User Guide and Code Explanation for the LSV Overpotential Analysis Python File

Levi YU

February 5, 2025

This Python file is designed to provide a user-friendly interface for LSV overpotential analysis by applying an advanced Tafel fitting method that incorporates statistical error evaluation. The two sections above describe how to use the application and explain the structure and theoretical background of the code.

1 How to Use This File

1. **Installation and Launch:**
   Ensure that Python 3 is installed on your system along with the required packages: `Tkinter`, `pandas`, `numpy`, `matplotlib`, `scipy`. Then run the Python file to open the GUI.

2. **Loading Your Data:**
   Click the `Browse` button to select an Excel file containing your LSV data. The file should have at least two columns:
   - Column 1: Voltage (V)
   - Column 2: Current density (A/cm^2)

3. **Entering Parameters:**
   In the GUI, enter:
   - **Temperature (K)**: This is used to compute the reversible potential.
   - R_{CL} **(Ohm·cm^2)**: The catalyst layer resistance.
   - **HFR (Ohm·cm^2)**: The high-frequency (ohmic) resistance.

4. **Running the Analysis:**
   Click the `Fit & Plot` button. The program will:
   - Remove the ohmic drop from the measured voltage.
   - Compute a corrected overpotential (η_{corr}) as
   $$\eta_{\text{corr}} = V_{\text{corr}} - E_{rev}, \quad \text{where } V_{\text{corr}} = V - i \times \text{HFR}.$$
   - Perform a Tafel analysis on η_{corr} versus $\log_{10}(i)$ in the current density range 0.005–0.1 A/cm^2 using an advanced method:
     - It randomly selects 10 candidate 60 mV windows within the available voltage range.

- For each candidate, three subwindows (20 mV, 40 mV, and 60 mV) are defined and a linear regression is performed to obtain the Tafel slope (b_{kin}) and intercept.
- The exchange current density is computed as

$$i_0 = 10^{-(\text{intercept}/b_{kin})}.$$

- The relative standard errors (SE%) for both b_{kin} and i_0 are calculated and combined to select the candidate with the smallest combined metric.

5. **Exporting Results:**
   Use the `Export Data` button to save the calculated overpotential components and related parameters to an Excel file.

2 Explanation of the Code

Overview

This Python file is designed for the analysis of LSV data by separating the measured voltage into different overpotential components. It estimates the kinetic parameters using a statistical Tafel fitting method that involves:

- Removing the ohmic drop.
- Performing Tafel analysis on a corrected overpotential.
- Evaluating multiple candidate voltage windows (10 candidates) by subdividing each into three subwindows.
- Selecting the best candidate based on the combined relative standard error (SE%) of the Tafel slope and the exchange current density.

Usage of Helper Functions

`read_data(filename)`: Reads the Excel file containing LSV data and returns the voltage and current arrays. It ensures that only valid (non-null) data is processed.

`fit_tafel(eta_corr, i)`: Implements the advanced Tafel fitting method:

- **Data Selection:** Limits analysis to data within 0.005–0.1 A/cm^2.
- **Candidate Windows:** Randomly selects 10 candidate 60 mV voltage windows from the corrected overpotential data.
- **Subwindow Analysis:** Each candidate is split into three subwindows (20 mV, 40 mV, and 60 mV) where linear regression (of η_{corr} vs. $\log_{10}(i)$) is performed.
- **Parameter Estimation:** The Tafel slope (b_{kin}) and intercept are averaged over the subwindows. The exchange current density i_0 is computed via

$$i_0 = 10^{-\text{intercept}/b_{kin}}.$$

- **Error Analysis:** The relative standard error (SE%) for both b_{kin} and i_0 is calculated. These are combined into a candidate metric which is used to select the best candidate.

`calculate_i0(b_kin, intercept)`: Provides a simple calculation of the exchange current density i_0 from the fitted Tafel parameters.

GUI and Main Structure

The code is organized into two main parts:

1. **GUI Implementation:**
   The class `IntegratedLSVAnalysisApp` implements the graphical user interface using Tkinter. It provides fields for data file selection, parameter input, and buttons to trigger the analysis and export the results. Diagnostic details, including the best candidate 60 mV window and its associated statistical metrics, are displayed in the GUI.

2. **Main Execution:**
   The main block creates an instance of the GUI class and starts the Tkinter event loop.

Overpotential Calculations

- **Reversible Potential (E_{rev}):**
  Calculated from the temperature using:

  $$E_{rev} = 1.2291 - 0.0008456 \times (T - 298.15).$$

- **Ohmic Overpotential (η_{ohm}):**
  Computed as:

  $$\eta_{ohm} = i \times \text{HFR}.$$

  This value is subtracted from the measured voltage to obtain the ohmic-corrected voltage.

- **Corrected Overpotential (η_{corr}):**
  Defined as:

  $$\eta_{\text{corr}} = (V - \eta_{ohm}) - E_{rev}.$$

- **Kinetic Overpotential (η_{kin}):**
  Obtained from the Tafel fitting process using:

  $$\eta_{kin} = b_{kin} \log_{10}\left(\frac{i}{i_0}\right),$$

  where b_{kin} and i_0 are determined by the best candidate window from the statistical analysis.

- **Catalyst Layer Overpotential (η_{RCL}):**
  Calculated using the model:

  $$\eta_{RCL} = -b_{kin} \log_{10}(U_J),$$

  with

  $$U_J = \left[1 + \left(\frac{i \cdot \ln(10) \cdot R_{CL}}{2b_{kin}}\right)^{1.1982}\right]^{-1/1.1982}.$$

- **Residual Overpotential (η_{res}):**
  Represents the difference between the total measured overpotential and the sum of the calculated components:

  $$\eta_{res} = (V - E_{rev}) - (\eta_{kin} + \eta_{ohm} + \eta_{RCL}).$$