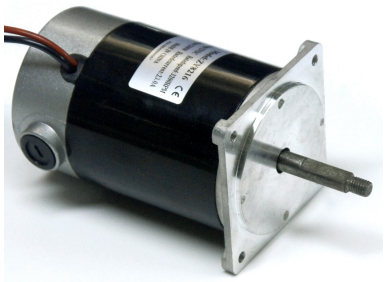
Séquence 01 - TP01 - Îlot 01

Lycée Dorian
Renaud Costadoat
Françoise Puig

Modélisation linéaire



Référence	S01 - TP01 - I01
Compétences	D2-06: Justifier le choix d'un capteur ou d'un appareil de mesure vis-à-vis de la grandeur physique à mesur D3-04: Identifier les erreurs de mesure. D3-05: Identifier les erreurs de méthode.
Description	Déterminer des caractéristiques par la mesure physique
Système	Moteur à courant continu

Objectif du TP:

Modéliser les caractéristiques d'un moteur à courant continu.

1 Modèle du moteur électrique à courant continu

$$u_m(t) = L_m.\frac{di(t)}{dt} + R_m.i(t) + e(t) \quad (1)$$

$$e(t) = K_e.\omega_m(t) \quad (2)$$

$$J.\frac{d\omega_m(t)}{dt} = C_m(t) - C_r(t) \quad (3)$$

$$C_m(t) = K_m.i(t) \quad (4)$$

Données :
- $u_m(t)$: tension aux bornes du moteur (V),
- $i(t)$: intensité du courant dans le moteur (A),
- $e(t)$: force électromotrice (V),
- $\omega_m(t)$: vitesse de rotation du moteur (rad.s^{-1}),
- $C_m(t)$: couple moteur (N.m),
- $C_r(t)$: couple résistant (N.m),
- L_m : inductance de la bobine du moteur (H),
- R_m : résistance électrique interne au moteur (Ω),
- K_e : constante électrique du moteur (V.rad^{-1}.s),
- J : inertie du moteur (kg.m^2),
- K_m : constante de couple du moteur (N.m.A^{-1}).

En général, on suppose $K_e = K_m$ pour une MCC.

Question 1 : D'après les équations (1) à (4), **écrire** une équation liant uniquement $u_m(t)$, $\omega_m(t)$ et $C_m(t)$.

Question 2 : Quelles sont les **hypothèses** à poser afin de mettre cette équation sous la forme $u_m(t) = K.\omega_m(t)$. Est-ce que cela vous paraît raisonnable de prendre ces hypothèses ?

Question 3 : En supposant que l'on arrive à mesurer le courant qui traverse le moteur, que devient-il alors possible d'**estimer** ?

2 Mesure des valeurs caractéristiques du moteur

Question 4 : **Déterminer** un protocole afin de mesurer la tension aux bornes du moteur. Donner la liste du matériel utilisé et ses caractéristiques (sensibilité, plage de mesure...).

Question 5 : **Déterminer** un protocole afin de mesurer la vitesse de rotation du moteur à l'aide du tachymètre. Donner les caractéristiques du matériel (sensibilité, plage de mesure...).

Question 6 : **Mettre en œuvre** ce protocole pour des tensions allant de 0V à 12V. Écrire les valeurs mesurés dans le script python à télécharger ici et l'utiliser afin de tracer la courbe $\omega_m(t) = f(u_m(t))$. Conclure quant à l'allure de ce tracé, quel modèle peut-on appliquer à ce phénomène ?

Question 7 : **Refaire** la mesure précédente en mesurant aussi le courant qui traverse le moteur.

Question 8 : **Proposer** et **mettre en œuvre** un protocole permettant de mesurer la résistance interne du moteur. Donner la liste du matériel utilisé et ses caractéristiques (sensibilité, plage de mesure...).

3 Vérification des modèles et analyse des résultats

Question 9 : À partir des résultats précédents, **déterminer** le couple résistant $C_r(t)$ pour le moteur libre. **Proposer** une solution permettant d'augmenter ce couple, **prédire** le comportement du système.

3.1 Vérifier expérimentalement un modèle théorique

Question 10 : **Mettre en œuvre** ce nouveau protocole et **conclure** quant à la validité de la prédiction de la question 9.

3.2 Déterminer la valeur de L_m

Question 11 : **Proposer** sans les mettre en œuvre des protocoles permettant de déterminer la valeur de L_m.

4 Synthèse du TP

Question 12 : **Conclure** quant au modèle obtenu pour ce moteur à courant continu, **réaliser** une synthèse de ce TP présentant votre démarche pour répondre à la problématique.

5 Annexes

5.1 Régression linéaire

A partir d'un ensemble de couples de valeurs (x_i, y_i), le but de la régression linéaire et la prédiction d'un modèle linéaire :

$$\hat{y} = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + ... + \theta_n \cdot x_n$$
$$\hat{y} = X\Theta$$

Ainsi, il faut trouver l'ensemble des θ_i qui minimisent l'écart (RMSE Root Mean Square Error ou critère des moindres carrés) entre $y = [y_0, y_1, ..., y_n]$ et $\hat{y}$. Pour la suite, on utilisera la MSE car le min de la RMSE est aussi celui de la MSE.

Il existe alors 2 solutions l'équation normale et la descente de gradient.

Équation normale

Il existe une solution analytique pour trouver la valeur de Θ qui minimise l'écart.

$$\hat{\Theta} = (X^T X)^{-1} X^Y y$$

Fonction python

```
1 def regression_lineaire_eq_normale(x,y):
2     X_b = np.array([[xi] for xi in x])
3     theta_best = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)
4     return theta_best
```

Descente de gradient

La descente de gradient est un algorithme d'optimisation très général, capable de trouver des solutions optimales à un grand nombre de problèmes.

L'idée générale de la descente de gradient est de corriger petit à petit les paramètres dans le but de minimiser une fonction de coût.

L'algorithme suit la pente de la descente de la fonction MSE vers le bas pour la minimiser.

Dans le cas où il y aurait plusieurs variables il faudrait les normaliser afin que l'impact de leur variation soit identique, ce n'est pas le cas pour notre étude.

La descente de gradient consiste à calculer la dérivée partielle (cf cours de math)

$$\frac{\partial}{\partial \theta_j} MSE(\theta) = \frac{2}{m} \sum_{i=1}^{m} (\Theta^T x^{(i)} - y^{(i)}) x_j^{(i)}$$

On prendra en entrée :

— η : le taux d'apprentissage qui correspond à la taille des pas que l'on effectue dans la descente (plus η est grand, plus vite on converge, mais cela peut générer de l'instabilité),

— le nombre d'itérations

Fonction python

```
def regression_lineaire_descente_gradient(x,y):
    X_b = np.c_[np.ones((len(x), 1)), x]
    eta = 0.1 # taux d'apprentissage (vitesse à laquelle on suit la pente)
    n_iterations = 1000 # tester plusieurs valeurs
    m = 2 # nombre de paramètres
    theta = np.random.randn(2,1)    # initialisation aléatoire de la solution
    for iteration in range(n_iterations):
        gradients = 2/m * X_b.T.dot(X_b.dot(theta) - y)
        theta = theta - eta * gradients
    return theta
```