

# Progetto di Modellazione del SOC di Batterie Li-ion

Giuseppe De Nicolao, Marco Capelletti  
Identificazione dei modelli e analisi dei dati

April 9, 2025

## 1 Introduzione

L'obiettivo di questo progetto è **identificare un modello di State of Charge (SOC) di una batteria agli ioni di litio** sulla base di misure di tensione e temperatura della batteria. Il SOC è un parametro cruciale poiché indica quanta carica residua è disponibile, influenzando direttamente l'autonomia e le prestazioni di un veicolo elettrico. La **sfida ingegneristica** consiste nel descrivere la relazione fra:

- *SOC* (in  $[0, 1]$ )
- *Tensione* misurata
- *Temperatura* operativa

tramite un modello affidabile e in grado di funzionare bene per **diverse** temperature operative.

## Dati a disposizione

Nel progetto, avrete due file:

1. `train_data.mat`
2. `val_data.mat`

Il primo contiene i dati (*SOC*, *Tensione*, *Temperatura*) utilizzati per *identificare* il modello e applicare criteri di validazione (AIC, FPE, MDL o test F) visti a lezione. Il secondo file serve per utilizzare la tecnica di *cross-validazione* su un set di dati diverso da quello di identificazione. In particolare, questo set contiene misure di *SOC* e *Tensione* a **Temperature intorno ai 10 °C**. Verrà inoltre utilizzato un set di dati di *test* con dati di temperatura intorno agli **0 °C (non a vostra disposizione)** per valutare in modo oggettivo le vostre predizioni finali.

## 2 Fasi del progetto

### 2.1 Punto 1: Caricamento e visualizzazione dei dati

1. Caricare i dati in *MATLAB* (`train_data.mat`, `val_data.mat`).
2. Visualizzare inizialmente i dati con *scatter plot* 2D o 3D (ad es. `scatter`, `scatter3`) per interpretare le relazioni fra SOC, Tensione e Temperatura.

## 2.2 Punto 2: Identificazione di un modello expit-polinomiale (SOC vs Tensione)

In questa fase, consideriamo inizialmente solo la **Tensione** come regressore:

1. **Filtrare** i dati di **training e validazione** con  $\epsilon$  ( $\epsilon = 1e-4$ ), mantenendo solo le osservazioni con  $\epsilon < \text{SOC} < 1 - \epsilon$ .
2. Applicare la trasformazione logit (o *logistica inversa*):

$$\text{logit}(x) = \ln \left( \frac{x}{1-x} \right).$$

L'identificazione (cioè la regressione polinomiale) avverrà nello *spazio logit*.

3. Applicare i **criteri oggettivi** visti a lezione (AIC, FPE, MDL) e **criteri soggettivi** (test F) e/o la **la crossvalidazione** per determinare il *miglior modello polinomiale*.
4. **Visualizzazione del modello**: **per visualizzare il modello, è necessario** ritornare nello *spazio originario* (SOC in  $[0, 1]$ ) tramite la funzione inversa della logit (chiamata funzione **expit**). Visualizzare il modello identificato, sovrapponendo i dati di *training* e le previsioni finali in un grafico (**scatter 2D** e **plot** della curva stimata).
5. Graficare anche il modello polinomiale nello spazio logit con i rispettivi dati e fare un confronto visivo con la figura costruita nel punto precedente.

## 2.3 Punto 3: Estensione con la Temperatura come ulteriore regressore

1. Ripetere la procedura del punto precedente includendo la variabile **temperatura** come ulteriore regressore.
2. Identificare e validare la migliore *superficie logit-polinomiale* che descriva la relazione tra SOC, tensione e temperatura.
3. Graficare la **superficie** stimata in 3D (**mesh** o **surf**) e confrontarla con i dati nelle variabili originali (**scatter3**).  
[serve a plottare superficie in 3D](#)
4. Graficare anche il modello polinomiale nello spazio logit con i rispettivi dati e fare un confronto visivo con la figura costruita nel punto precedente.

## 2.4 Punto 4: Competizione e Modello Finale

1. Ogni gruppo dovrà cercare di ottenere il miglior modello predittivo del SOC date tensione e temperatura; non ci sono limitazioni alle tecniche che potete applicare.
2. Una volta trovato il modello finale, dovrete creare una funzione Matlab che, in ingresso, riceva i vettori di tensione e temperatura e restituisca le previsioni di SOC:

```

function soc_pred = mySOCmodel(voltage_vec, temp_vec)
    % mySOCmodel: restituisce le previsioni di SOC
    % date in ingresso le misure di tensione e temperatura.
    % (La vostra implementazione)

end

```

3. I modelli verranno testati su un dataset di test (a nostra disposizione) e valutati in base all'RMSE nello spazio logit. Come unico indizio, è possibile fare riferimento all'RMSE del modello benchmark logit-polinomiale ottenuto con la procedura della Parte 2 (in cui l'unico iperparametro validato è il grado massimo del polinomio) che è pari a: **0.3201**.

### 3 Valutazione

Il progetto sarà valutato con un punteggio massimo di 3 punti in base a:

- **Presentazione Finale:** completezza, chiarezza e correttezza nell'esposizione del lavoro svolto.
- **Performance del Modello:** l'RMSE ottenuto sul dataset di test, in confronto al modello benchmark.