



Master IESE_ S3

Projet : Projet : Estimation de l'État de Charge (SOC) d'une Batterie Li-ion

Module: Electric Traction Motors and control systems

Pr. Yahia Mazzi

yahia.mazzi@usmba.ac.ma

Objectifs :

Dans le cadre du module *Electric Traction Motors and control systems*, vous réaliserez un projet pratique d'estimation du SOC d'une batterie Li-ion à partir de données expérimentales.

Objectifs du projet :

- Implémenter un algorithme d'estimation avancé (**EKF**).
- Développer et entraîner des **réseaux de neurones** appliqués à un problème réel.
- Évaluer la précision d'un algorithme par des métriques objectives.
- Interpréter les performances d'une méthode pour usage dans un BMS.
- Comparer deux approches (**modèle vs data-driven**) de manière critique.

Les deux approches d'estimation qu'il faut étudier dans ce projet :

1. **Approche filtrage avancé** : Filtre de Kalman étendu (EKF)
2. **Approche intelligence artificielle** : Réseaux de neurones (LSTM vs Fully Connected)

Chaque étudiant ou groupe (**3 max**) devra développer les deux méthodes, les comparer et présenter une analyse critique des performances.

I. Projet 1 : Estimation du SOC par Filtre de Kalman Étendu (EKF)

Implémenter un EKF basé sur un modèle équivalent de la batterie (Thevenin de premier ordre) pour estimer le SOC en temps réel.

- Construire le modèle d'état de la batterie (dynamique SOC + tension terminale).
 - Formuler les équations non linéaires.
 - Implémenter les étapes du filtre **EKF** :
 - Prédiction
 - Linéarisation (jacobbiennes)
 - Mise à jour
 - Ajuster les matrices Q et R.
 - Tracer dans la même figure :
 - SOC réel (ou de référence)
 - SOC estimé par EKF
 - Tracer l'erreur d'estimation.
 - Discussion et interprétation des résultats obtenus.
 - Pour faciliter votre travail, vous trouverez ci-dessous le modèle discret de la batterie ainsi que les paramètres associés.
 - Le fichier Simulink à compléter est également fourni en pièce jointe.
 - Le profil de courant utilisé comporte deux séquences : une phase de décharge et une phase de charge.

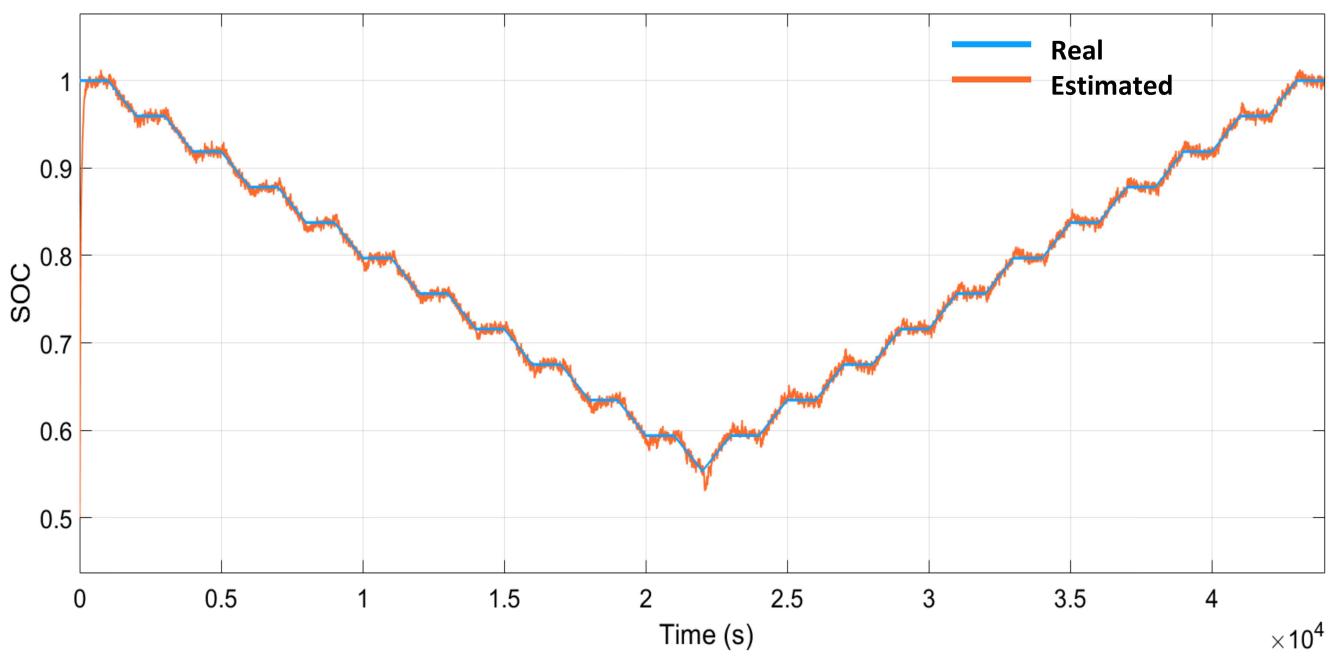
$$\begin{cases} x_{k+1} = \begin{bmatrix} \text{SOC}_{k+1} \\ V_{t_{k+1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\frac{T}{R_p C_p}} \end{bmatrix} x_k + \begin{bmatrix} -\frac{T}{3600 C_n} \\ R_p \left(1 - e^{-\frac{T}{R_p C_p}}\right) \end{bmatrix} I_{b_k} + v_k \\ y_k = h(x_k, I_{b_k}) = V_{oc}(\text{SOC}) - V_{p_k} - R_0 I_{b_k} + w_k \end{cases}$$

Avec: R0 = 0.24; R1 = 0.1; C1 = 920; Cn = 2.6 ;

$$V_{oc}(Z) = 10.52Z^7 - 30.1Z^6 + 28.38Z^5 - 9.727Z^4 + 3.512Z^3 - 3.697Z^2 + 1.983Z + 3.439$$

Z Représente SOC (pour simplifier l'écriture de l'équation)

Exemple du résultat attendu



Sorties à analyser

- Convergence de l'EKF vers le SOC initial.
 - Robustesse lors des transitions de courant (charge/décharge).
 - Erreur en régime permanent (steady-state error).

II. Projet 2 : Estimation du SOC par Intelligence Artificielle

Comparer deux architectures de réseaux de neurones pour estimer le SOC :

- ▶ **LSTM** (réseaux séquentiels pour signaux temporels)
 - ▶ **Fully Connected (MLP)** (réseau classique)

Pour le développement, l'entraînement, la validation et le test des modèles d'intelligence artificielle (LSTM et Fully Connected), vous pouvez utiliser **Google Colab**, qui offre un environnement Python complet avec GPU gratuit, ainsi que toutes les bibliothèques nécessaires (**TensorFlow**, **Keras**, **NumPy**, **Matplotlib**...).

Travail attendu

- ☞ Prétraitement des données (normalisation, segmentation en séquences pour LSTM).
 - ☞ Implémentation d'un **LSTM** et d'un **MLP**.
 - ☞ Entraînement, validation et test des modèles.



☞ Tracer :

- SOC réel vs SOC estimé (pour les deux réseaux).
- Erreur d'estimation dans le temps.
- Calculer les métriques suivantes pour chaque méthode : **MAE, RMSE, R²**.

☞ Puis comparer :

- La précision globale
- La stabilité dans les zones dynamiques (changement de courant)
- L'erreur en régime permanent

Rapport à rendre

Votre rapport doit comprendre :

1. **Introduction** : rappel du rôle d'un BMS et importance du SOC.
2. **Méthodologie** :
 - Modèle et équations EKF
 - Architecture des réseaux IA utilisés
3. **Résultats expérimentaux** :
 - Figures SOC réel vs estimé
 - Courbes d'erreur
 - MAE, RMSE, R²
4. **Analyse et interprétation** :
 - Convergence de l'algorithme
 - Zones de transition de courant
 - Écarts en régime permanent
 - Discussion des limites
5. **Conclusion** : comparaison finale EKF vs IA.

Remarques :

- ⇒ Un seul document est à rendre, format **.zip**, contenant le rapport et les fichiers de simulation & code.
- ⇒ **Date limite : 27 Décembre**