OSIF models

1. Transmission Line Model (Default Model)

Mathematical Formulation:

The Transmission Line model used is based on the Setzler et al. model:

$$Z(\omega) = L_{\mathrm{wire}}(j\omega)^{\theta} + R_{\mathrm{mem}} + \sqrt{\frac{R_{\mathrm{cl}}}{Q_{\mathrm{dl}}(j\omega)^{\phi}}} \coth\left(\sqrt{R_{\mathrm{cl}}Q_{\mathrm{dl}}(j\omega)^{\phi}}\right)$$

- L_{wire} (inductance-like parameter)
- R_{mem} (membrane resistance)
- R_{cl} (catalyst layer resistance)
- Q_{dl} (constant phase element related parameter)
- ϕ (constant phase element exponent, typically between 0 and 1)
- θ (frequency exponent for inductance-like element, typically close to 1 for a pure inductance)

The code implementing this model is:

```
def funcreal(self, param):
    return np.real(
        param[0] * pow((1j * 2 * np.pi * self.activeData.frequency),
param[5]) +
        param[1] +
        pow((param[2] / (param[3] * pow((1j * 2 * np.pi *
self.activeData.frequency), param[4]))), 0.5) *
        self.JPcoth(pow((param[2] * param[3] * pow((1j * 2 * np.pi *
self.activeData.frequency), param[4])), 0.5))
    )

def funcImg(self, param):
    return np.imag(
        param[0] * pow((1j * 2 * np.pi * self.activeData.frequency),
param[5]) +
        param[1] +
        pow((param[2] / (param[3] * pow((1j * 2 * np.pi *
self.activeData.frequency), param[4]))), 0.5) *
        self.JPcoth(pow((param[2] * param[3] * pow((1j * 2 * np.pi *
```

```
self.activeData.frequency), param[4])), 0.5))
    )
```

Here, the `JPcoth` function is defined as:

```
def JPcoth(self, x):
    return (pow(np.e, x) + pow(np.e, -x)) / (pow(np.e, x) - pow(np.e, -x))
```

2. 1-D Linear Diffusion Model

Mathematical Formulation:

$$Z(\omega) = L_{\text{wire}}(j\omega)^{\theta} + R_{\text{mem}} + R_{\text{cl}}\frac{\coth\left(\sqrt{R_{\text{cl}}Q_{\text{dl}}(j\omega)^{\phi}}\right)}{\sqrt{R_{\text{cl}}Q_{\text{dl}}(j\omega)^{\phi}}}$$

Code implementation:

```
def funcreal_l(self, param):
    return np.real(
        param[0] * pow((1j * 2 * np.pi * self.activeData.frequency),
param[5]) +
        param[1] + param[2] * pow((param[2] * (param[3] * pow((1j * 2 *
np.pi * self.activeData.frequency), param[4]))), -0.5) *
        self.JPcoth(pow((param[2] * param[3] * pow((1j * 2 * np.pi *
self.activeData.frequency), param[4])), 0.5))
    )

def funcImg_l(self, param):
    return np.imag(
        param[0] * pow((1j * 2 * np.pi * self.activeData.frequency),
param[5]) +
        param[1] + param[2] * pow((param[2] * (param[3] * pow((1j * 2 *
np.pi * self.activeData.frequency), param[4]))), -0.5) *
        self.JPcoth(pow((param[2] * param[3] * pow((1j * 2 * np.pi *
self.activeData.frequency), param[4])), 0.5))
    )
```

3. 1-D Spherical Diffusion Model

Mathematical Formulation:

1-D spherical Warburg-type diffusion impedance model:

$$Z_{\text{diff, spherical}}(\omega) = \frac{R_{\text{cl}}}{\sqrt{R_{\text{cl}} Q_{\text{dl}} (j\omega)^{\phi}} \coth\left(\sqrt{R_{\text{cl}} Q_{\text{dl}} (j\omega)^{\phi}}\right) - 1}$$

The complete impedance model:

$$Z(\omega) = L_{\text{wire}}(j\omega)^{\theta} + R_{\text{mem}} + \frac{R_{\text{cl}}}{\sqrt{R_{\text{cl}} Q_{\text{dl}} (j\omega)^{\phi}} \coth\left(\sqrt{R_{\text{cl}} Q_{\text{dl}} (j\omega)^{\phi}}\right) - 1}$$

Code implementation:

```
def funcreal_s(self, param):
    return np.real(
        param[0] * pow((1j * 2 * np.pi * self.activeData.frequency),
param[5]) +
        param[1] + param[2] /
        (pow(param[2] * param[3] * pow((1j * 2 * np.pi *
self.activeData.frequency), param[4]), 0.5) *
        self.JPcoth(pow((param[2] * param[3] * pow((1j * 2 * np.pi *
self.activeData.frequency), param[4])),0.5)) - 1)
    )

def funcImg_s(self, param):
    return np.imag(
        param[0] * pow((1j * 2 * np.pi * self.activeData.frequency),
param[5]) +
        param[1] + param[2] /
        (pow(param[2] * param[3] * pow((1j * 2 * np.pi *
self.activeData.frequency), param[4]), 0.5) *
        self.JPcoth(pow((param[2] * param[3] * pow((1j * 2 * np.pi *
self.activeData.frequency), param[4])),0.5)) - 1)
    )
```

---

2. Understanding and Comparison

Transmission line model

$$Z(\omega) = L_{\text{wire}}(j\omega)^{\theta} + R_{\text{mem}} + \sqrt{\frac{R_{\text{cl}}}{Q_{\text{dl}}(j\omega)^{\phi}}} \coth\left(\sqrt{R_{\text{cl}} Q_{\text{dl}} (j\omega)^{\phi}}\right)$$

- Represents a **distributed resistor-capacitor network**.
- **Commonly used for porous electrodes** (e.g., PEM fuel cell catalyst layers) or any system exhibiting distributed charge-transfer reactions along a conductive pathway.
- Transmission line analogy: captures the gradual distribution of resistance and capacitance along electrode thickness or pore length.

Pros:

- Captures complex, realistic electrode structures.
- Well-suited for porous and composite electrodes.
- Highly flexible; accurately models frequency-dependent impedance characteristics (e.g., inductive loops at low frequencies).

Cons:

- More parameters mean greater complexity and potential for parameter correlation (overfitting).
- Requires careful interpretation; parameters like θ, ϕ, and Q_{dl} can lose clear physical interpretability.

1-D Linear Diffusion Model (Warburg impedance)

$$Z(\omega) = L_{\mathrm{wire}}(j\omega)^{\theta} + R_{\mathrm{mem}} + R_{\mathrm{cl}}\frac{\coth\left(\sqrt{R_{\mathrm{cl}}Q_{\mathrm{dl}}(j\omega)^{\phi}}\right)}{\sqrt{R_{\mathrm{cl}}Q_{\mathrm{dl}}(j\omega)^{\phi}}}$$

- Describes linear, one-dimensional diffusion processes of ions/molecules at planar electrodes.
- Represents a **Warburg-type diffusion** that occurs when reaction rates are limited by linear diffusion from bulk electrolyte to electrode surfaces.

Pros:

- Standard model: straightforward interpretation.
- Parameters directly related to well-known physical diffusion properties.
- Suitable for electrodes where diffusion is essentially linear (e.g., planar geometry).

Cons:

- Not suitable for porous or complex geometries where diffusion is non-linear or multi-dimensional.
- May fail to accurately describe real systems exhibiting distributed or non-ideal characteristics.

1-D Spherical Diffusion Model (Warburg impedance)

$$Z(\omega) = L_{\text{wire}}(j\omega)^{\theta} + R_{\text{mem}} + \frac{R_{\text{cl}}}{\sqrt{R_{\text{cl}}Q_{\text{dl}}(j\omega)^{\phi}}\coth\left(\sqrt{R_{\text{cl}}Q_{\text{dl}}(j\omega)^{\phi}}\right) - 1}$$

- Represents diffusion-limited processes for spherical particles or electrodes.
- Commonly applied to battery electrodes, nano-sized particles, or any electrode/electrolyte interfaces with spherical geometry.

Pros:

- Captures curvature effects inherent in spherical electrode geometries.
- Suitable for nanoparticle-based electrodes or electrodes composed of spherical particles.
- Offers insights into finite-length (radius-based) diffusion, relevant for many practical battery and supercapacitor systems.

Cons:

- Specific to spherical geometries, thus limited to certain systems.
- Complexity in interpretation of parameters if the actual physical geometry deviates from ideal spherical assumptions.

Aspect	Transmission Line	1-D Linear Diffusion	1-D Spherical Diffusion
Geometry	Porous/Distributed	Planar, linear	Spherical particle-based
Physical Meaning	Distributed resistance & capacitance	Diffusion-limited planar	Diffusion-limited spherical
Parameter Complexity	High (more parameters, more complexity)	Moderate (standard diffusion)	Moderate-high (geometry-specific)
Applicability	Porous electrodes, fuel cells, PEMFC	Planar electrodes, simpler cells	Batteries, nanoparticles, spherical electrodes
Interpretability	Moderate (parameters may become abstract)	High (parameters related directly to	High for spherical systems, low otherwise

Aspect	Transmission Line	1-D Linear Diffusion	1-D Spherical Diffusion
		physical properties)	
Typical systems	Fuel cell catalyst layers, porous electrodes, porous capacitors	Electrochemical sensors, planar electrode experiments, thin-layer cells	Lithium-ion battery cathodes/anodes, nanoparticle-modified electrodes
Accuracy	High (if correctly applied)	Low-medium (may miss geometry)	Medium-high (geometry-dependent)

The L_{wire}

The Transmission Line model includes multiple parameters, some with similar effects, making it easy for parameters (especially L_{wire}) to become redundant or statistically insignificant.

- The inductance-like parameter L_{wire} influences mainly the **high-frequency** region of impedance spectrum.
- If the experimental frequency range does **not extend sufficiently high**, or if the inductive loop is small or absent, the data will provide very little constraint for accurately estimating Lwire
- If the data doesn't explicitly show inductive loops (positive imaginary component at high frequency), the optimization algorithm will struggle to accurately define inductive parameters.

So just set L_{wire} to 0 and ignore it when fitting.