



Caratterizzazione di un diodo tramite spettroscopia d'impedenza

Simone Vortice
N85001494

Relatore:

- Dott. Pierluigi Casolaro

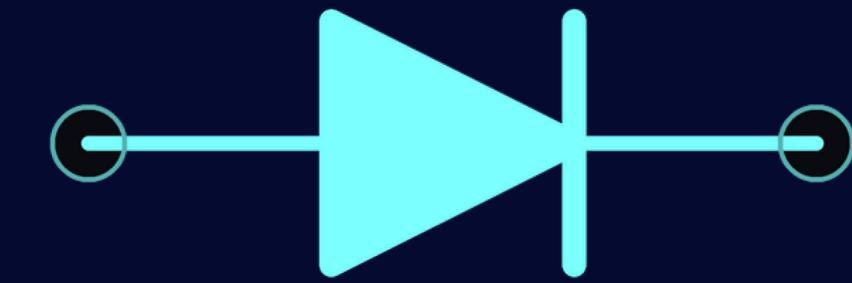
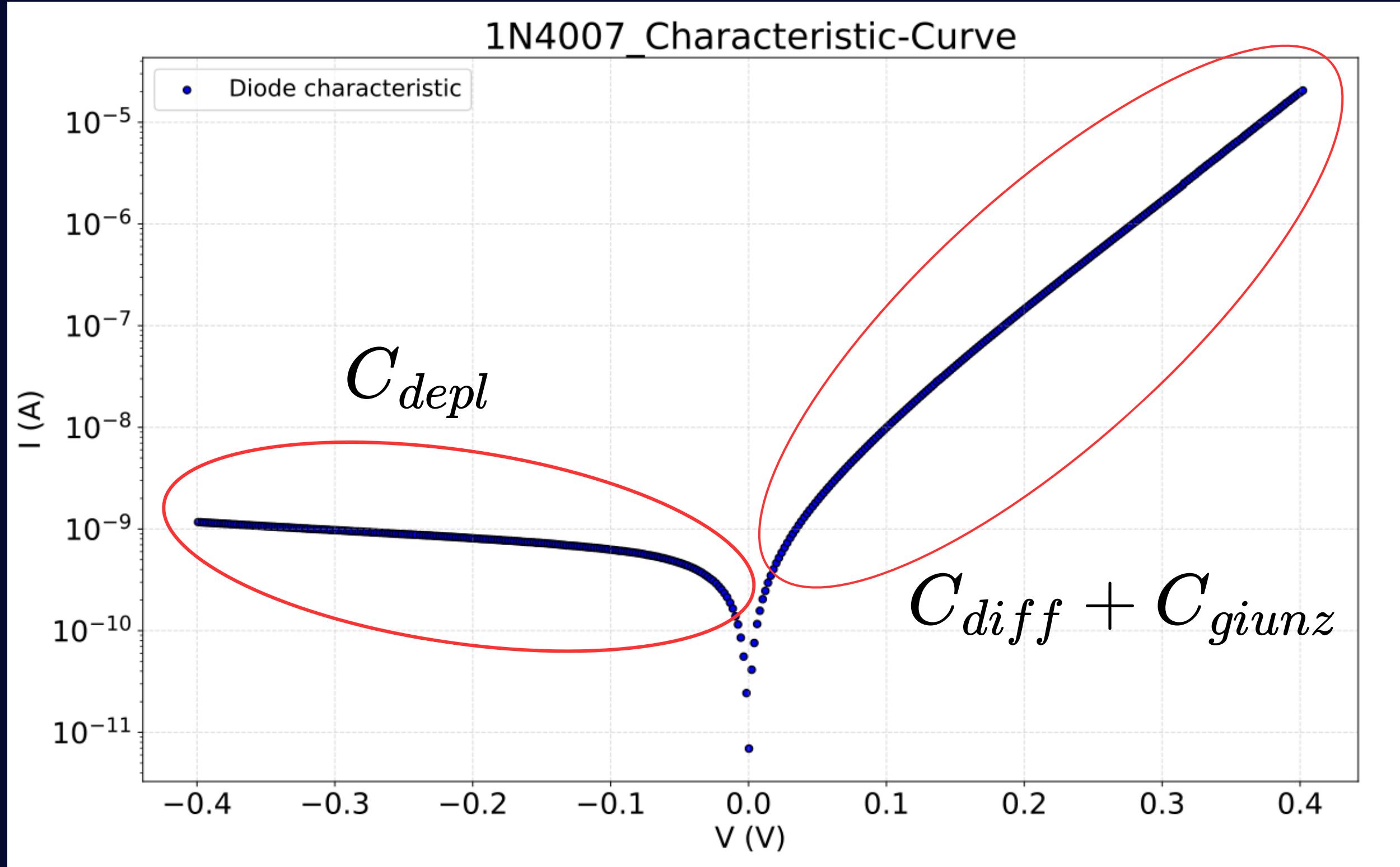
Correlatore:

- Prof. Alberto Aloisio

Sommario

- 1 Diodo a giunzione pn
- 2 Circuiti Equivalenti
- 3 Spettroscopia d'impedenza
- 4 Strumentazione
- 5 Fit dei parametri

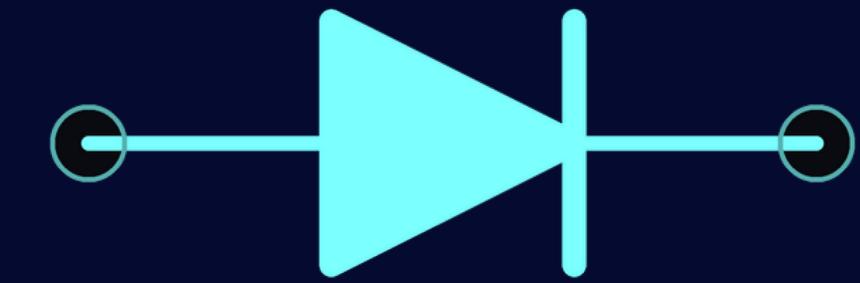
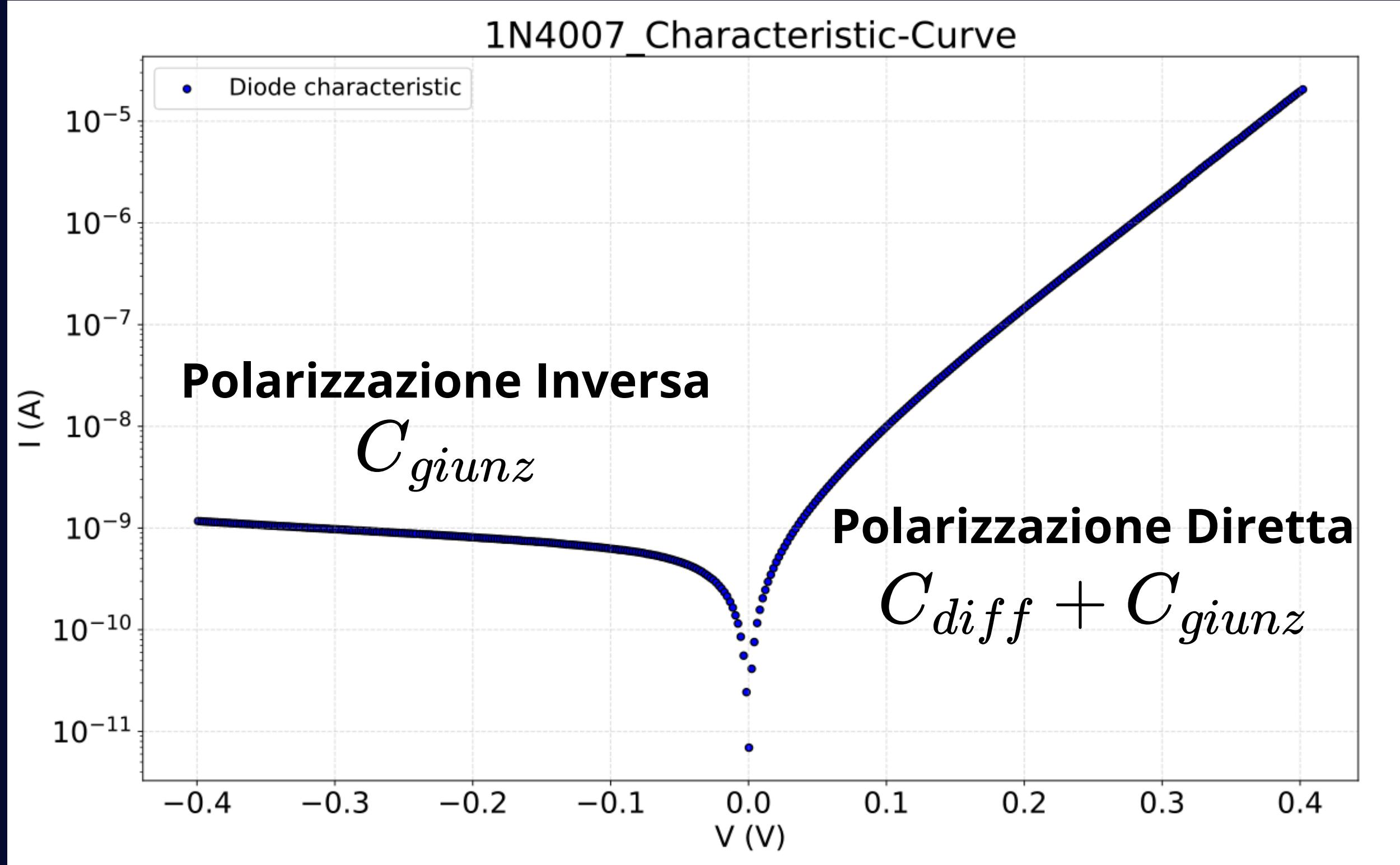
Diodo a giunzione pn



Equazione di Shockley:

$$I(V) = I_S \left(e^{\frac{V}{V_T}} - 1 \right)$$

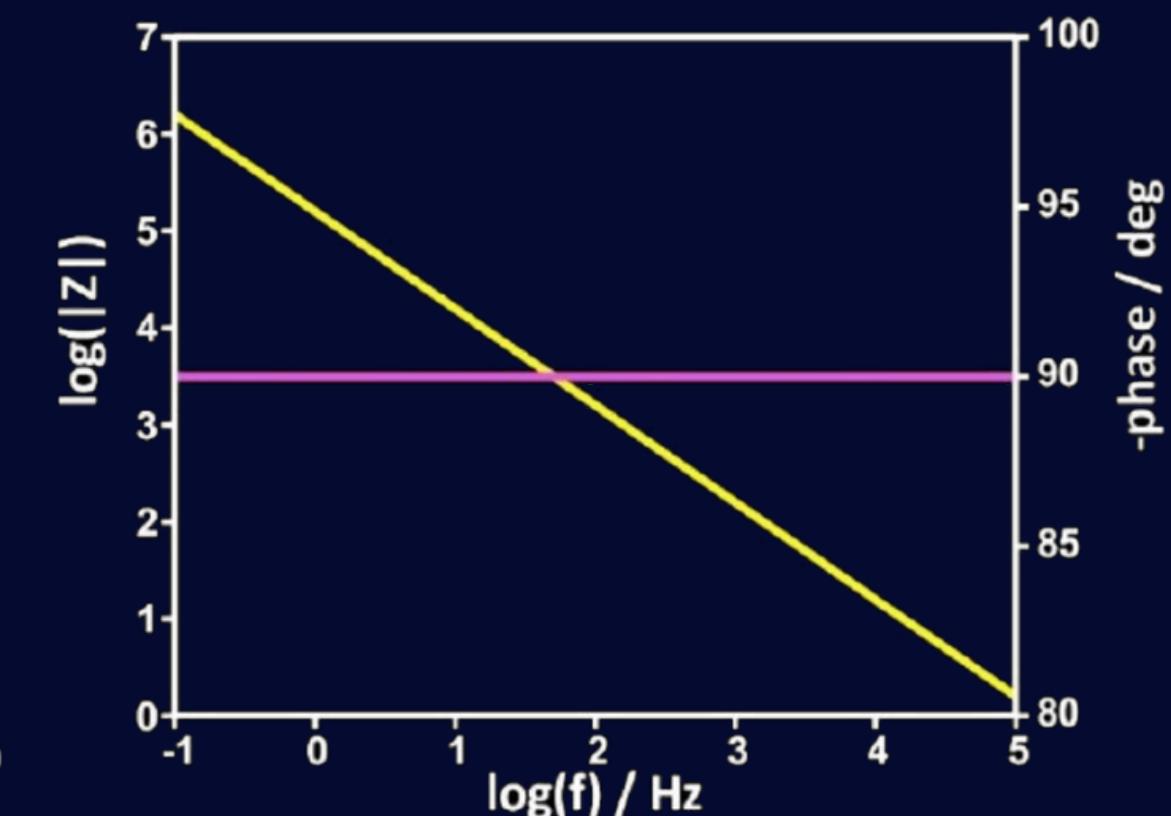
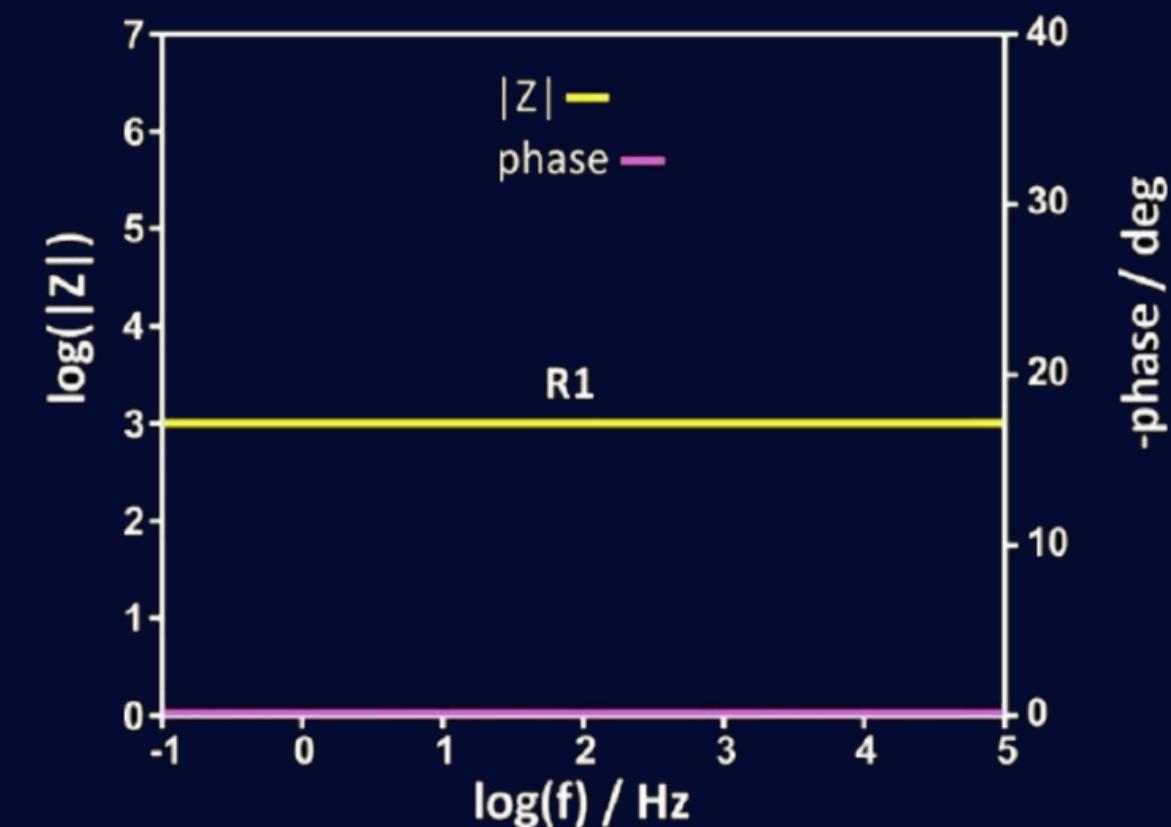
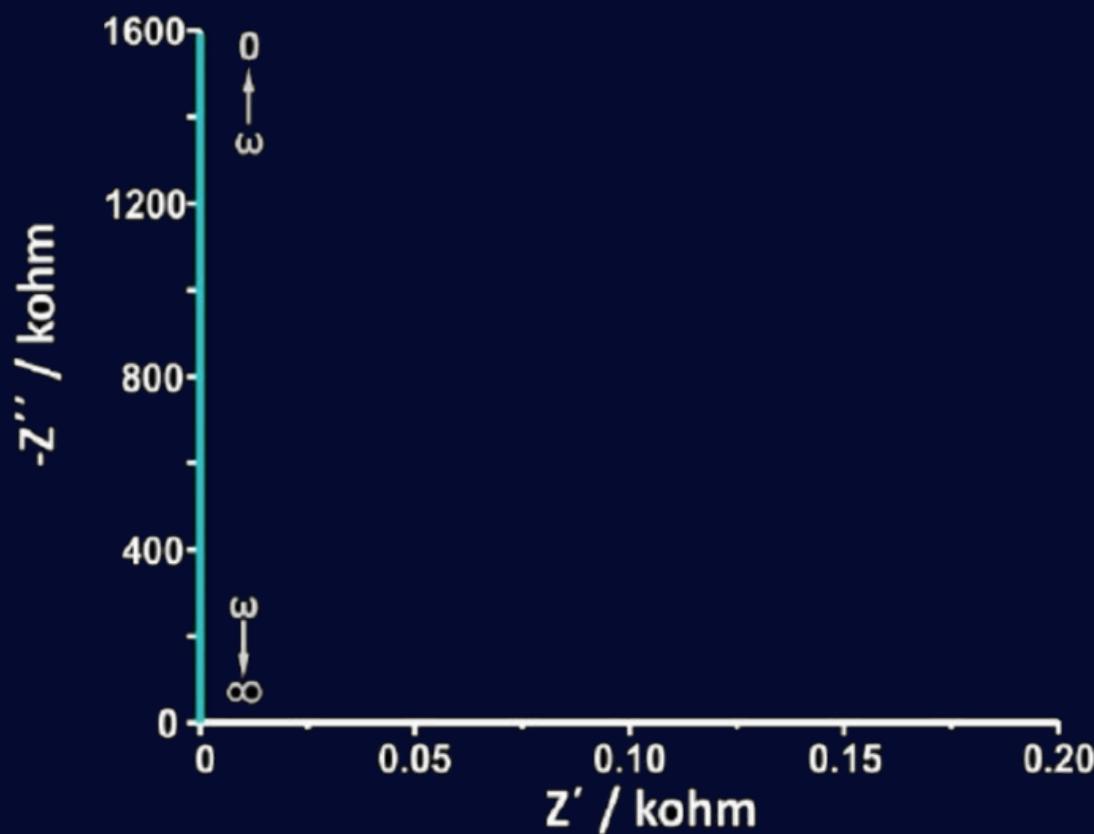
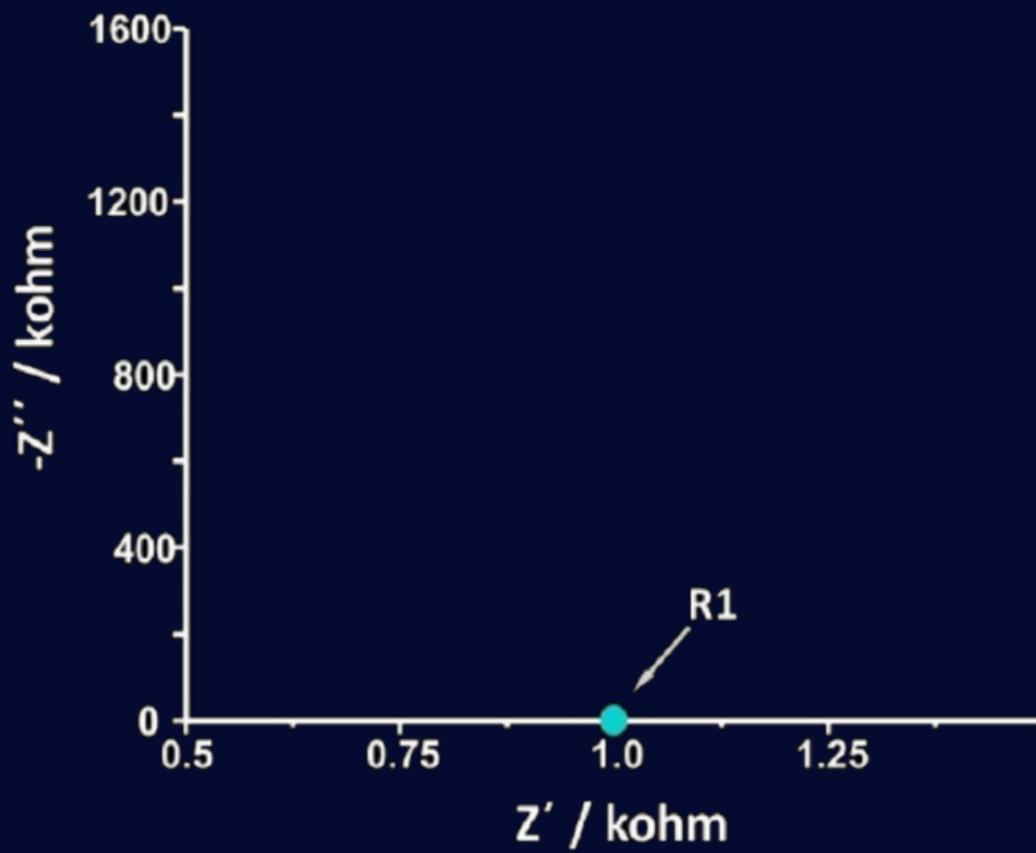
Diodo a giunzione pn



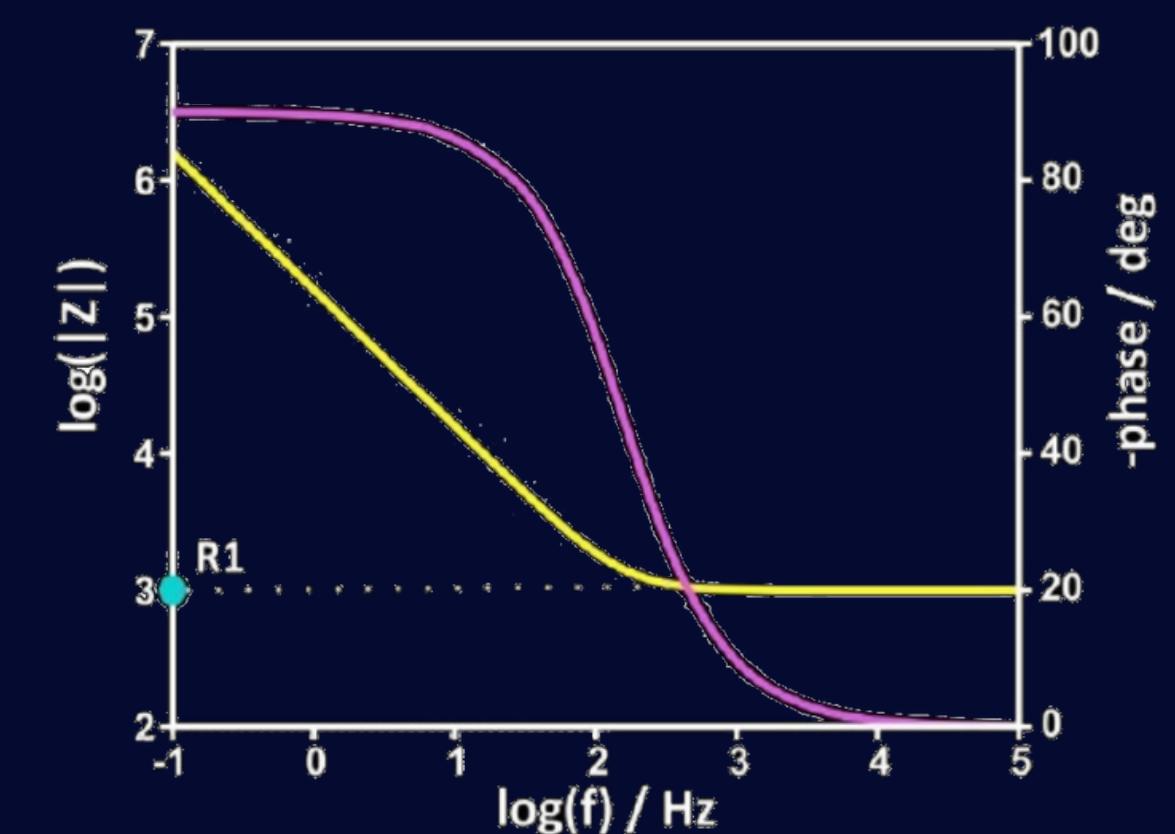
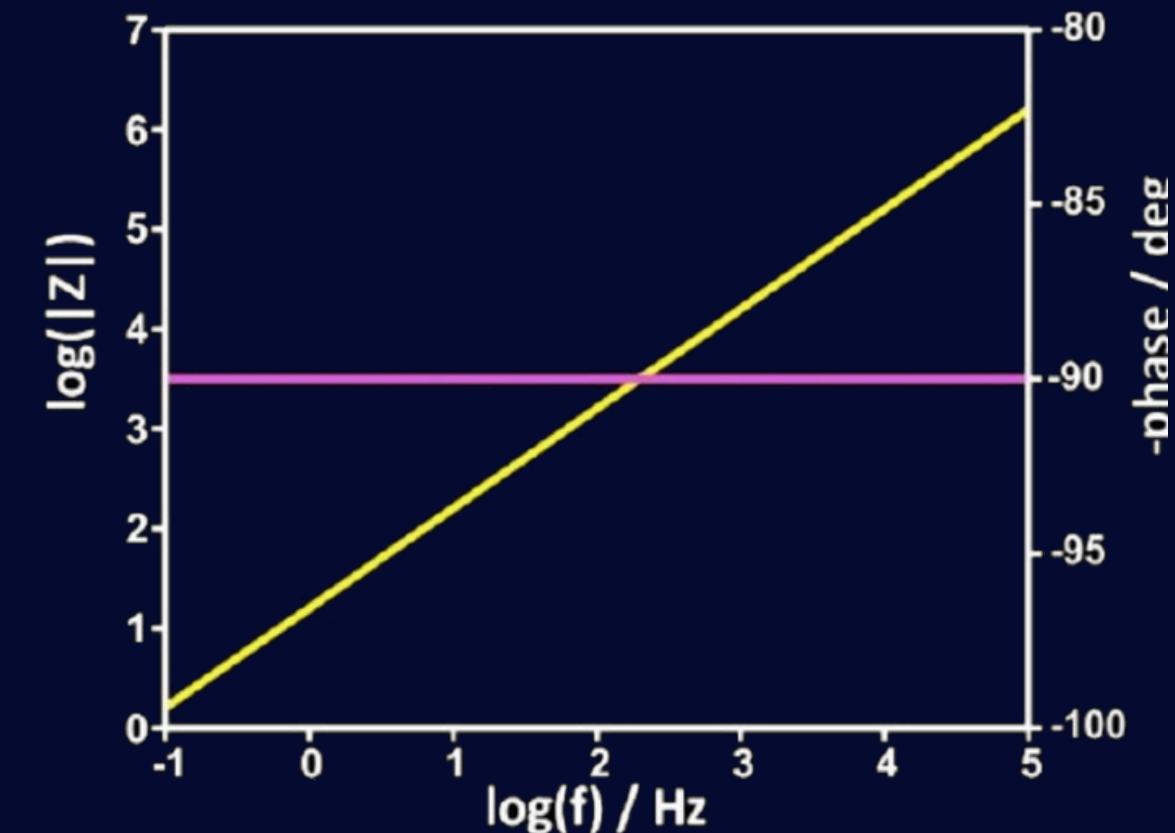
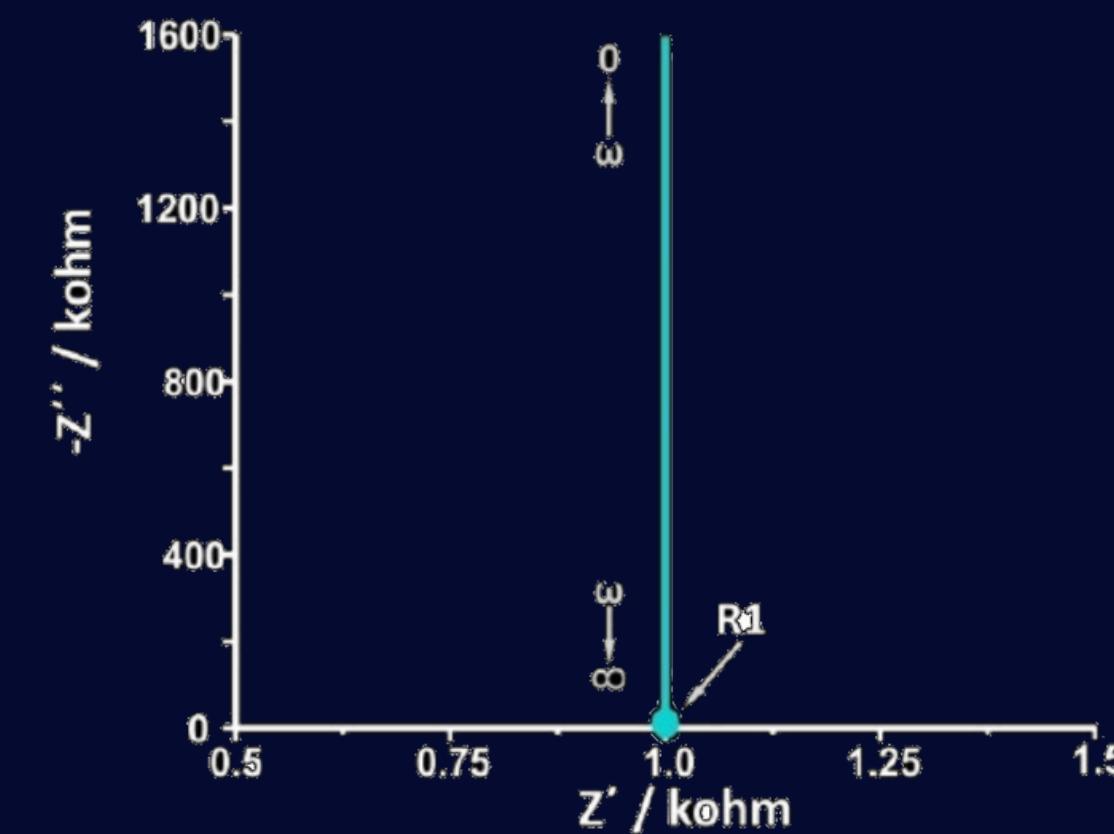
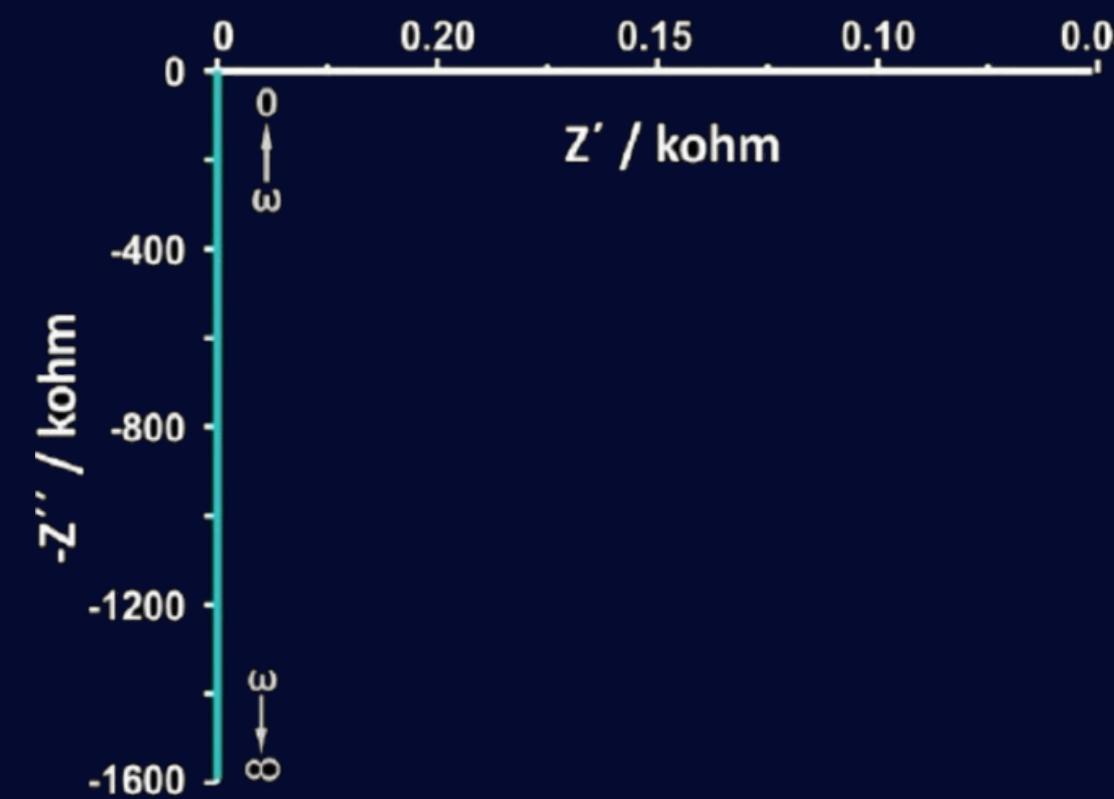
Equazione di Shockley:

$$I(V) = I_S \left(e^{\frac{V}{V_T}} - 1 \right)$$

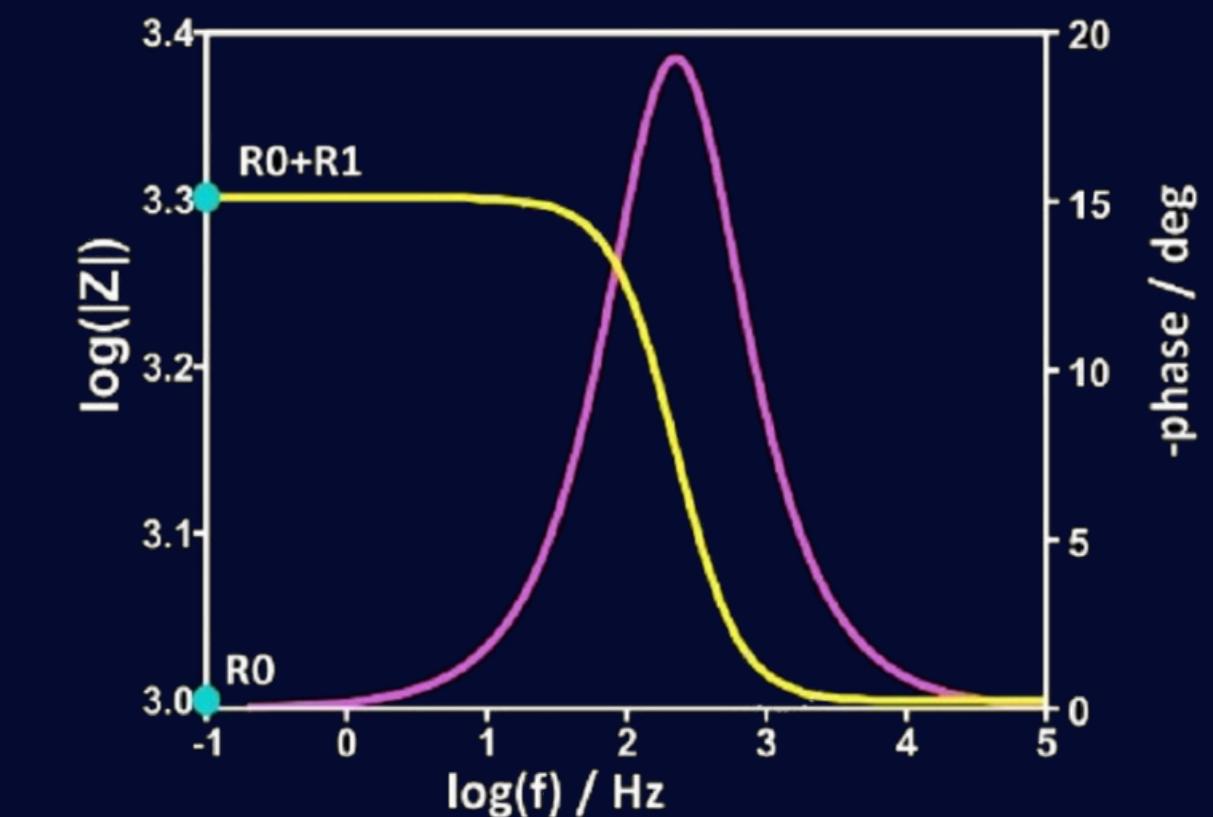
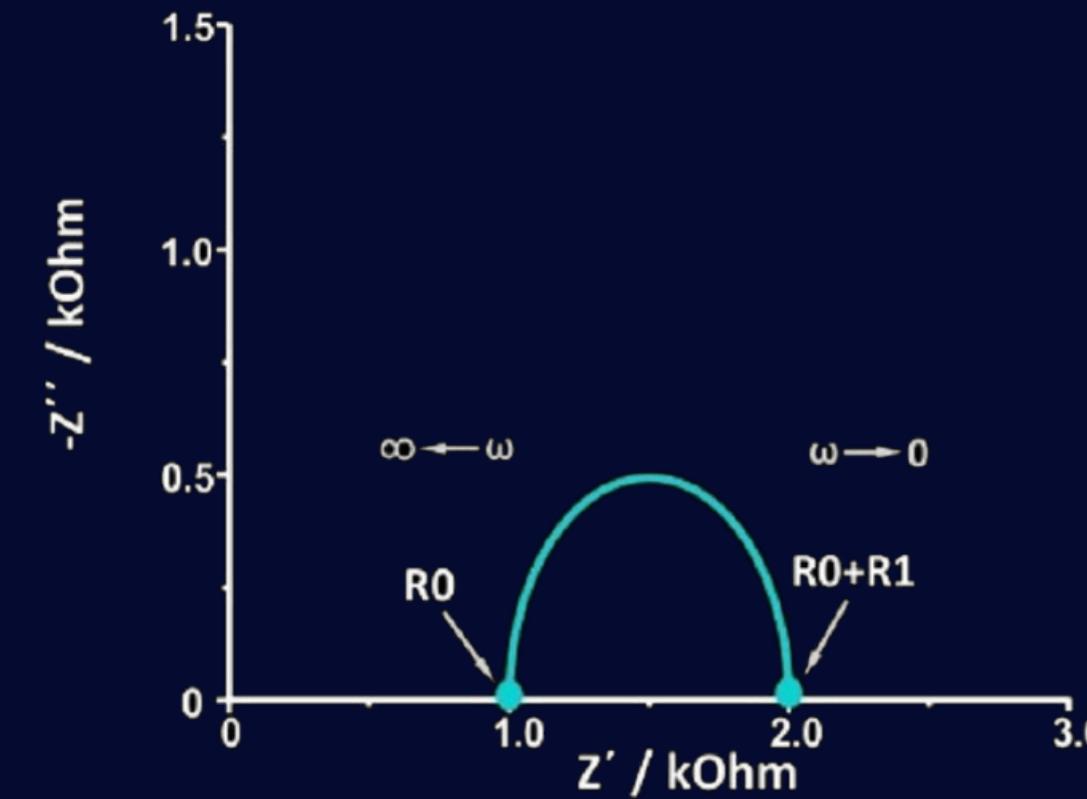
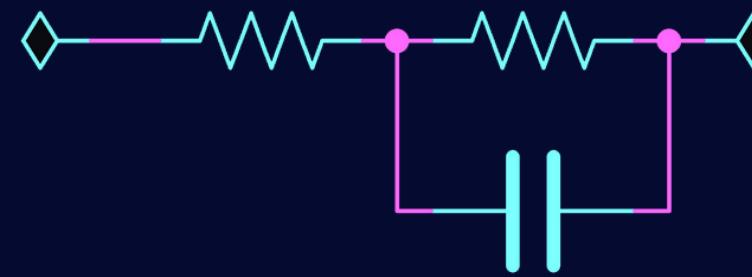
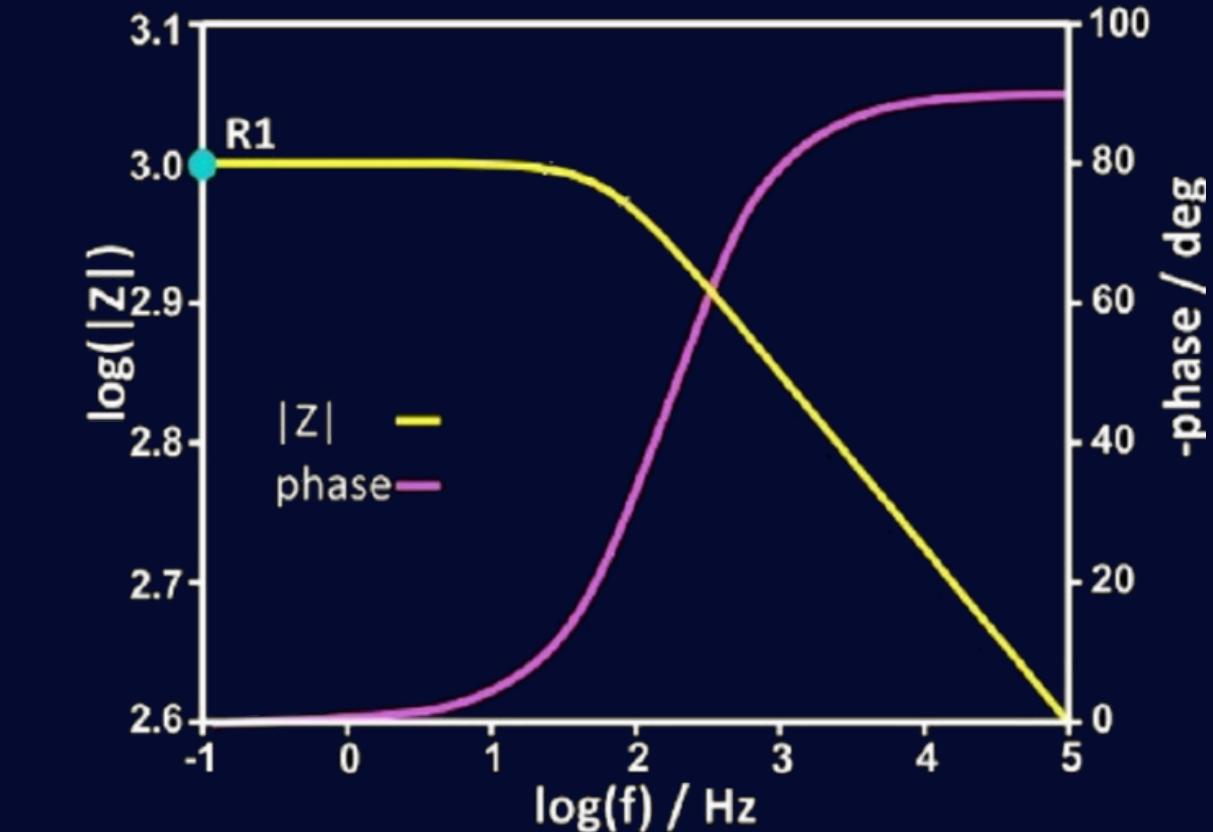
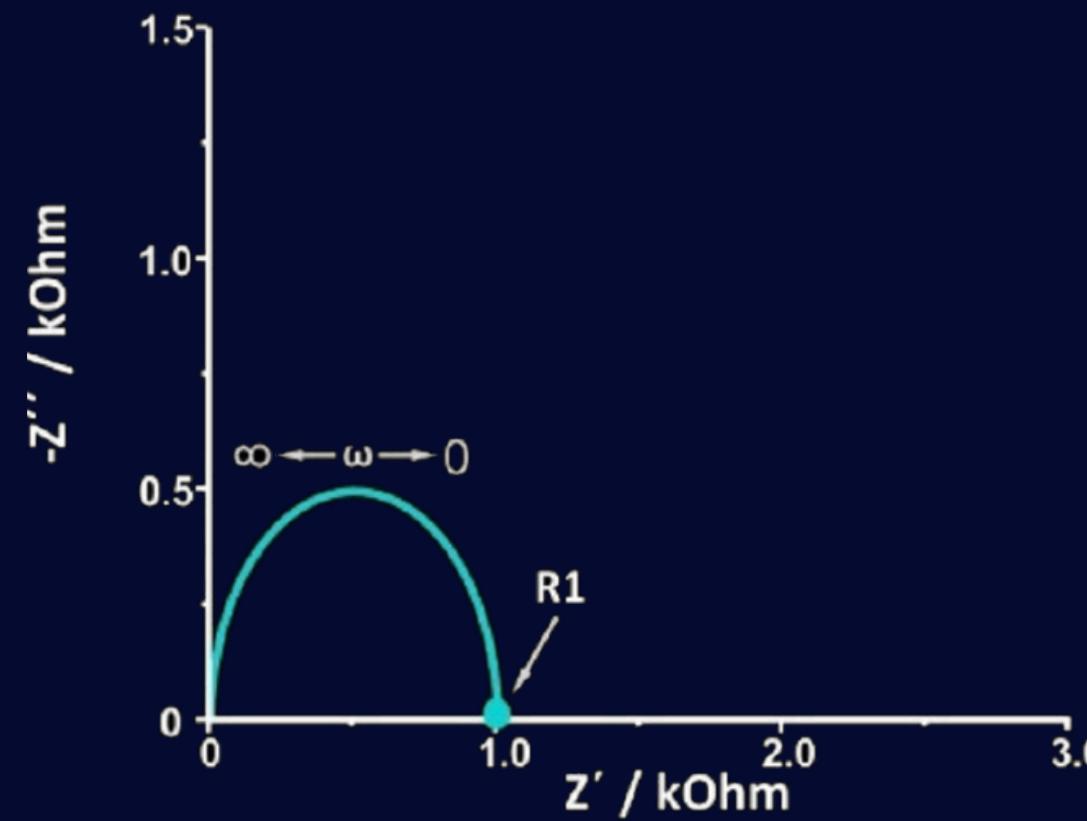
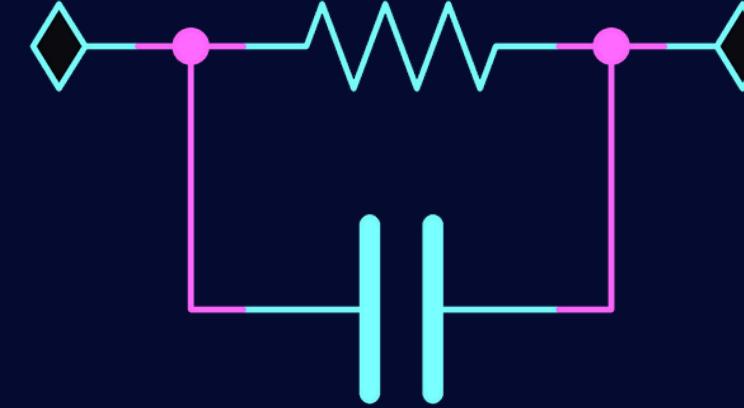
Circuiti equivalenti



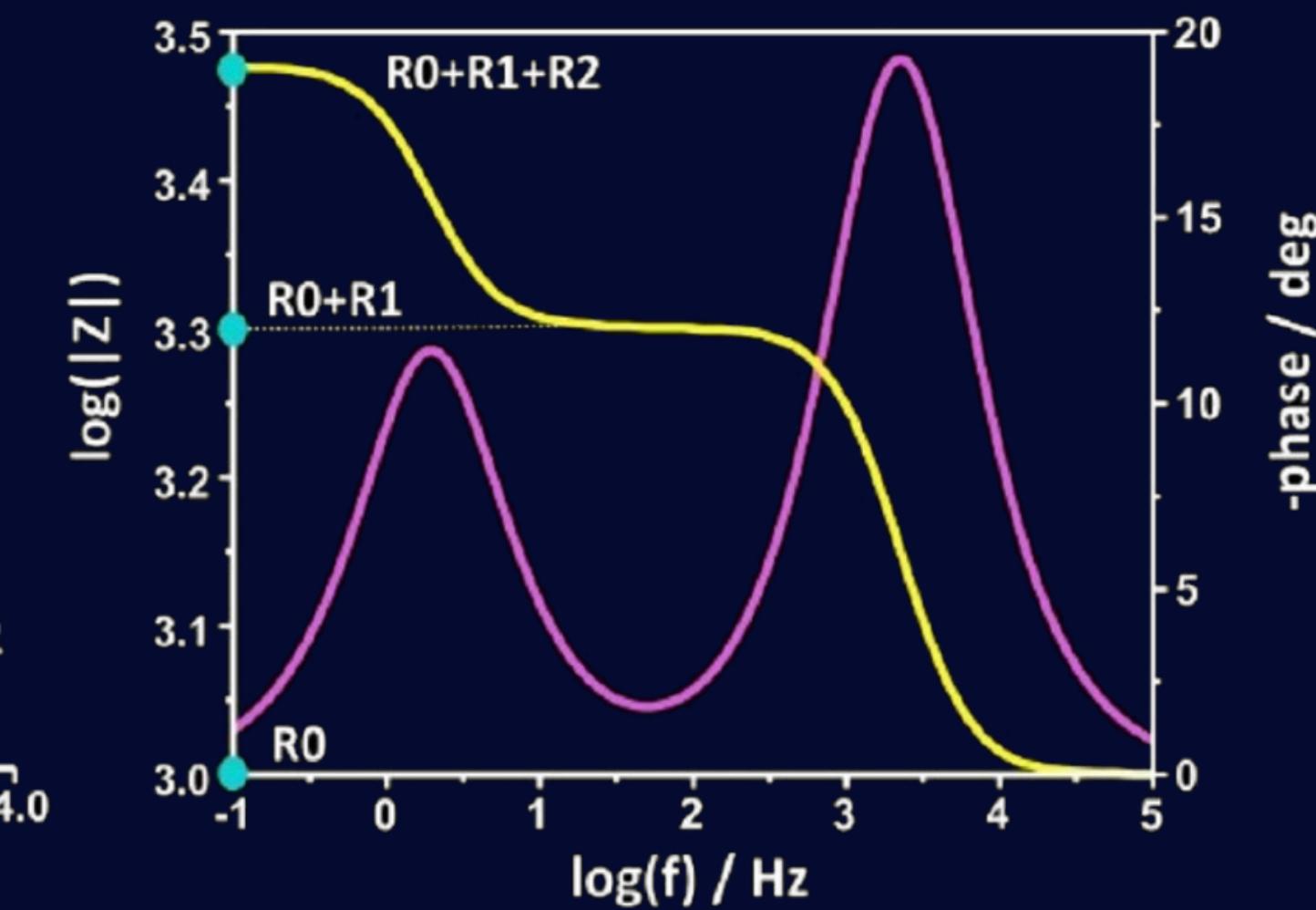
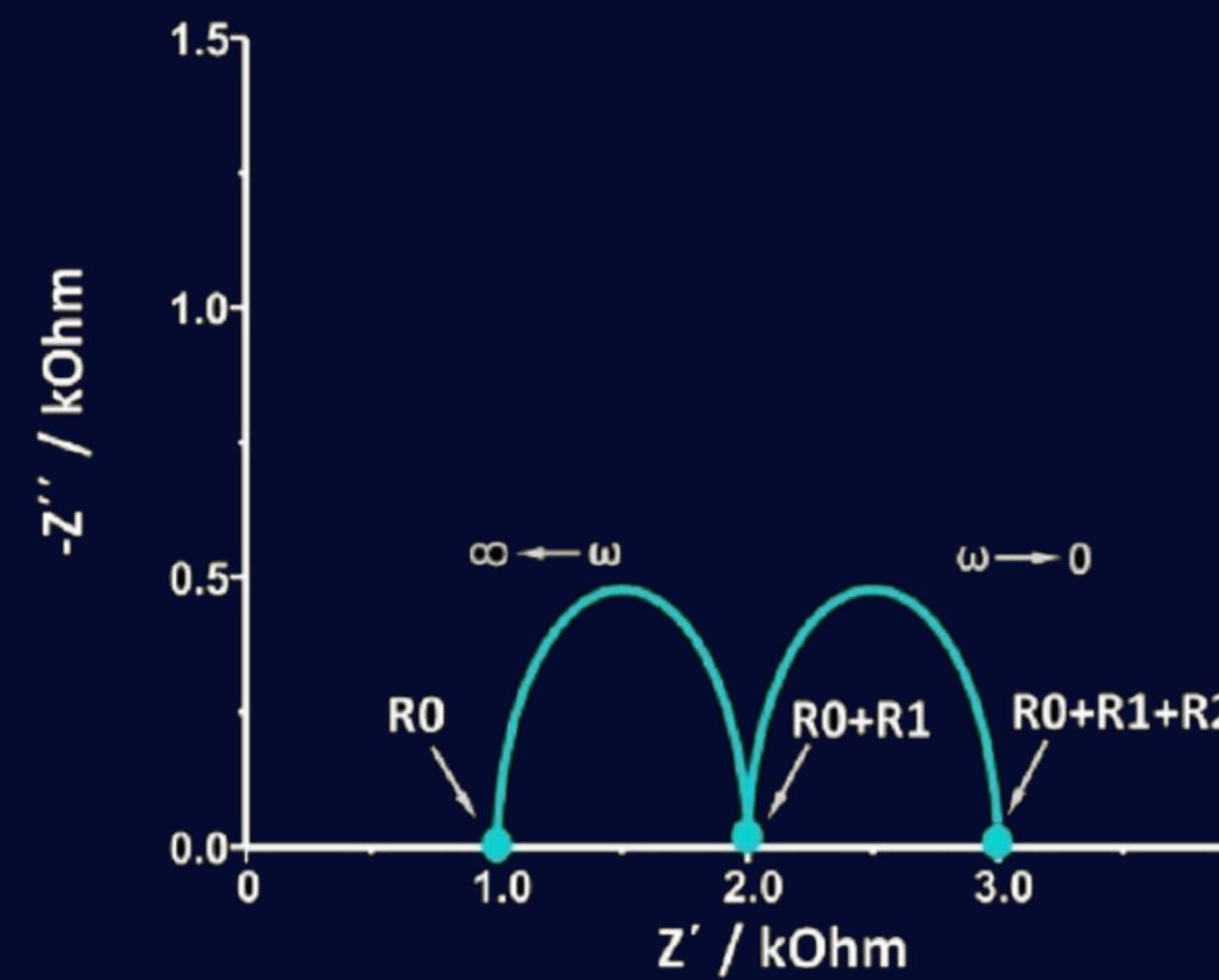
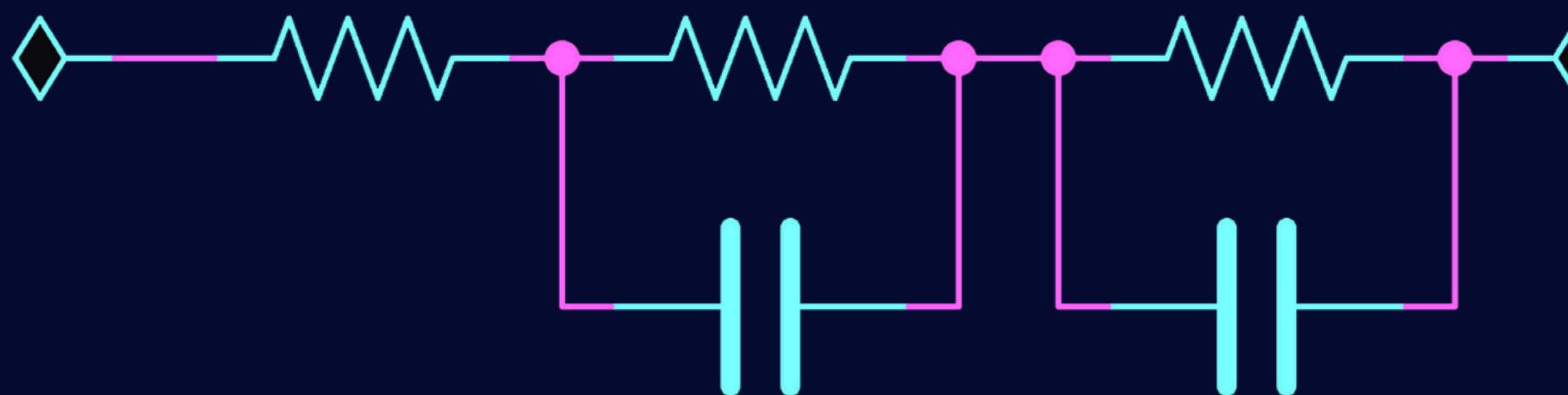
Circuiti equivalenti



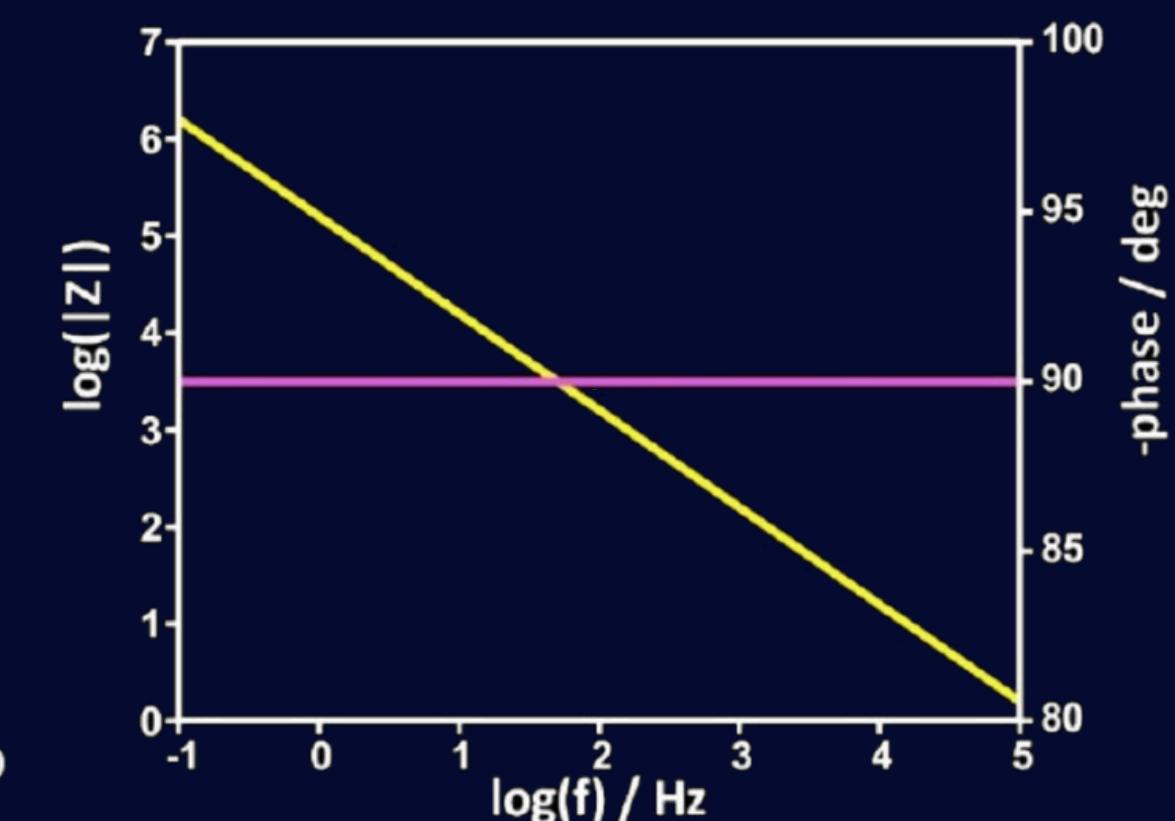
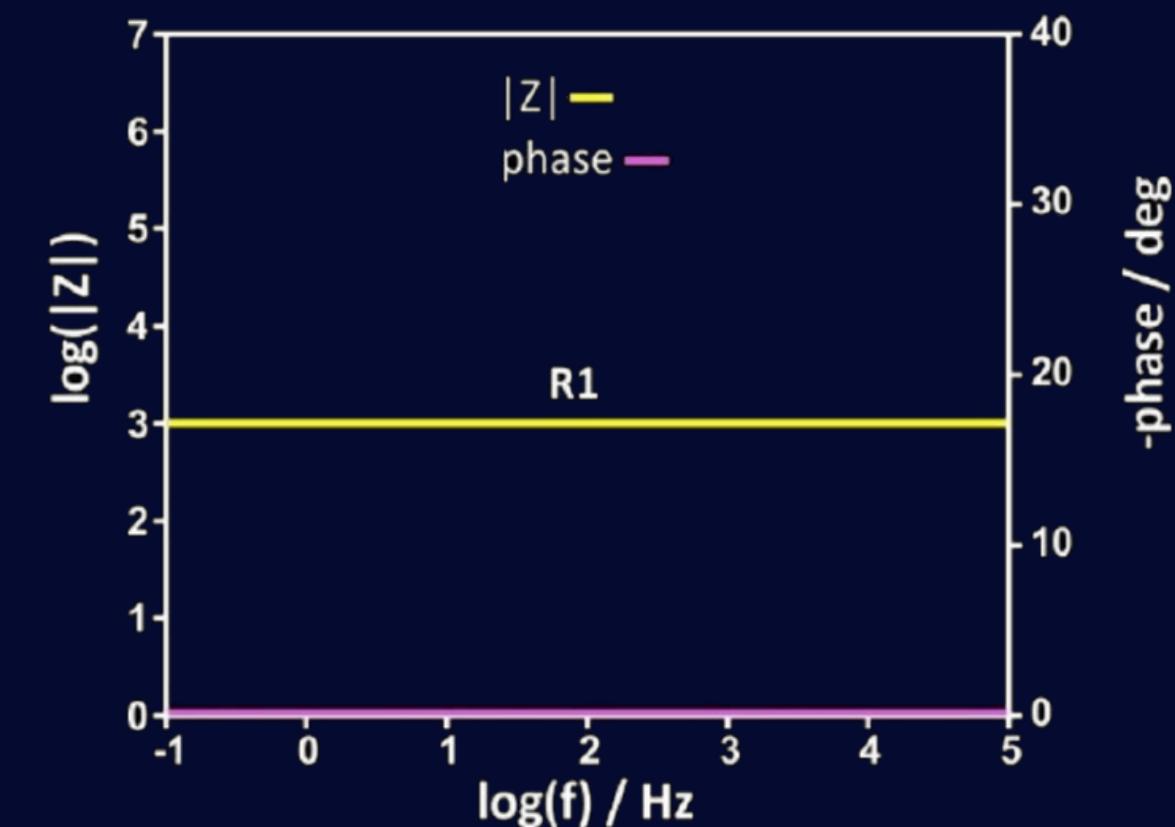
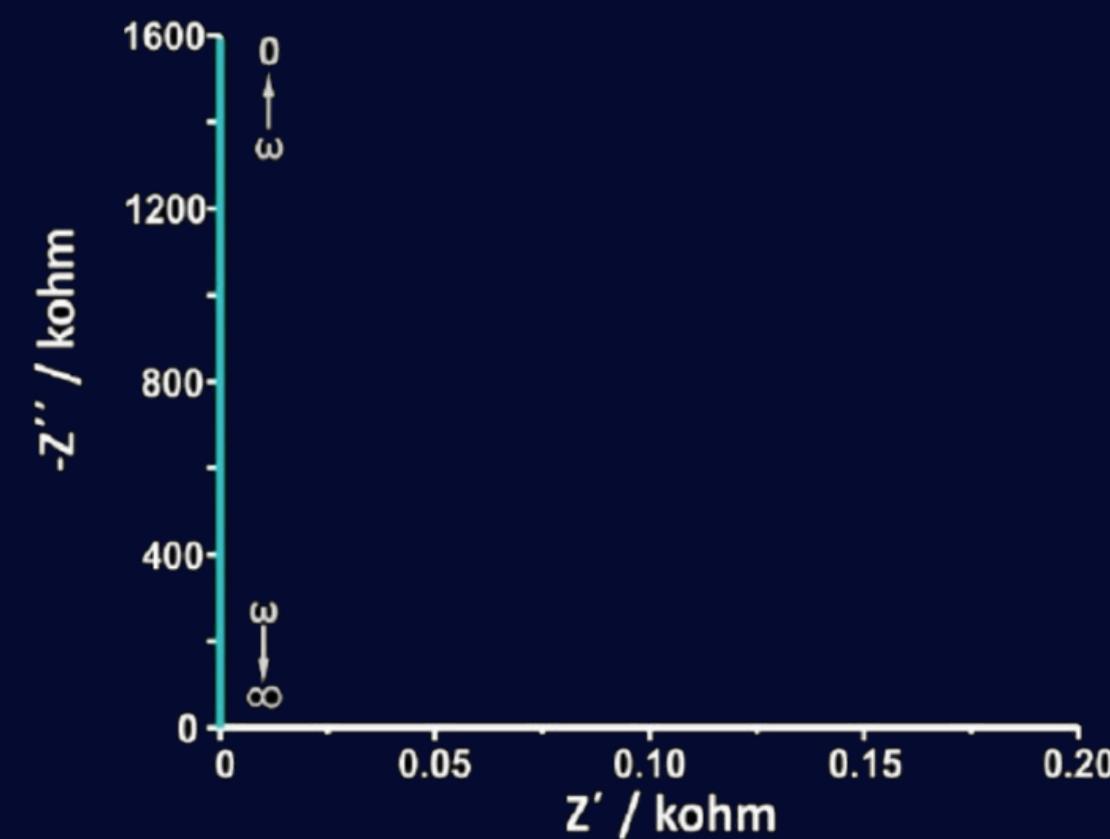
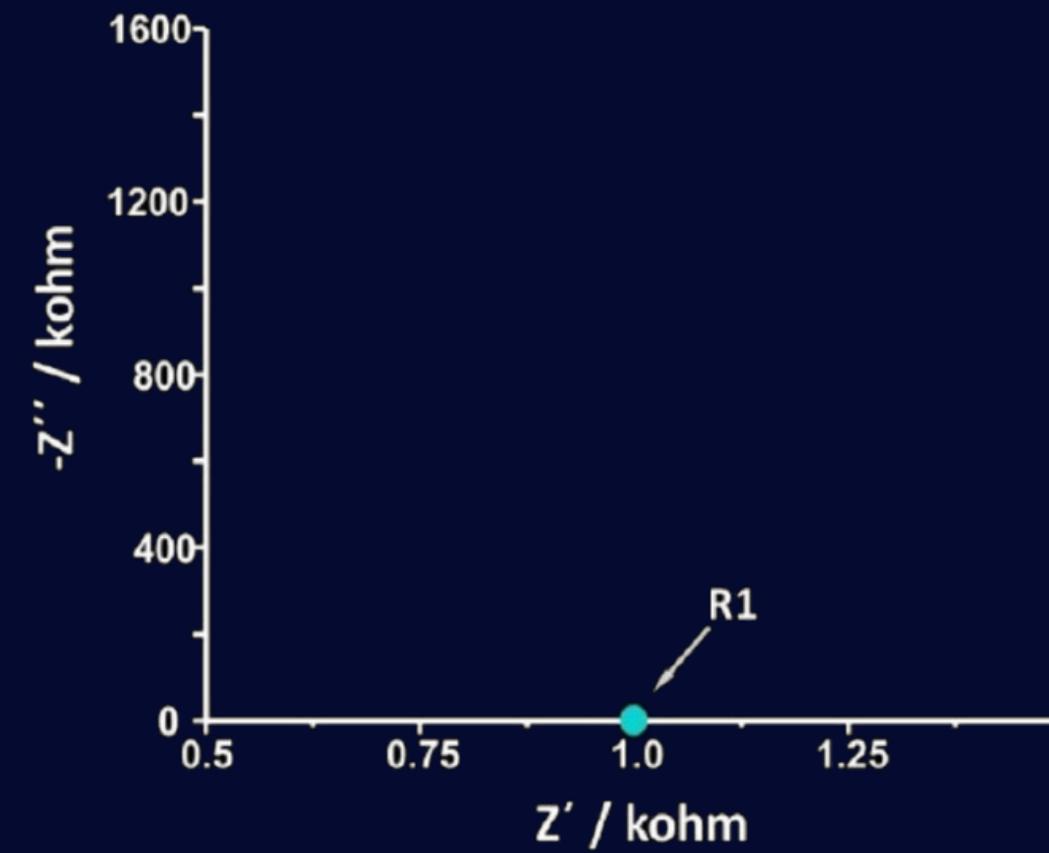
Circuiti equivalenti



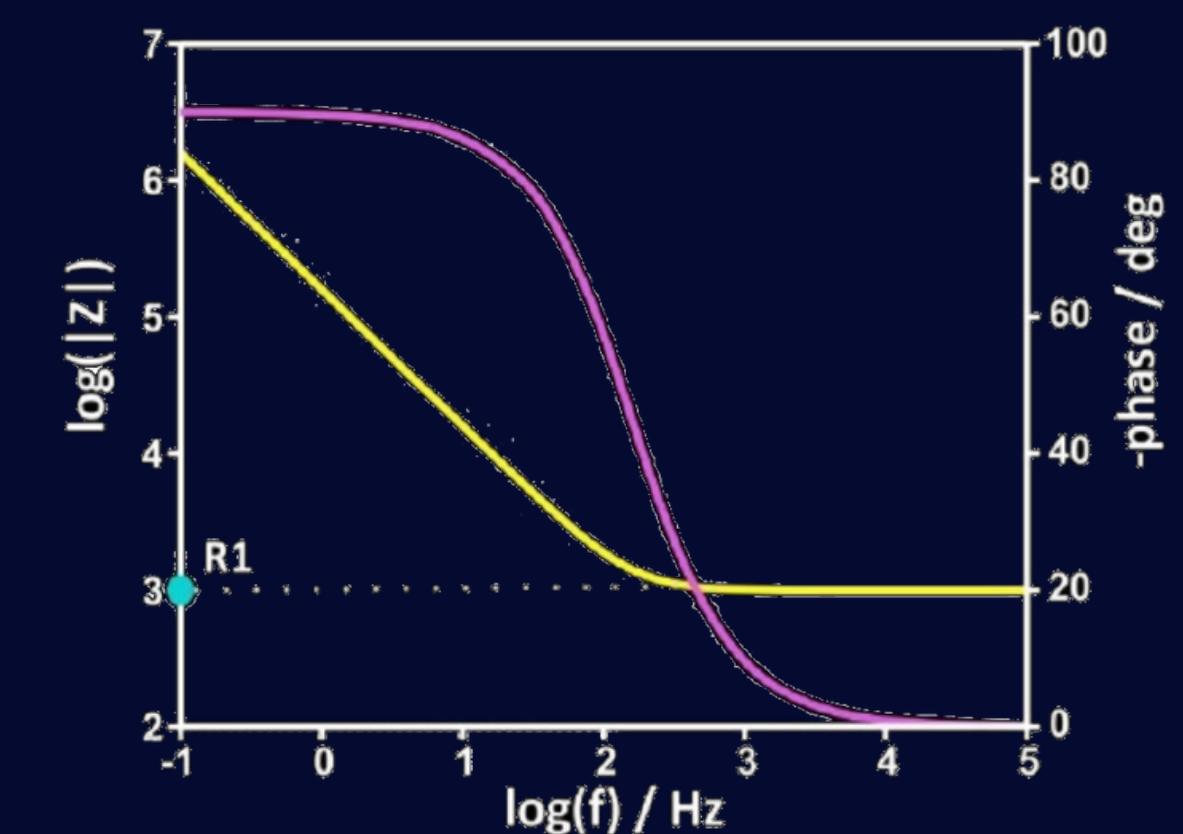
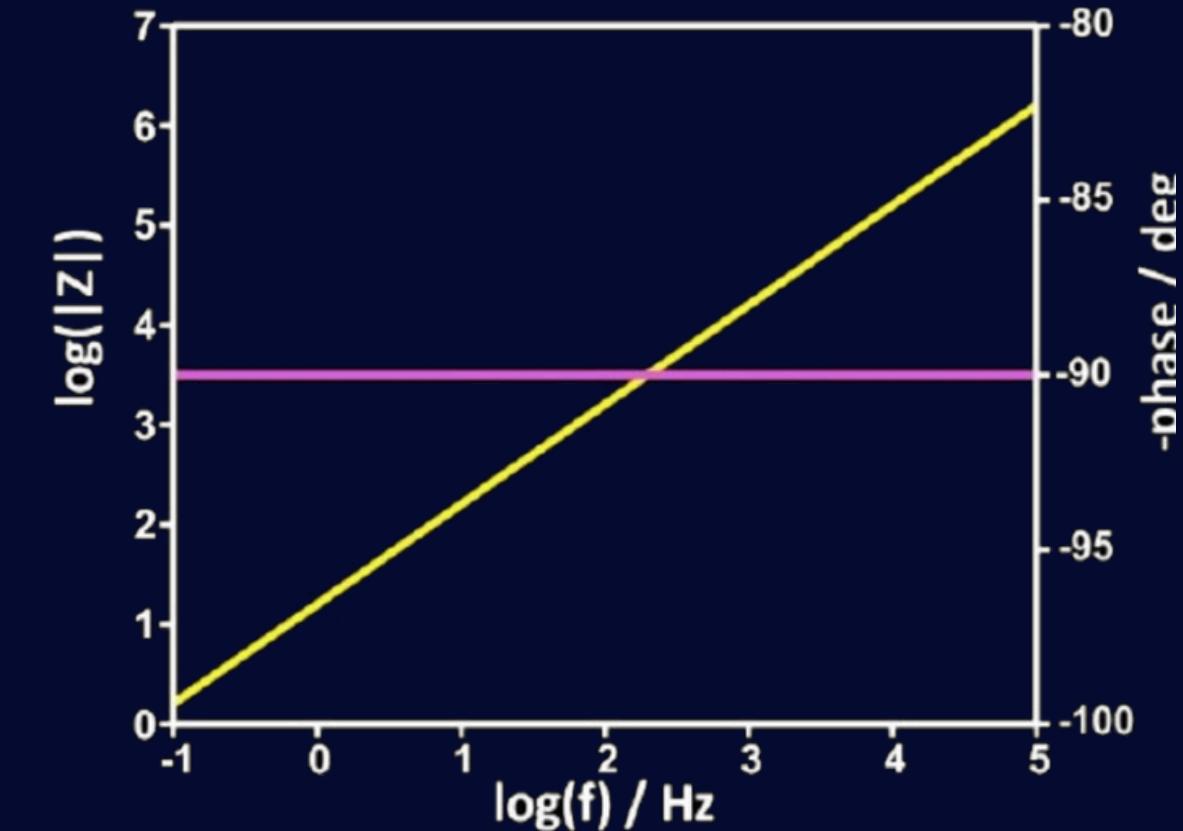
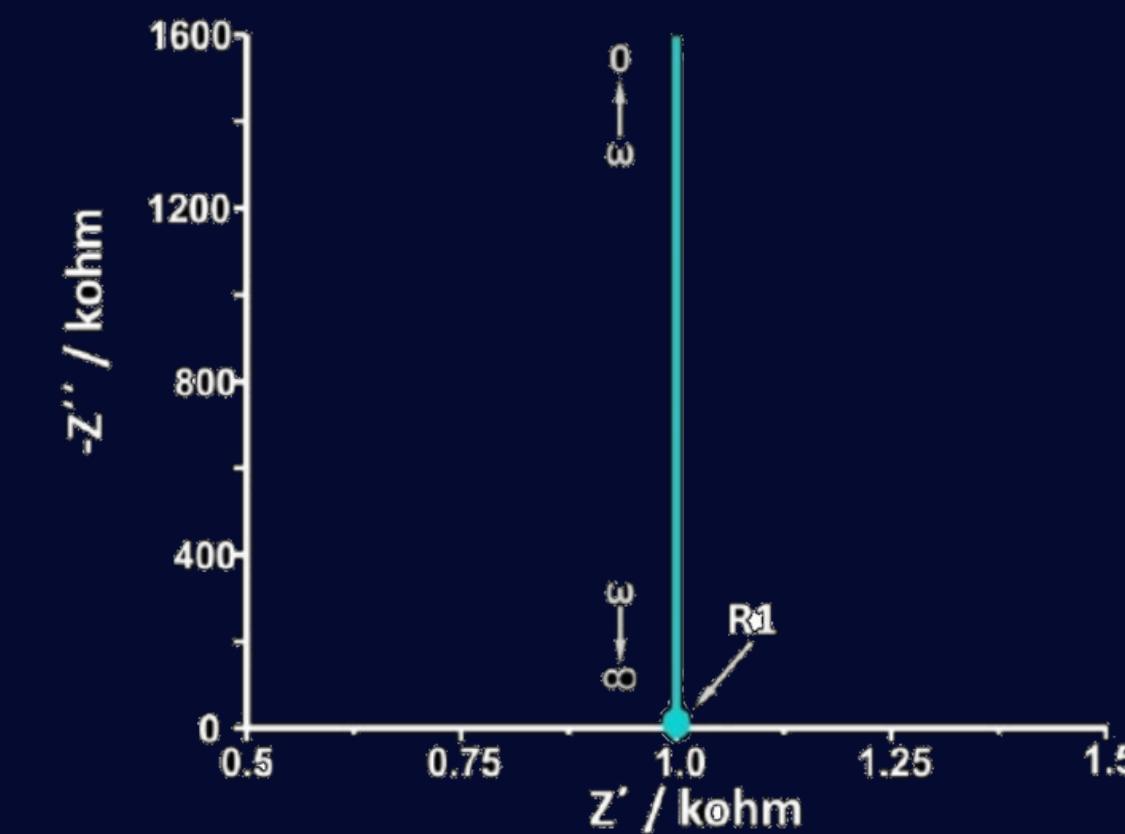
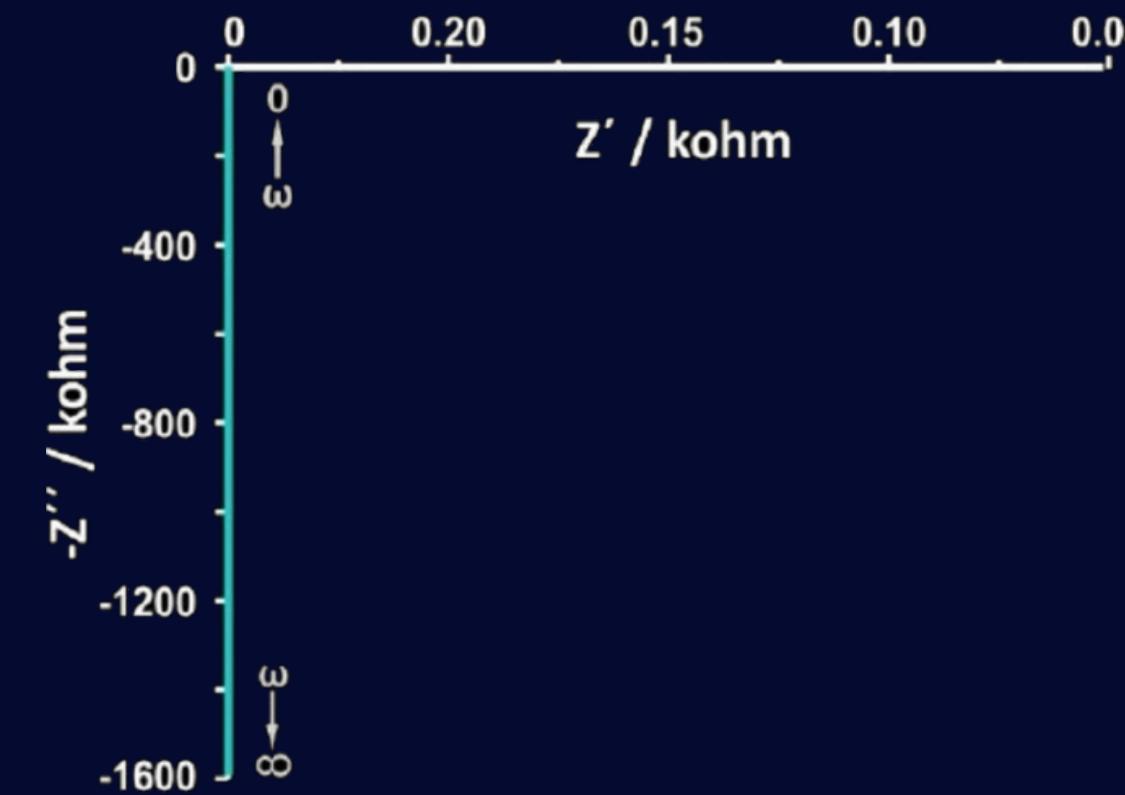
Circuiti equivalenti



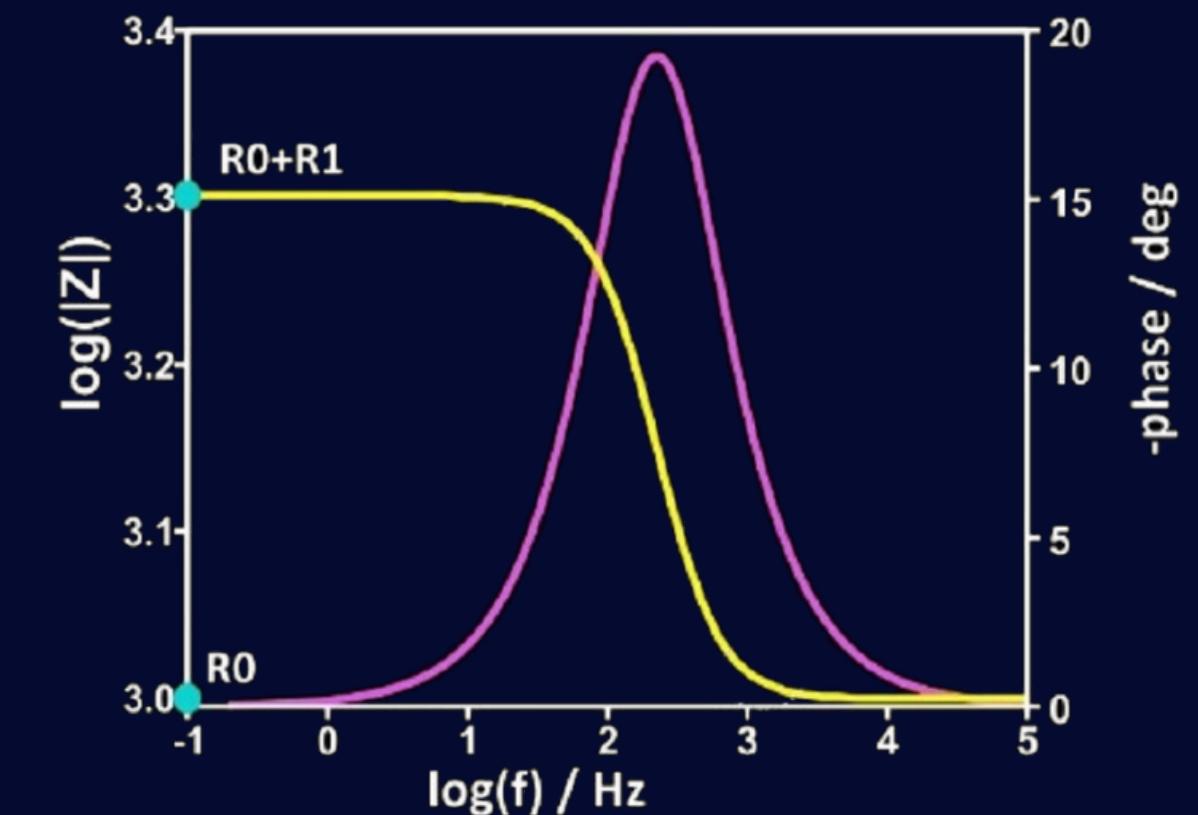
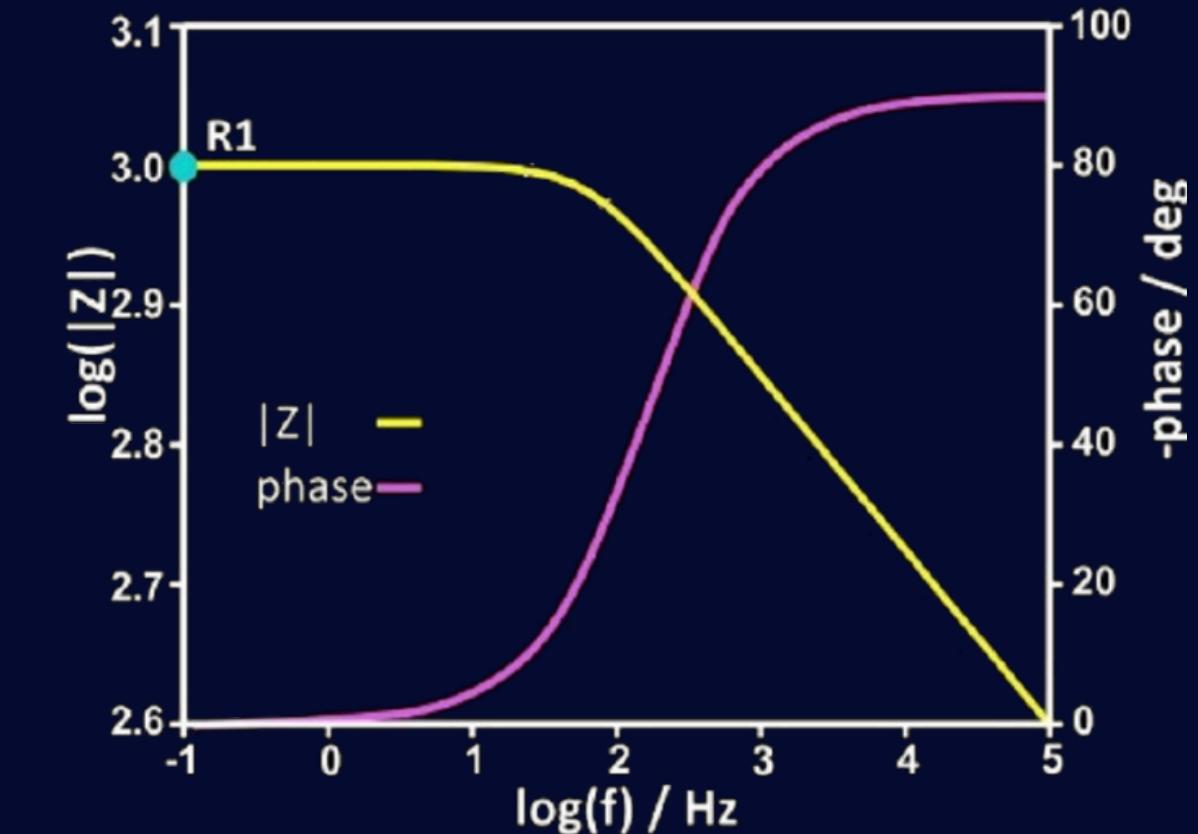
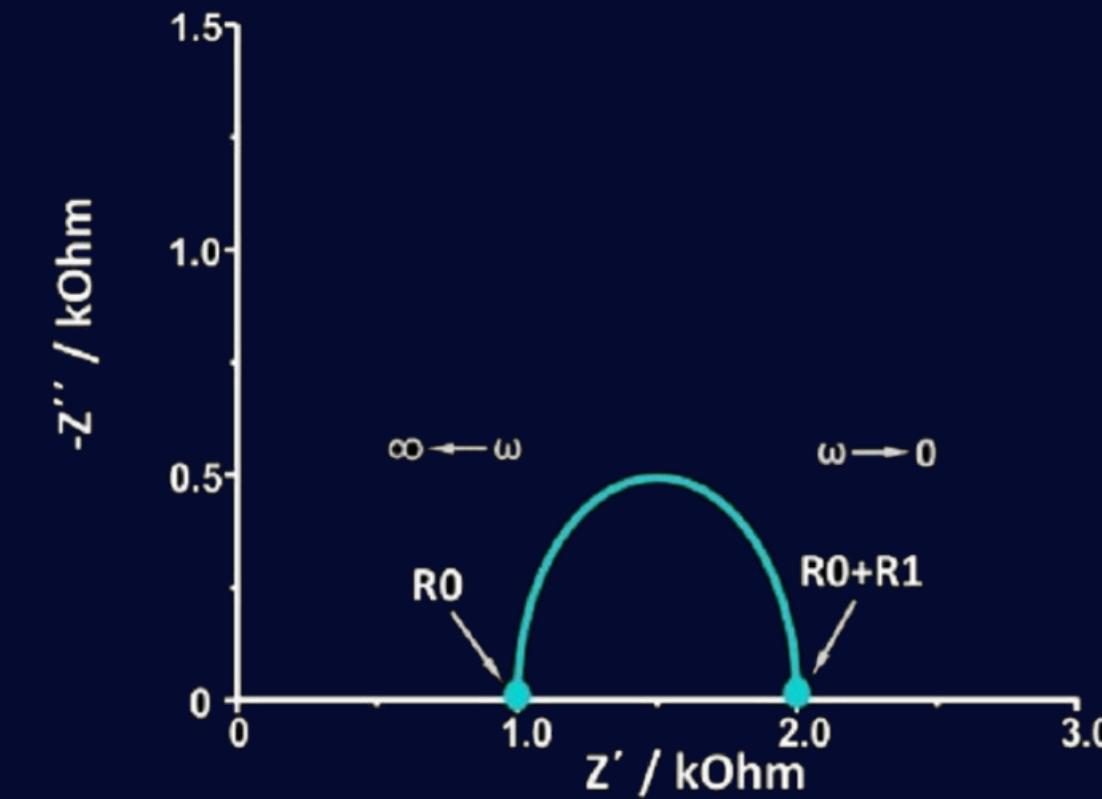
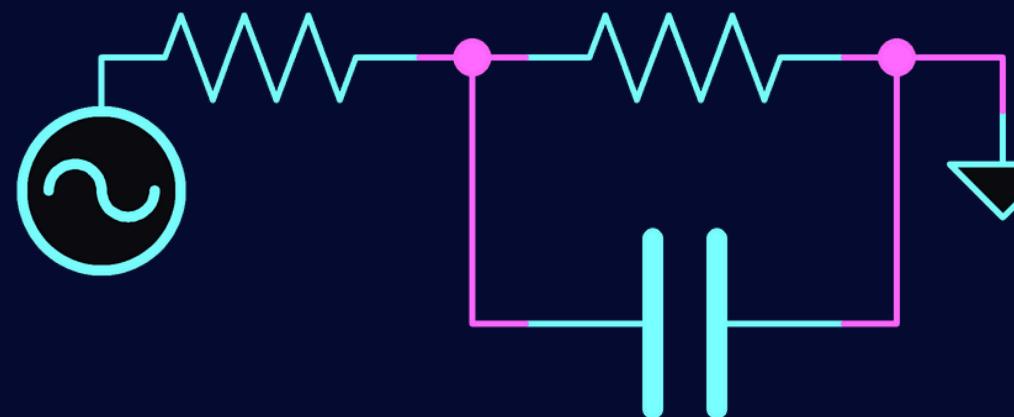
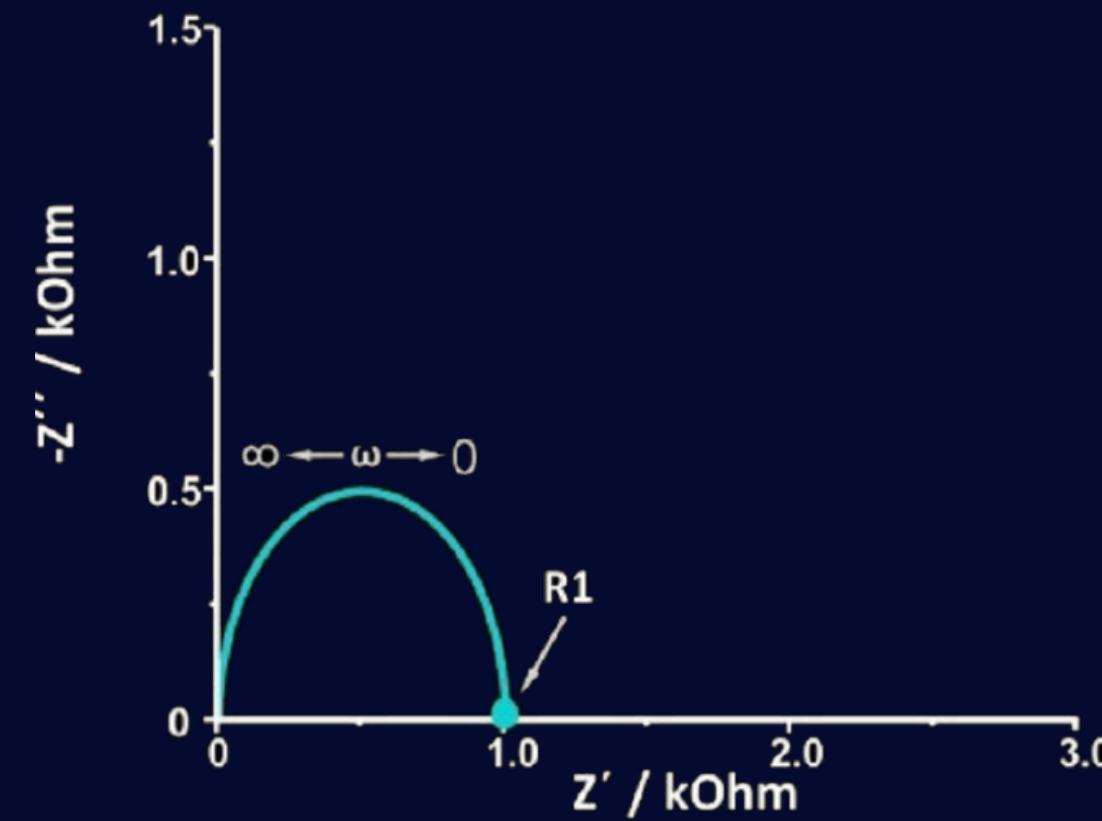
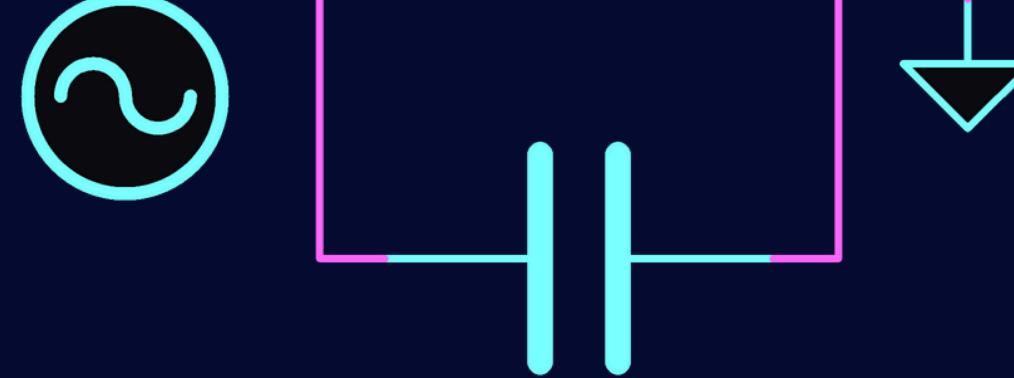
Circuiti equivalenti



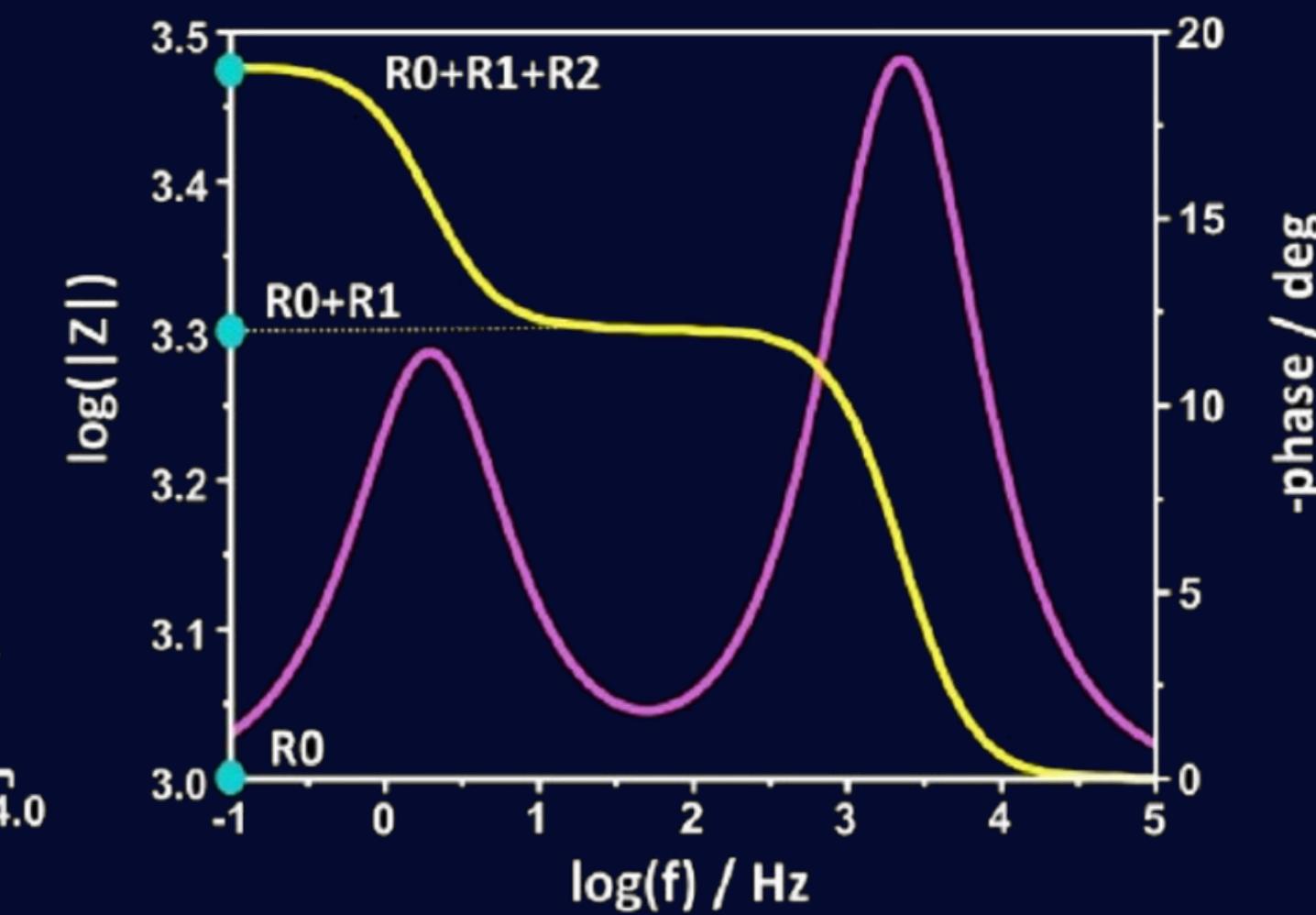
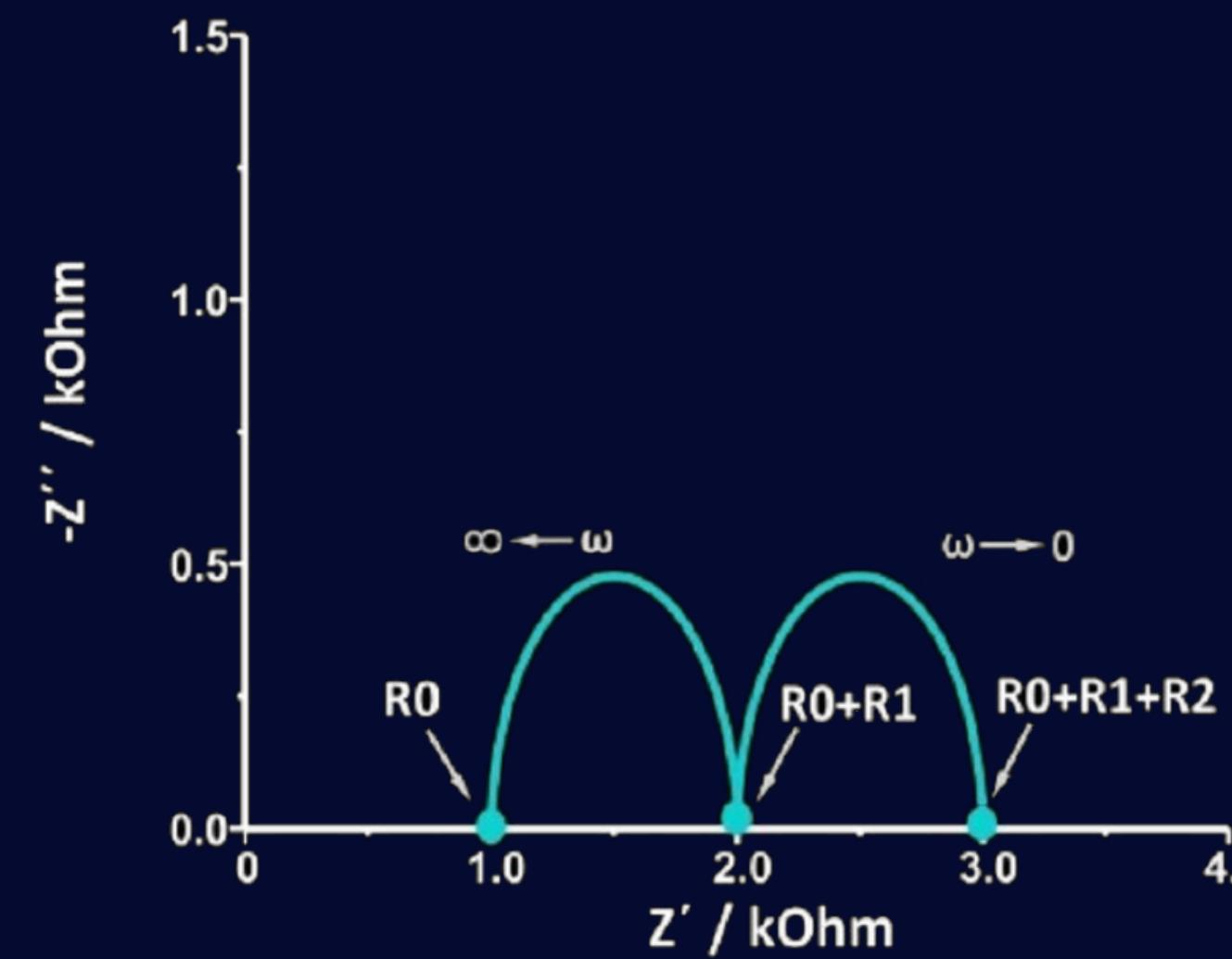
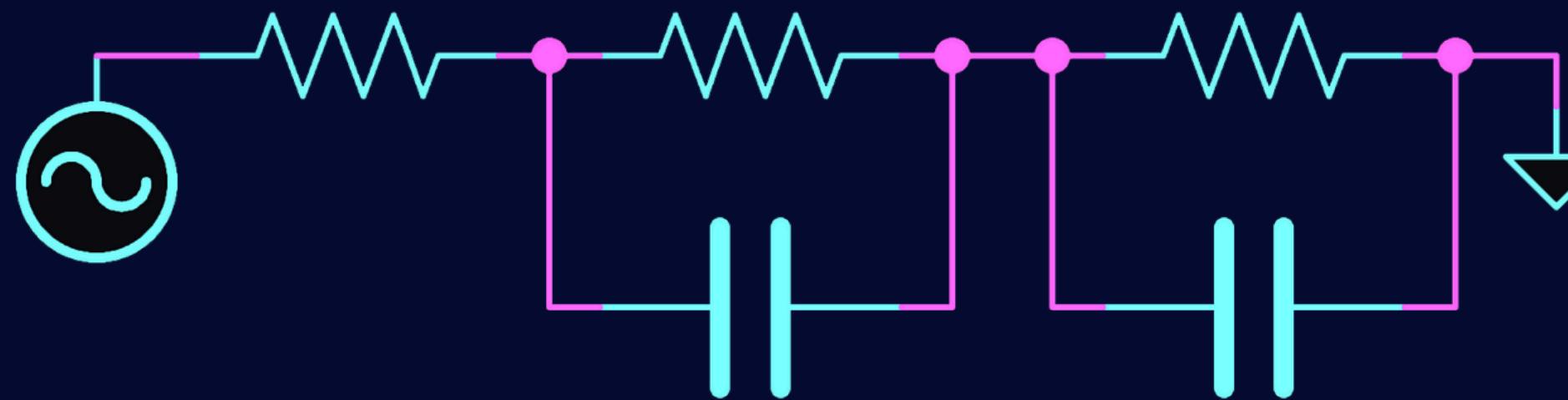
Circuiti equivalenti



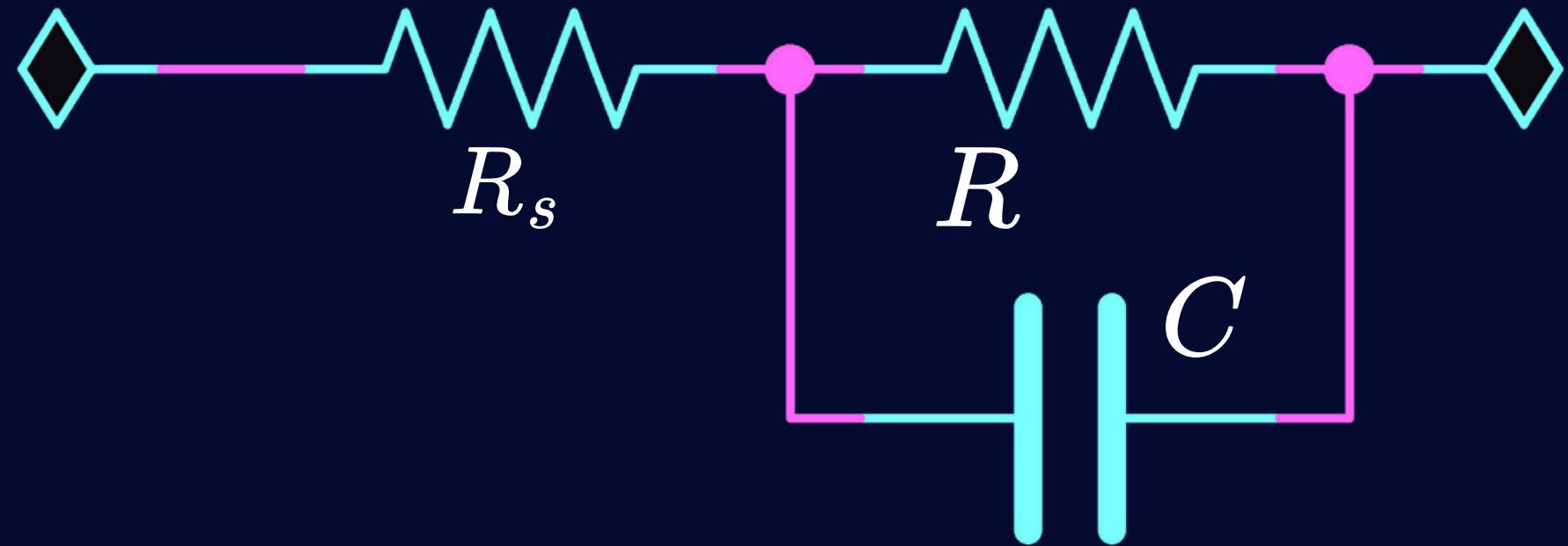
Circuiti equivalenti



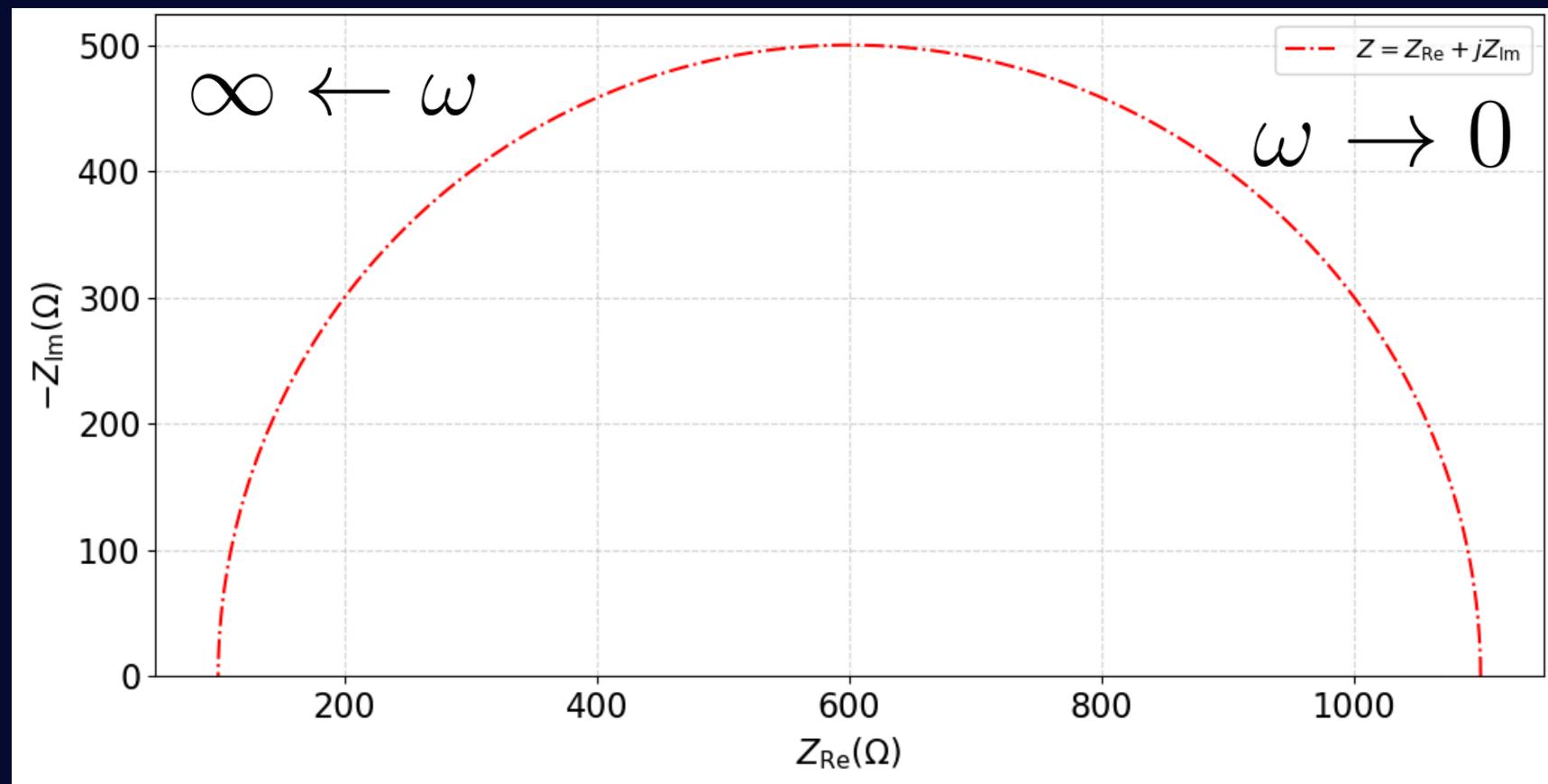
Circuiti equivalenti



Modello a singola cella RC



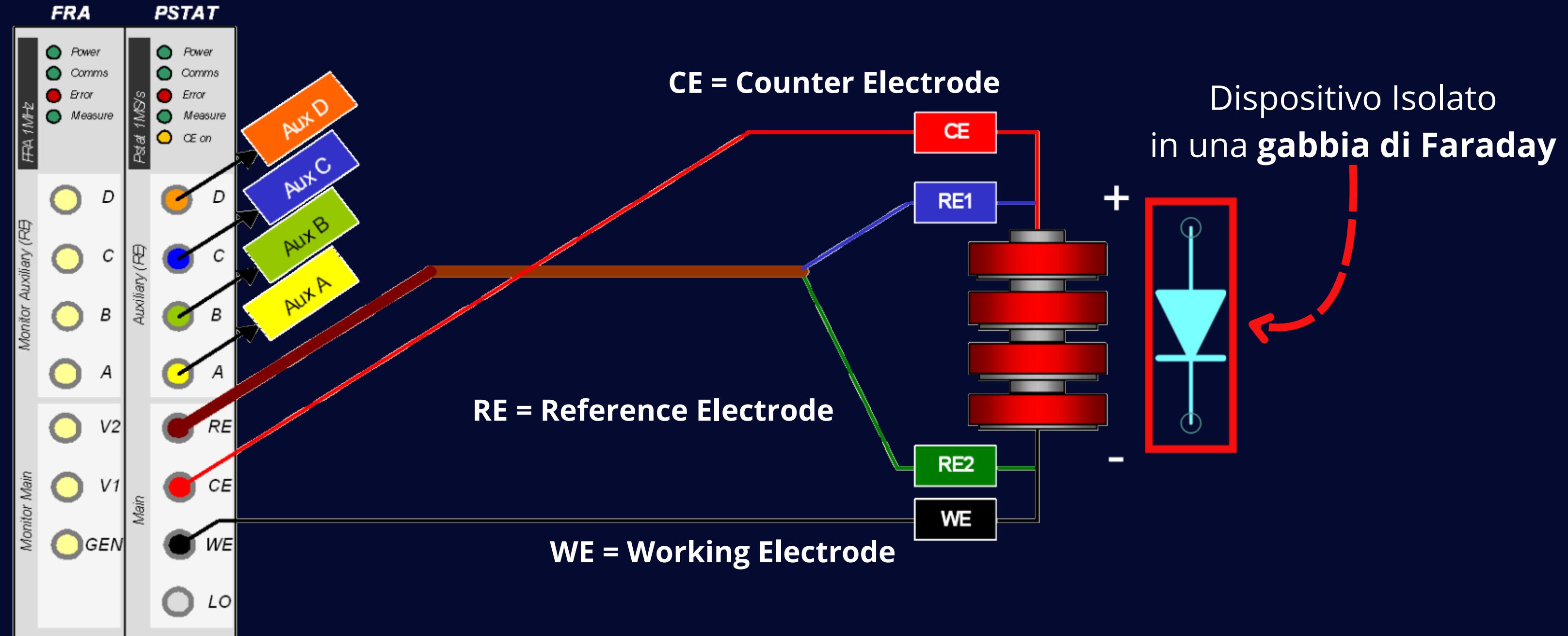
$$Z = R_s + \frac{R}{1 + (\omega RC)^2} - j \frac{\omega R^2 C}{1 + (\omega RC)^2}$$



Derivazione formula parametrica

Strumentazione

- ModuLab XM MTS - Materials Test System

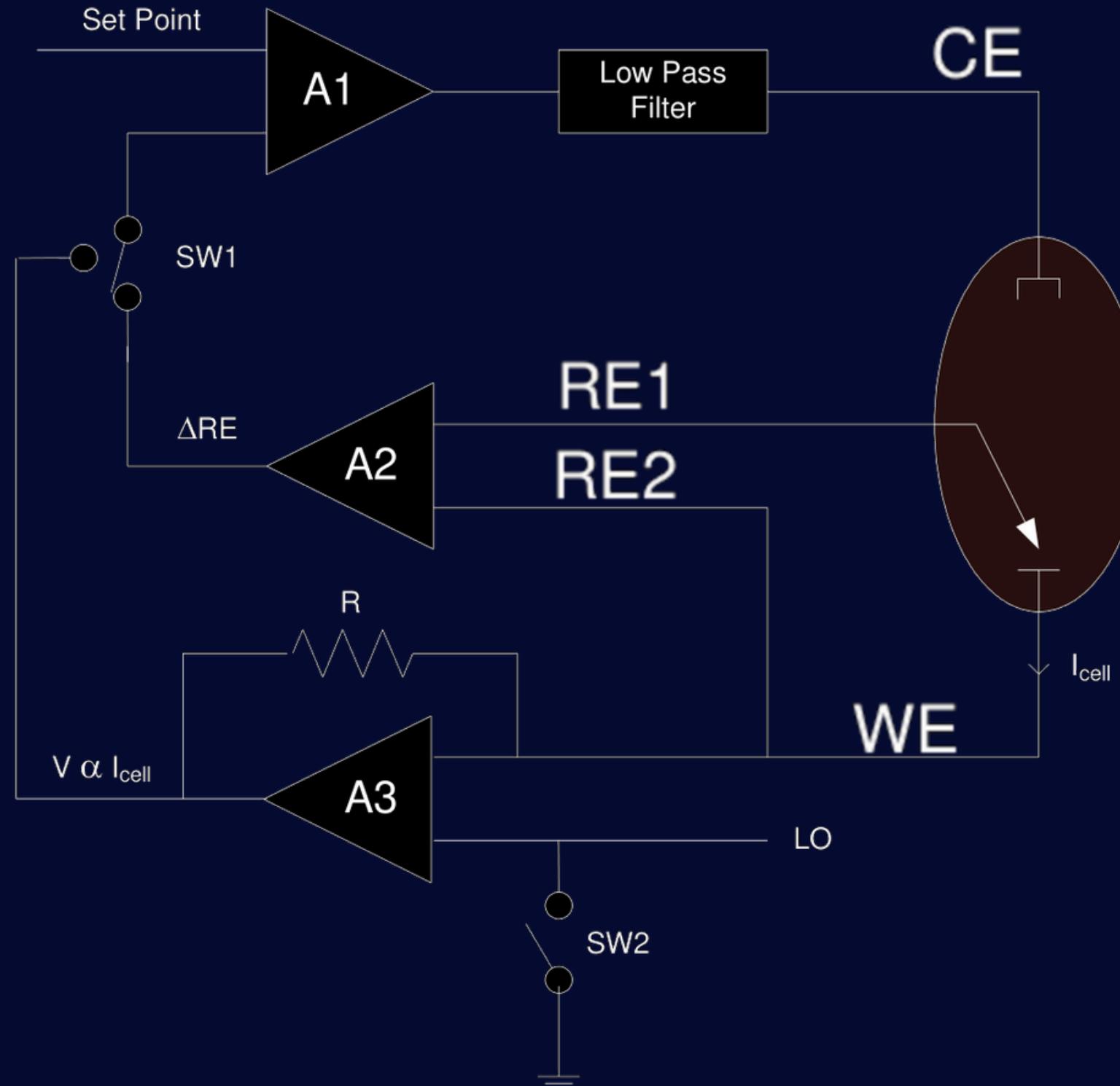


PSTAT: Potentiostat/Galvanostat

FRA: Frequency Response Analyzer

Strumentazione

- ModuLab XM MTS - Materials Test System

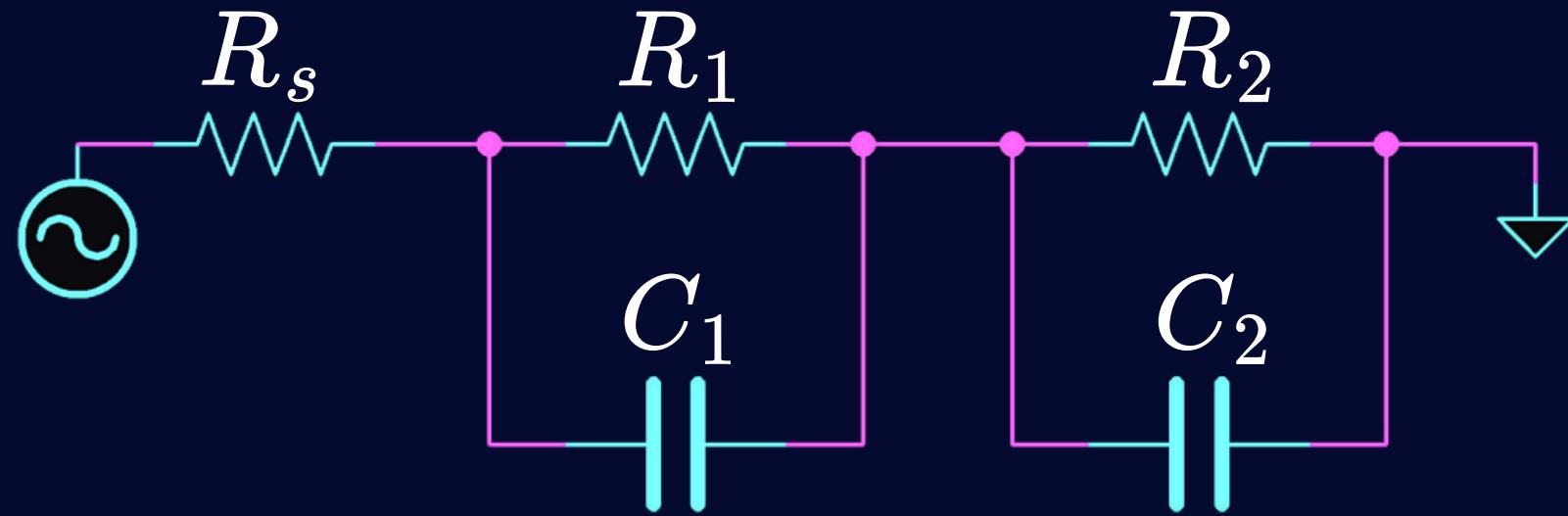


- **CE: Counter Electrode**
- **WE: Working Electrode**
- **RE: Reference Electrode**
- **LO: Riferimento di terra “*interno*”**

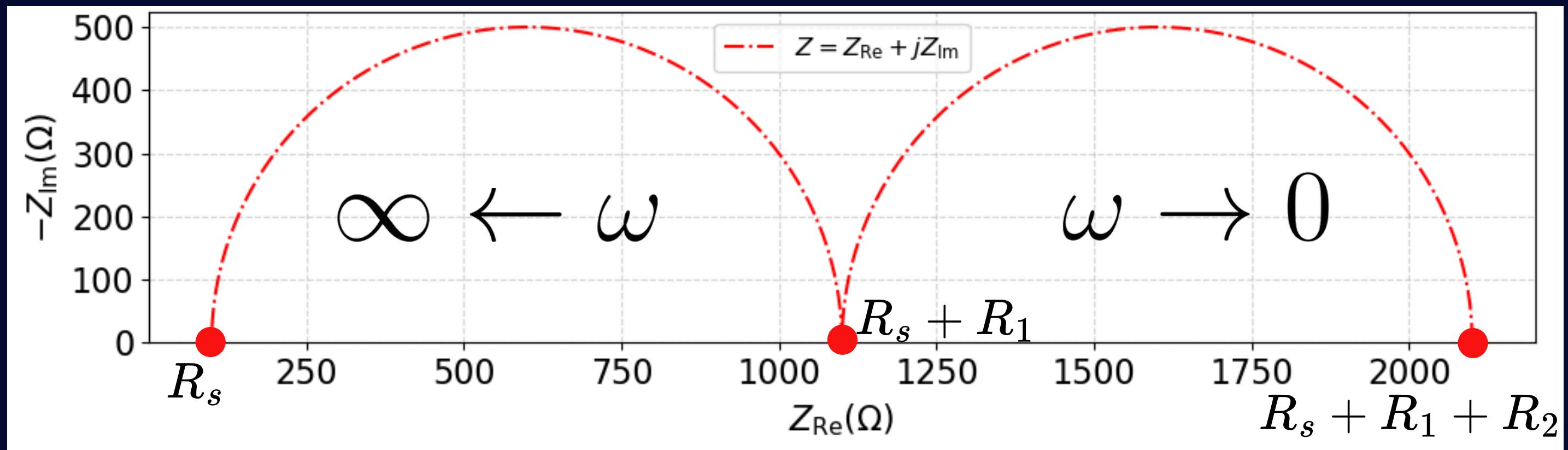
Per assicurare una tensione/corrente costante il ModuLab sfrutta un circuito di feedback.

- SW1 decide tra misure potenziostatiche o galvanostatiche
- SW2 decide quale usare tra il riferimento di terra esterno o interno

Modello a doppia cella RC



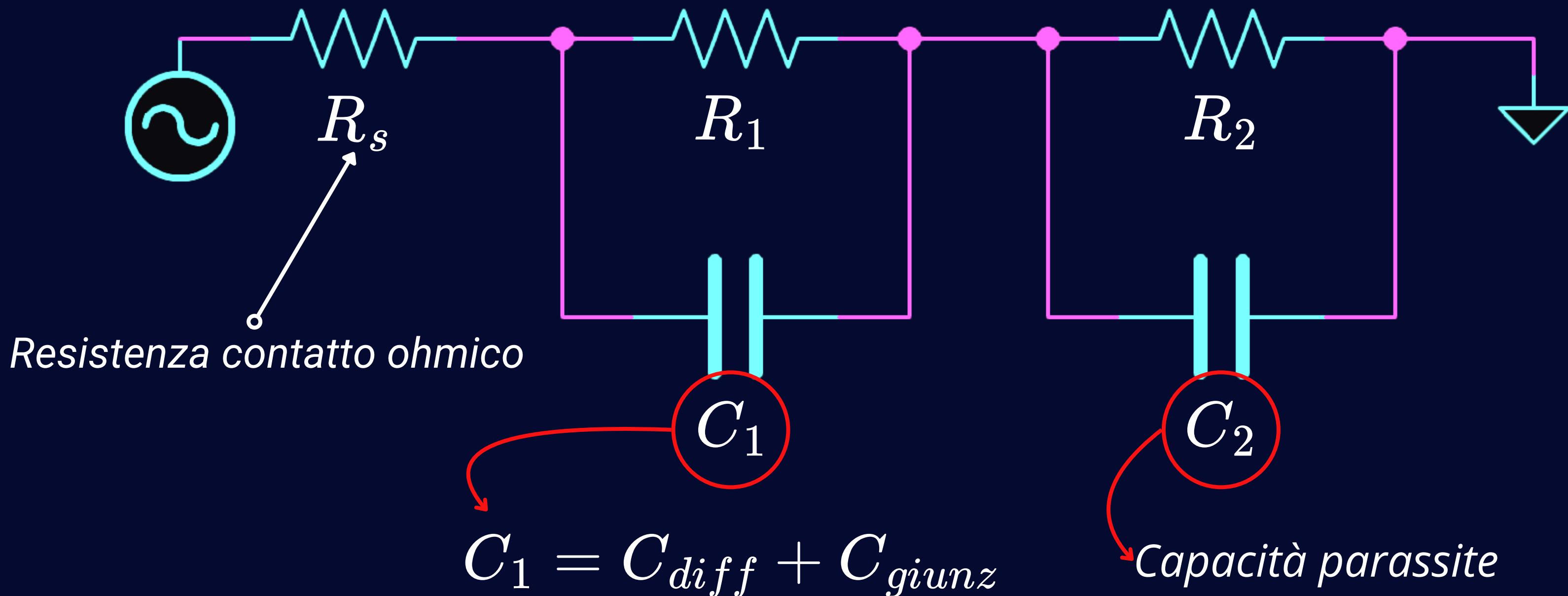
$$Z' = R_s + \frac{R_1}{1 + (\omega R_1 C_1)^2} + \frac{R_2}{1 + (\omega R_2 C_2)^2}$$
$$Z'' = \frac{\omega C_1 R_1^2}{1 + (\omega R_1 C_1)^2} + \frac{\omega C_2 R_2^2}{1 + (\omega R_2 C_2)^2}$$



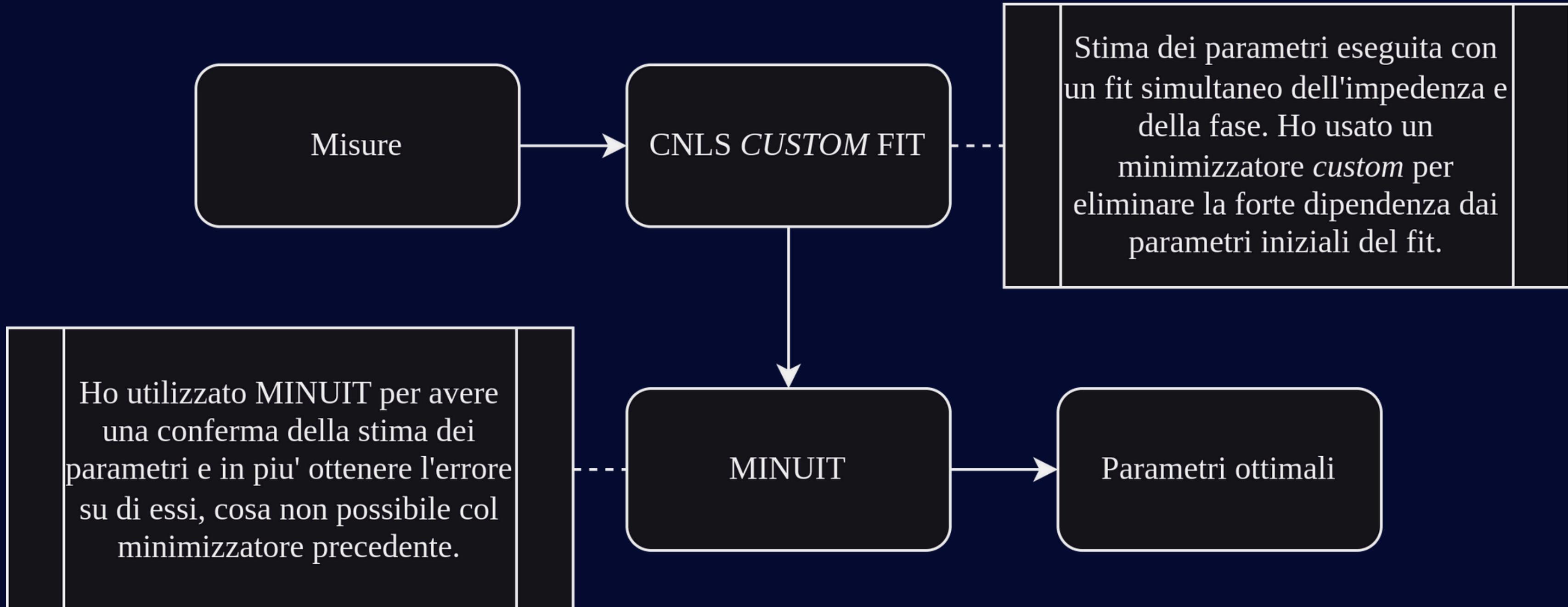
$$R_s = 100\Omega$$
$$R_1 = 1000\Omega$$
$$C_1 = 10^{-6}F$$
$$R_2 = 1000\Omega$$
$$C_2 = 10^{-12}F$$

Modello a doppia cella RC

Interpretazione dei parametri

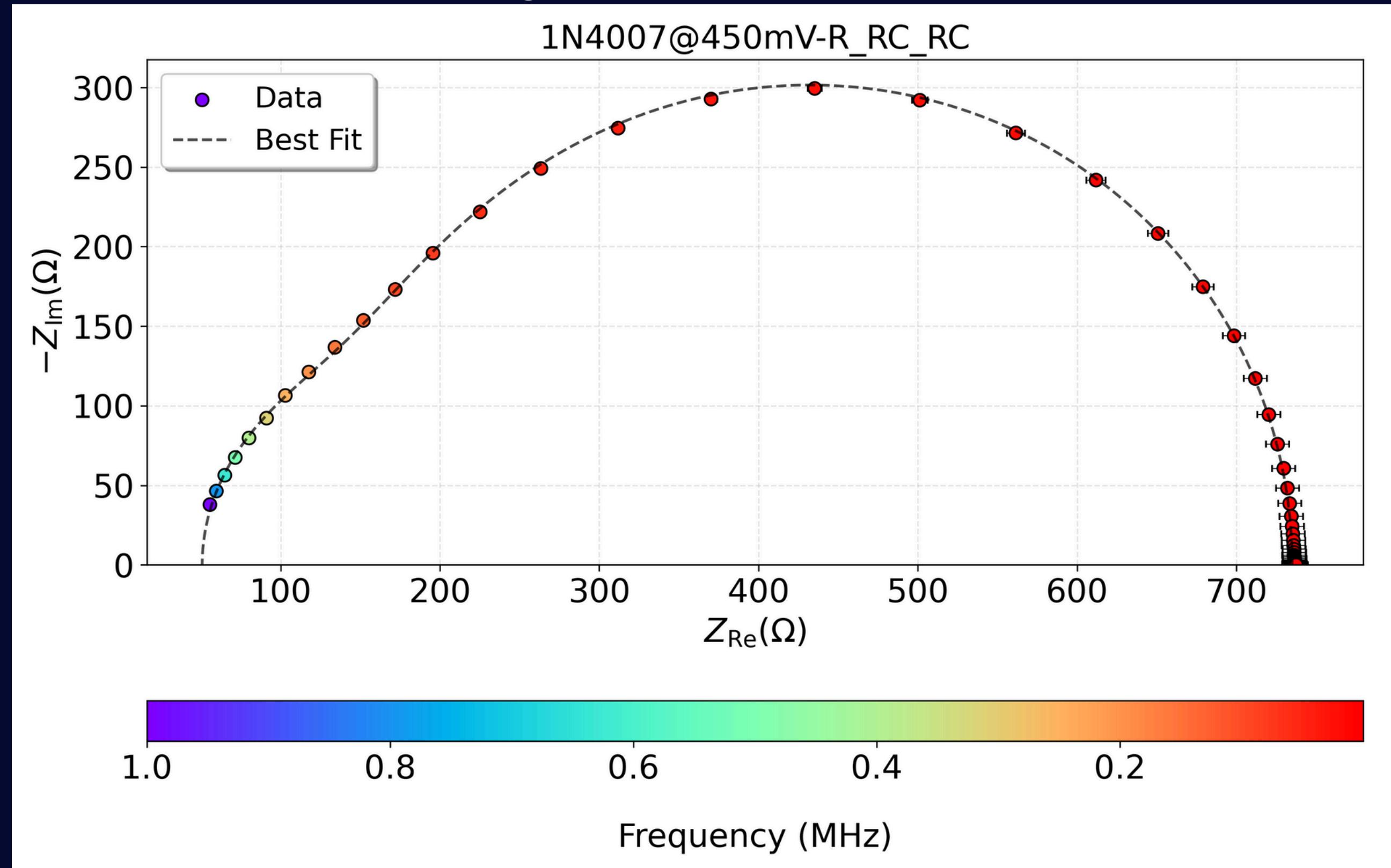


Pipeline dati per il fit

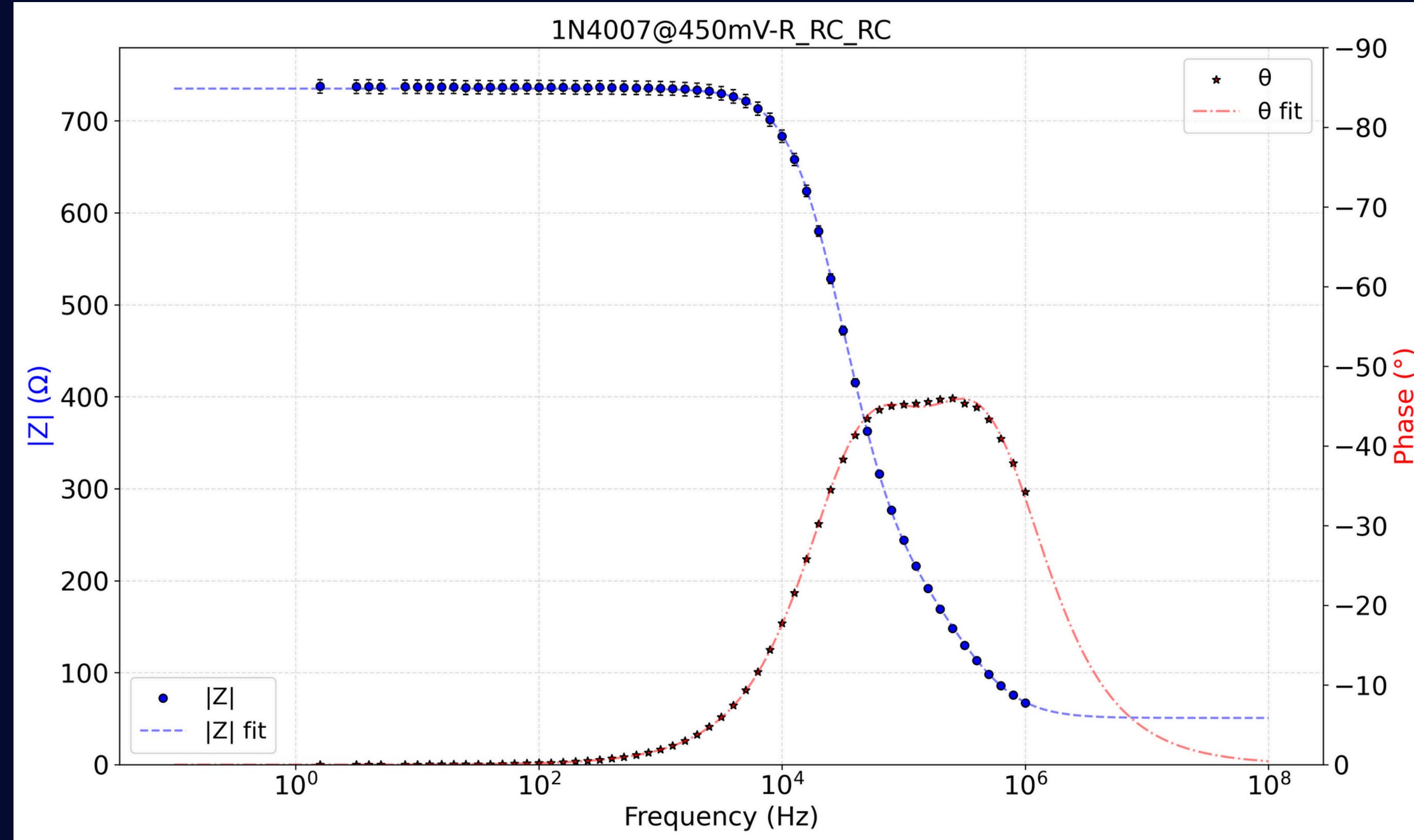


Il codice è disponibile su github.com/siegmone/bsc-thesis

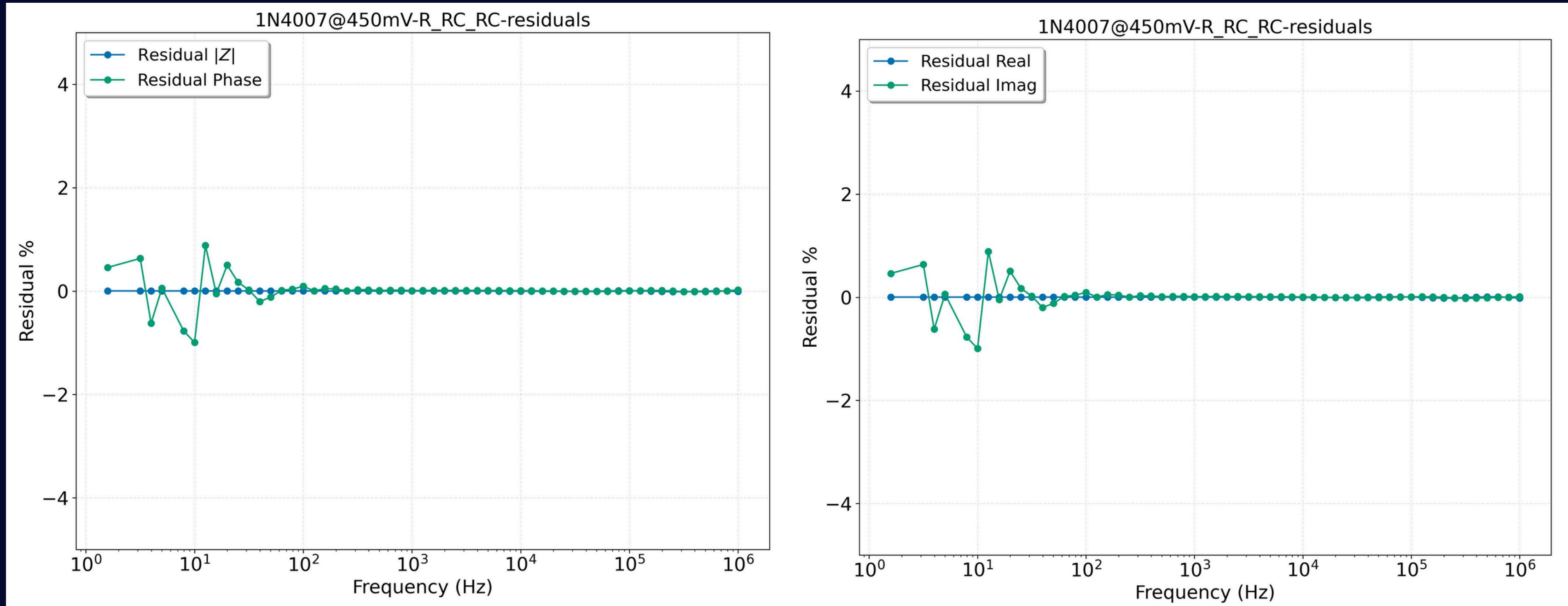
Diagramma di Nyquist



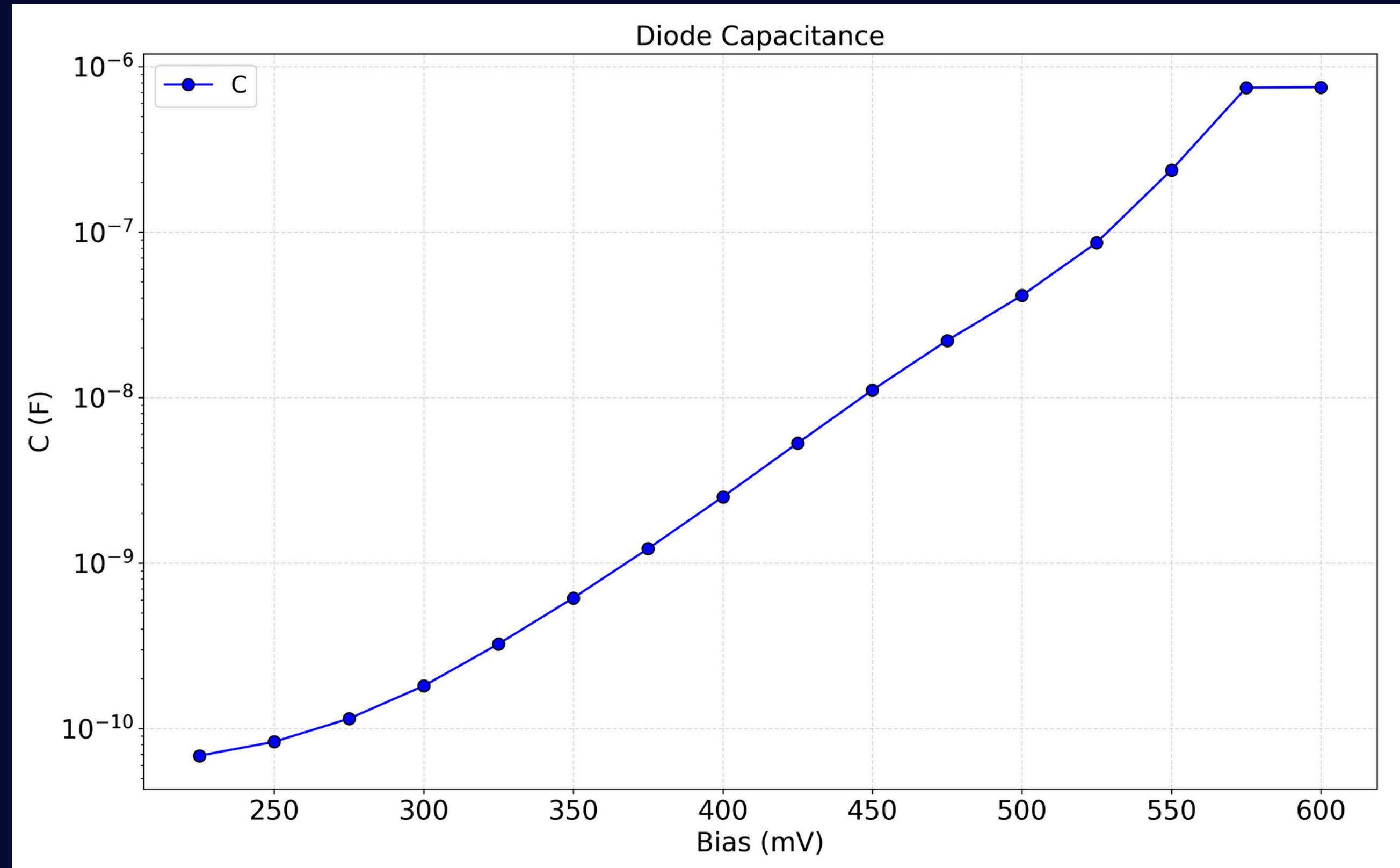
Bode Plot



Residui



Capacità vs. Bias



GRAZIE
PER
L'ATTENZIONE

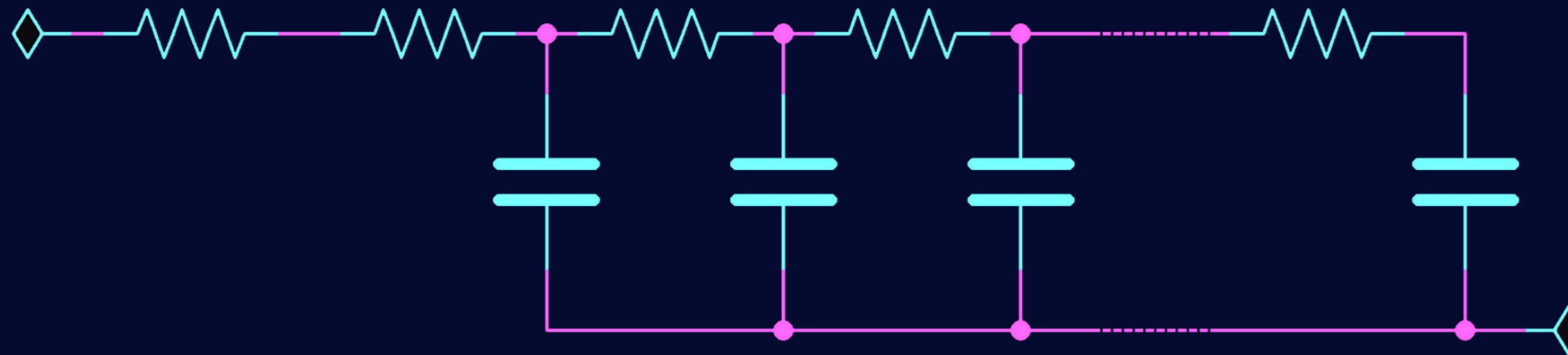
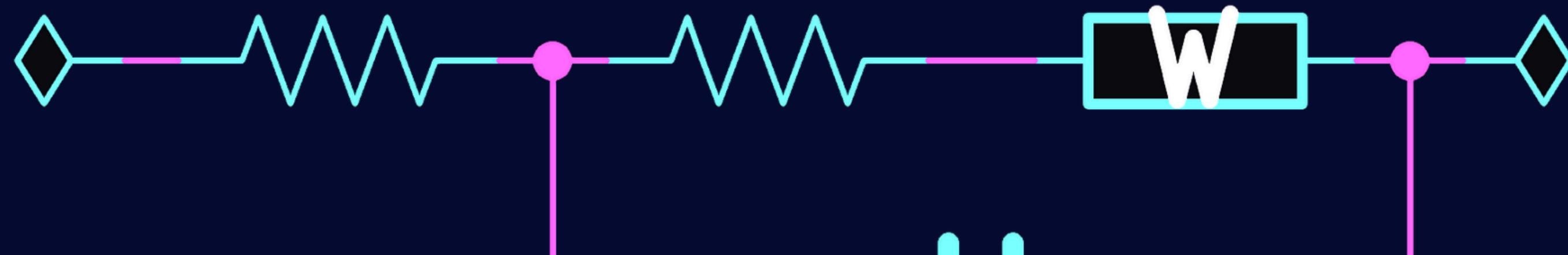
Bibliografia

Perché la spettroscopia d'impedenza?

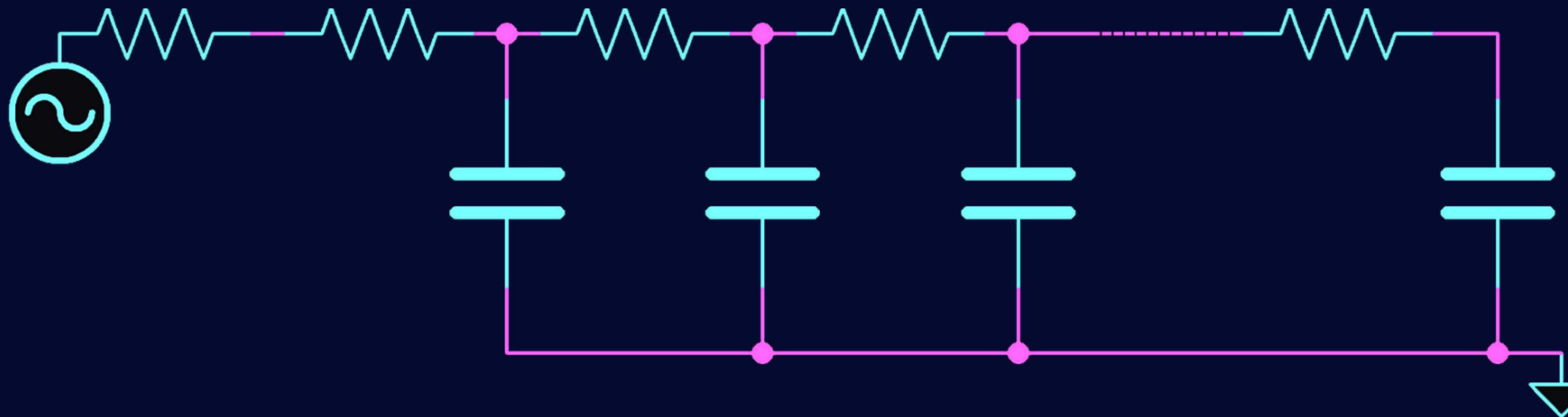
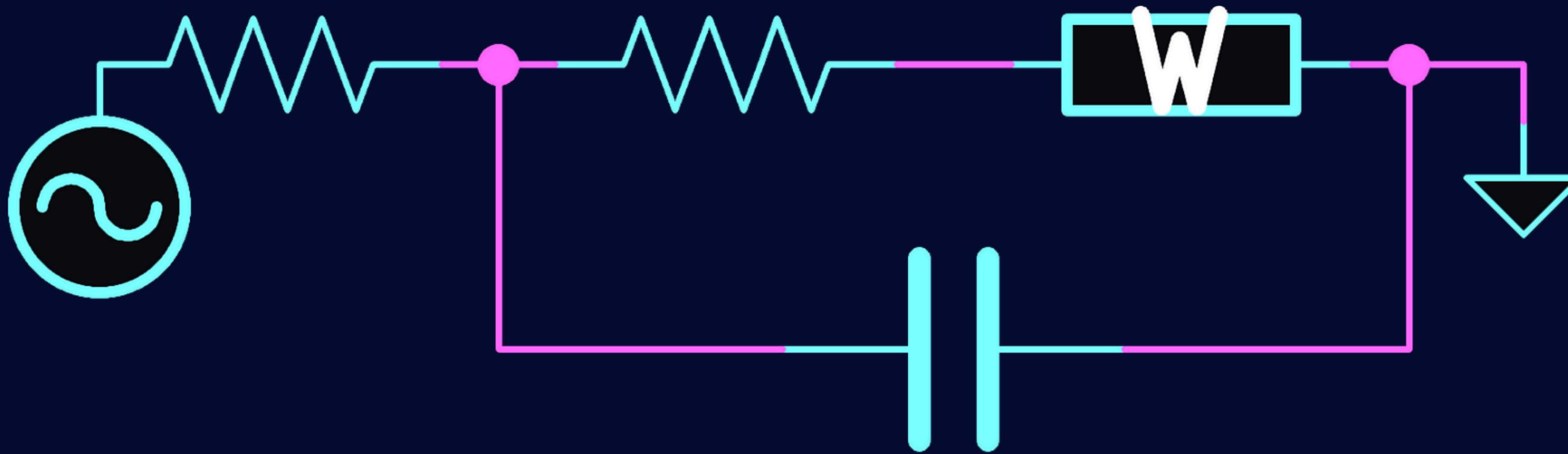
E' sicuramente possibile usare metodi piu' avanzati e precisi per caratterizzare un dispositivo a semiconduttore, ma e' molto piu' proibitivo rispetto alla spettroscopia d'impedenza.

Quindi questo metodo si rivela un'alternativa semplice e immediata per lo studio di un dispositivo.

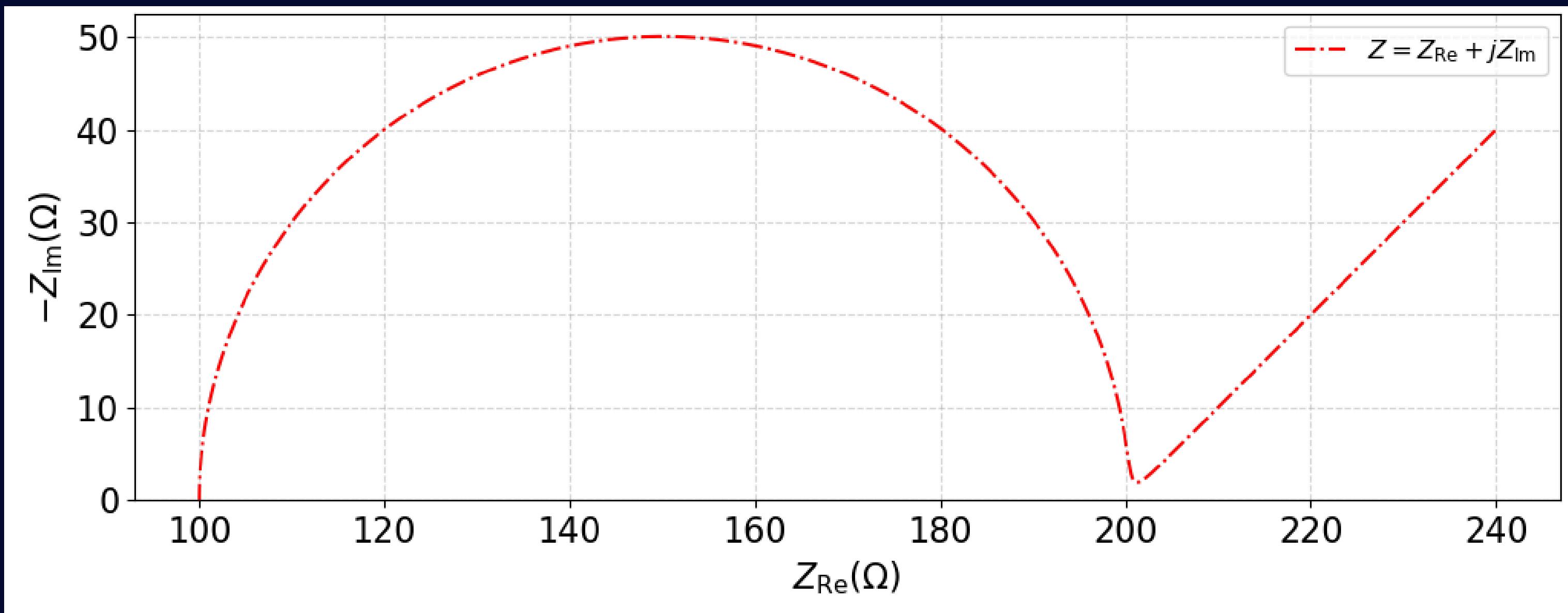
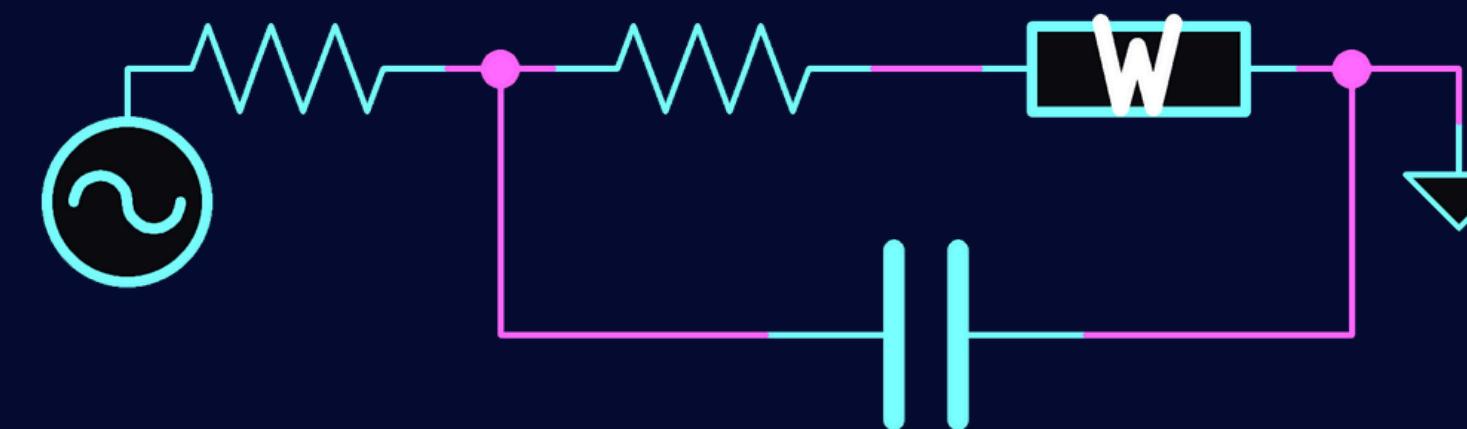
Altri Modelli



Altri Modelli



Warburg



Capacita' Parassite

In polarizzazione **diretta** entrano in gioco entrambe le *capacita' parassite* principali del diodo, ovvero di **diffusione** e di **giunzione**.

- La capacita' di **diffusione** e' dovuta all'iniezione di portatori di carica tra le zone **P** e **N** al variare della tensione.
- La capacita' di **giunzione** e' dovuta al *restringimento* della zona di svuotamento all'aumentare della tensione.

In polarizzazione **inversa** invece c'e' solo la capacita' di giunzione, causata dall'*allargamento* della zona di svuotamento al diminuire della tensione.

In generale: $C_{giunz} \ll C_{diff}$

Stima dei Parametri

- Complex Nonlinear Least Squares (CNLS) Technique

```
...  
  
def residuals(params, x, data, model):  
    y_model = model.func(params, x)  
    res_real = y_model.real - data.real  
    res_imag = y_model.imag - data.imag  
    return np.array([res_real, res_imag]).flatten()  
  
def loss(params, x, data, model):  
    return residuals(params, x, data, model)  
  
def fit_complex(params0, x, data, model):  
    params_fit = least_squares(  
        loss, params0, bounds=(0, np.inf), args=(x, data, model),  
        method='trf', loss='linear', ftol=1e-8, gtol=1e-8, xtol=1e-8,  
    )  
    data_fit = model.func(params_fit.x, x)  
    return params_fit, data_fit
```

Il fit dipende fortemente dal *guess* iniziale.

Per eliminare questa dipendenza, il fit viene ripetuto più volte partendo dai nuovi parametri + un rumore random per evitare i minimi locali

```
...  
  
def best_fit_complex(x, data, model, convergence_threshold=100, sigma=0.1):  
    np.random.seed(42)  
    params0 = [1 for _ in range(model.params_num)]  
    best_params_fit, best_data_fit = fit_complex_phase(params0, x, data, model)  
    best_cost = np.inf  
    convergence_counter = 0  
    while convergence_counter < convergence_threshold:  
        params_fit, data_fit = fit_complex(params0, x, data, model)  
        params0 = params_fit.x * \  
            (1 + np.random.uniform(low=-sigma, high=sigma, size=model.params_num))  
        if params_fit.cost < best_cost:  
            best_cost = params_fit.cost  
            best_params_fit = params_fit  
            best_data_fit = data_fit  
            convergence_counter = 0  
        else:  
            convergence_counter += 1  
    return best_params_fit, best_data_fit
```