en mettant l'ensemble au format d'une page A4 recto. La macro devra demander à l'utilisateur à quel emplacement et sous quel nom enregistrer le fichier, puis afficher un message confirmant l'enregistrement une fois terminé. Peux-tu me générer cette macro ?

La macro obtenue est donnée dans le fichier « Macro 3-1.txt ».

Pour implémenter cette macro, on peut si on ne sait pas le faire utiliser un LLM. Exemple de prompt :

Comment implémenter cette macro dans ma feuille de calcul Excel ?

Pour associer un bouton à la macro, on peut encore utiliser un LLM. Exemple de prompt :

Comment créer et associer un bouton cliquable à cette macro ?

On parvient très facilement à automatiser des tâches dans Excel (ou autre) en rédigeant des macros / scripts. On garde ici à l'esprit que les LLM fonctionnent très bien pour la génération de code informatique, comme vu en partie 1.

3.2 ARCHIVER AUTOMATIQUEMENT UN RAPPORT

Objectif : Archiver automatiquement le rapport d'essai précédent au format pdf dans le dossier "Rapports_essai" sur le bureau de l'ordinateur. Si le dossier n'est pas créé, le créer. Formater le nom du fichier comme suit : "date – personne en charge de l'essai – Marque – Référence capteur".

On peut par exemple utiliser le prompt suivant pour obtenir la macro :

Créer une macro pour effectuer les tâches suivantes : Créer un rapport en pdf, réduit à une seule page A4 portrait, et correspondant à la plage de données A1:L46 du fichier excel. Archiver ensuite automatiquement ce rapport dans un dossier "Rapports_essai" qui sera créé automatiquement si besoin sur le bureau de l'ordinateur. Le nom du fichier doit être formaté comme suite : "date - contenu cellule C5 - contenu cellule C9 - contenu cellule C10".

La macro obtenue est donnée dans le fichier « Macro 3-2.txt ».

- Q3. *Avez-vous noté des points particuliers dans la manière de communiquer des prompts à l'algorithme d'IA dans cette partie. Imaginez un cas d'utilisation de macros similaires dans une entreprise (les possibilités sont bien plus vastes que celles abordées dans ce TP : envoi de courriels, manipulation d'images, interactions avec Word ou d'autres logiciels...).*

On note qu'il faut toujours communiquer des informations précises et complètes au LLM pour obtenir des réponses pertinentes. Si le résultat n'est pas pertinent ou renvoie des erreurs, on peut procéder par itérations. Encore une fois, le LLM permet ici un net gain de temps.



Autres exemples d'utilisation des macros (générées avec un LLM ou non) :

- Génération d'un trombinoscope à partir des images contenues dans un dossier et noms contenus dans un tableau.
- Export automatique de données depuis un tableur vers un document Word existant (préformaté) pour générer un rapport (qui sera plus propre que celui obtenu dans cette partie du TP)
- Publipostage : génération de rapports personnalisés (+ envoi automatique par email)
- Interaction avec d'autres applications via l'utilisation d'APIs (possibilités infinies)

CAS 4 : POUR ALLER PLUS LOIN – GENERER UNE PAGE WEB

Dans cette partie optionnelle, on propose d'explorer les autres possibilités des IA génératives.

Pas de correction pour cette partie ouverte du TP.