

UNIVERSITÀ
POLITECNICA
DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Elettronica e delle Tecnologie Digitali

Studio ed implementazione dei principali metodi per la misura dei parametri LCR

Study and implementation of the main methods for measuring LCR parameters

Tesi di Laurea di:
Thomas CARBONARI

Relatore:
Prof.ssa **Susanna SPINSANTE**

Correlatore:
Dott. Ing. **Gianluca CIATTAGLIA**

Anno Accademico 2025/2026

Abstract

La misura dei parametri elettrici R, C e L riveste un ruolo fondamentale nella caratterizzazione dei componenti e dei circuiti elettronici. Il presente lavoro analizza i principali metodi impiegati per la misura di tali parametri, con particolare attenzione alle tecniche basate su ponte, sulla risposta transitoria e sul metodo voltamperometrico.
Nella prima parte della tesi vengono analizzati i modelli equivalenti dei componenti reali e i principali metodi di misura dei parametri resistivi, capacitivi e induttivi. Viene inoltre condotto un confronto sperimentale tra il misuratore LCX200 di Rohde & Schwarz e la piattaforma Moku:Go di Liquid Instruments, attraverso una campagna di misure effettuata su differenti componenti e intervalli di valore, al fine di valutarne e confrontarne le prestazioni.
Successivamente viene sviluppato e caratterizzato un sistema di misura basato su una scheda Arduino Uno e, in particolare, sul microcontrollore ATmega328P integrato nella scheda, finalizzato alla realizzazione di un RCL meter. Il sistema viene sottoposto a diverse prove sperimentali per la misura di resistenze e capacità isolate e di reti RC serie e parallelo, valutandone le prestazioni mediante il confronto con valori di riferimento.
Nella parte conclusiva viene realizzato un ohmetro basato sul metodo a ponte, utilizzando Arduino Uno per l'acquisizione delle tensioni mediante il convertitore analogico-digitale integrato e uno stadio buffer. Il sistema viene caratterizzato attraverso diverse portate, mediante una campagna di misure su un ampio intervallo di valori resistivi. I risultati vengono confrontati con quelli ottenuti mediante il metodo precedentemente implementato e con i valori di riferimento misurati tramite l'LCX200. L'analisi dell'incertezza evidenzia un buon comportamento del metodo a ponte per valori di resistenza medio-alti, mentre per le basse resistenze si osserva un significativo aumento dell'errore relativo. Tale comportamento mette in evidenza la maggiore dipendenza del metodo a ponte dalle caratteristiche dei componenti impiegati e dalle condizioni operative del circuito, in particolare per quanto riguarda gli effetti termici e le resistenze parassite.

Abstract

The measurement of the electrical parameters R, C, and L plays a fundamental role in the characterization of electronic components and circuits. This work analyzes the main methods employed for the measurement of these parameters, with particular focus on bridge-based techniques, transient response methods, and the voltammetric method.

The first part of the thesis examines the equivalent models of real components and the main methods used for the measurement of resistive, capacitive, and inductive parameters. An experimental comparison is also carried out between the Rohde & Schwarz LCX200 meter and the Moku:Go platform by Liquid Instruments, through a measurement campaign performed on different components and over various value ranges, with the aim of evaluating and comparing their performance.

Subsequently, a measurement system based on an Arduino Uno board, and specifically on the ATmega328P microcontroller integrated into the board, is developed and characterized for the implementation of an RCL meter. The system is subjected to several experimental tests for the measurement of individual resistors and capacitors, as well as series and parallel RC networks. Its performance is evaluated by comparing the measured values with reference values.

In the final part of the thesis, an ohmmeter based on the bridge method is developed, using an Arduino Uno for voltage acquisition through its integrated analog-to-digital converter and a buffer stage. The system is characterized over several measurement ranges through a measurement campaign covering a wide range of resistance values. The results are compared with those obtained using the previously implemented method and with reference values measured using the LCX200. The uncertainty analysis highlights the good performance of the bridge method for medium-to-high resistance values, whereas a significant increase in relative error is observed for low resistance values. This behavior highlights the greater dependence of the bridge method on the characteristics of the components used and on the operating conditions of the circuit, particularly with regard to thermal effects and parasitic resistances.

Indice

1 Introduzione **1**

2 Stato dell'Arte **4**
2.1 Misura dei parametri *R, C* e *L* dei componenti 4
2.2 Modelli equivalenti associati ai componenti reali 6
2.2.1 Resistore Reale 6
2.2.2 Condensatore Reale 8
2.2.3 Induttore Reale 10
2.3 Metodi per la misura dei parametri *R, C* e *L* 12
2.3.1 Metodo a ponte 12
2.3.2 Metodo della risonanza 13
2.3.3 Metodo della risposta transitoria 14
2.3.4 Metodo voltamperometrico 16
2.3.5 Metodo a ponte autobilanciante 17
2.4 Strumenti per la misura dei parametri *R, C* e *L* 18
2.4.1 *LCR* meter 19
2.4.2 Analizzatori d'impedenza 20
2.4.3 Analizzatori di reti vettoriali 21

3 Confronto tra strumenti **22**
3.1 Banco di misura basato su LCX200 di R&S 23
3.1.1 Procedura di misura 23
3.2 Frequency Response Analyzer del Moku:GO M2 29
3.2.1 Procedura di misura 30
3.3 Analisi dei risultati sperimentali ottenuti con i due banchi di misura . 43

4 Implementazione di un *RCL* meter su ATMega328P **57**
4.1 Descrizione del metodo 57
4.1.1 Misura di una resistenza isolata 59
4.1.2 Misura di una capacità isolata 60
4.1.3 Caratterizzazione di una rete *RC* serie 61
4.1.4 Caratterizzazione di una rete *RC* parallelo 62
4.2 Implementazione dei metodi di misura e analisi dei risultati sperimentali 64
4.2.1 Resistenza isolata e taratura dei parametri resistivi. 64

4.2.2 Capacità isolata 71
4.2.3 Rete RC serie 78
4.2.4 Rete RC parallelo 85
4.3 Misura di una resistenza isolata mediante metodo a ponte con Arduino Uno 93
4.4 Confronto tra i due metodi per la misura di resistenza isolata 105

5 Conclusioni **110**

Bibliografia **112**

Capitolo 1

Introduzione

Nel corso dell'ultimo secolo l'elettronica ha rivoluzionato profondamente ogni aspetto della società, accompagnando l'uomo verso un progresso tecnologico e scientifico senza precedenti. Per far fronte alle nuove esigenze, si è assistito a una rapida diffusione di dispositivi intelligenti, sistemi di comunicazione avanzati e infrastrutture innovative, con un conseguente incremento della domanda di circuiti elettronici caratterizzati da stringenti requisiti in termini di prestazioni, affidabilità ed efficienza.
Secondo la classificazione riportata dall'IEEE [1], i componenti impiegati nella realizzazione di circuiti possono essere distinti in due categorie fondamentali: componenti attivi e componenti passivi. Alla prima categoria appartengono i dispositivi in grado di amplificare, generare o controllare segnali elettrici, il cui normale funzionamento richiede una fonte di alimentazione esterna.
Alla seconda categoria invece appartengono i componenti passivi, i quali non forniscono un guadagno di potenza, ma svolgono funzioni altrettanto rilevanti grazie alla loro capacità di immagazzinare o dissipare energia.

Tra i componenti passivi si distinguono resistori, condensatori ed induttori, i quali sono presenti nella quasi totalità dei circuiti elettronici.
In fase di progetto essi vengono descritti mediante modelli ideali, nei quali ciascun componente è caratterizzato esclusivamente dal rispettivo parametro fondamentale: la resistenza per i resistori, la capacità per i condensatori e l'induttanza per gli indut-

tori. Con il passaggio dalla progettazione teorica alla realizzazione fisica del circuito, tuttavia, i modelli ideali non sono più sufficienti ad una descrizione accurata del comportamento dei componenti. Nel processo produttivo le tolleranze costruttive, le proprietà dei materiali impiegati e gli effetti parassiti provocano la comparsa di fenomeni di non idealità che alterano le caratteristiche elettriche dei componenti [2], introducendo una dipendenza del loro comportamento dalla frequenza dei segnali applicati e dalle condizioni operative, quali temperatura e umidità. Una descrizione accurata del comportamento reale dei componenti risulta indispensabile nella progettazione di sistemi elettronici destinati ad applicazioni ad elevata affidabilità o precisione. Numerosi settori come il biomedicale, l'aerospaziale, le telecomunicazioni e il militare necessitano di circuiti capaci di garantire prestazioni prevedibili anche in condizioni operative non particolarmente favorevoli.

Per tenere conto delle non idealità introdotte dai componenti fisici, questi vengono rappresentati attraverso dei modelli equivalenti nei quali, oltre al parametro fondamentale, vengono introdotti elementi parassiti con lo scopo di descriverne con precisione il comportamento reale [3]. Il ricorso a questi modelli richiede la determinazione sperimentale dei parametri caratteristici della rappresentazione scelta, evidenziando l'importanza dei metodi di misura e degli strumenti in grado di stimarne i valori con elevata accuratezza e ripetibilità.

Alla luce di queste considerazioni, la presente tesi si pone l'obiettivo di analizzare differenti metodologie di misura finalizzate alla determinazione di tali parametri e alla valutazione del comportamento in frequenza dei rispettivi componenti. A tale scopo, sono state confrontate misurazioni effettuate con strumenti caratterizzati da diversi principi di funzionamento: un LCX200, strumento RCL di riferimento, di Rohde & Schwarz ed un Moku:GO, configurato come Frequency Analyzer. In seguito, è stata esaminata una soluzione a basso costo basata sull'utilizzo di una scheda Arduino Uno, sfruttandone il convertitore analogico-digitale (ADC) integrato nel microcontrollore ATmega328P, per stimare stimare i parametri RLC analizzando i transitori di carica e scarica di circuiti RC e RL. Infine, è stata analizzata una metodologia alternativa per la sola stima della resistenza, adottando una misura a ponte in regime continuo, così

da poterne confrontare i risultati con quelli ottenuti mediante il metodo a basso costo.

La presente tesi è articolata come segue. Il Capitolo 2 presenta lo stato dell'arte e una rassegna della letteratura scientifica relativa ai differenti metodi e strumenti impiegati per la misura e la caratterizzazione dei parametri R, L e C.
Il Capitolo 3 descrive l'implementazione del banco di misura per il confronto tra lo strumento LCX200 e il Moku:GO, illustrandone la composizione, le metodiche adottate e i codici sviluppati per il raggiungimento dei risultati sperimentali.
Il Capitolo 4 è dedicato all'applicazione del metodo proposto nell'articolo [4], descrivendone i principi di funzionamento e commentandone gli esiti. Successivamente, viene presentata una metodologia alternativa, basata sulla misura a ponte, per la caratterizzazione del parametro resistivo, confrontandone le prestazioni con quelle della soluzione a basso costo. Il Capitolo 5, con il quale si conclude la tesi, riporta le principali osservazioni ricavate dall'analisi dei risultati sperimentali ottenuti, evidenziando in particolare i limiti e le prestazioni che caratterizzano ciascun metodo di misura analizzato.

Capitolo 2

Stato dell'Arte

2.1 Misura dei parametri *R, C* e *L* dei componenti

La caratterizzazione dei componenti elettronici riveste un ruolo essenziale nella progettazione dei circuiti, poichè consente di prevederne il comportamento e valutarne le prestazioni. Una grandezza fondamentale impiegata in questa descrizione è l'**impedenza**, la quale in regime alternato sinusoidale è rappresentata mediante una quantità complessa e definita, per un bipolo lineare, come il rapporto tra i relativi fasori di tensione e corrente

$$Z = \frac{\overline{V}}{\overline{I}} \tag{2.1}$$

dove $\overline{V}$ e $\overline{I}$ rappresentano rispettivamente i fasori della tensione e della corrente [5]. Trattandosi di una quantità complessa, essa può essere espressa sia in forma rettangolare sia in forma polare :

$$Z = R + jX = |Z|e^{j\varphi} \tag{2.2}$$

La rappresentazione rettangolare permette di separare il contributo resistivo R, associato a fenomeni dissipativi, da quello reattivo X, rappresentativo dei fenomeni di accumulo e restituzione di energia del campo elettrico e magnetico.

La rappresentazione polare consente invece di descrivere l'impedenza ricorrendo al

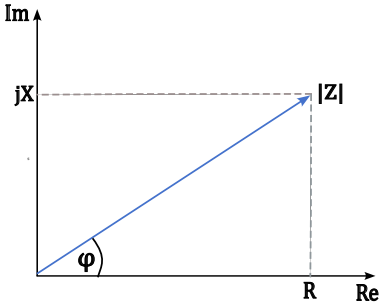


Figura 2.1: Rappresentazione dell'impedenza nel piano complesso.

suo modulo $|Z|$ e fase φ. Nel caso ideale, l'impedenza di un componente può essere espressa dal solo parametro caratteristico dello stesso, in particolare per un resistore, un induttore e un condensatore ideali si ottengono le seguenti relazioni:

$$Z_R = R \qquad Z_L = j\omega L = j2\pi f L \qquad Z_C = \frac{1}{j\omega C} = \frac{1}{j2\pi f C} \tag{2.3}$$

L'impedenza del resistore ideale è descritta dal solo parametro resistivo R, risultando puramente reale ed indipendente dalla frequenza. Al contrario, le impedenze dell'induttore e del condensatore ideali presentano unicamente una componente immaginaria, descritta dalle singole reattanze $X_L = \omega L$ e $X_C = -\frac{1}{\omega C}$, mostrando rispettivamente una dipendenza direttamente proporzionale ed inversamente proporzionale alla frequenza.

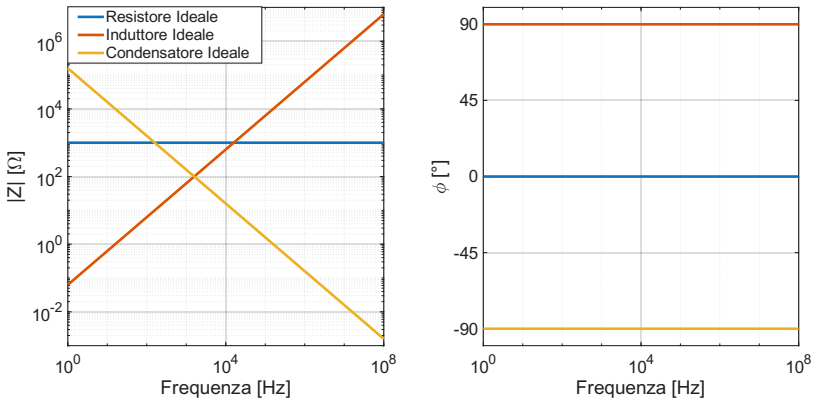


Figura 2.2: Andamento di modulo e fase dell'impedenza dei componenti ideali in funzione della frequenza (R=1 kΩ, L=10 mH, C=1 µF)

Nei componenti reali la presenza di fenomeni di non idealità ne altera il comportamento elettrico rispetto ai modelli ideali, rendendo l'impendenza non solo funzione

della frequenza, ma anche delle condizioni operative, quali temperatura, umidità e intensità delle eccitazioni. Per considerare tali effetti, ai modelli ideali vengono aggiunti elementi parassiti, ottenendo modelli equivalenti in grado di rappresentare con maggiore accuratezza il comportamento reale [3].
La misura dei parametri R, L e C consiste quindi nella stima sperimentale dei parametri che caratterizzano tali modelli equivalenti.

2.2 Modelli equivalenti associati ai componenti reali

In generale, la rete equivalente associata ad un componente reale, ottenuta dalla combinazione del suo modello ideale con i relativi elementi parassiti, può risultare notevolmente complessa e dipendente dalla struttura fisica e dalla tecnologia costruttiva del componente stesso.

2.2.1 Resistore Reale

Nel caso di un **resistore reale**, il suo comportamento elettrico può essere modellato tramite la serie di una resistenza R con un'induttanza parassita L', poiché gli effetti induttivi sono associati alla stessa corrente che attraversa il resistore, il tutto posto in parallelo ad una capacità parassita C', che rappresenta l'accoppiamento capacitivo distribuito tra i terminali. L'impedenza complessiva, in regime sinusoidale a pulsazione ω, risulta pari a

$$Z_{eq} = (R + j\omega L') \parallel \frac{1}{j\omega C'} = \frac{R + j\omega L'}{1 - \omega^2 L'C' + j\omega RC'}.$$

Dal modello equivalente completo, adottando opportune ipotesi sull'intervallo di lavoro del componente, è possibile ricavare le rappresentazioni equivalenti di tipo serie e parallelo.

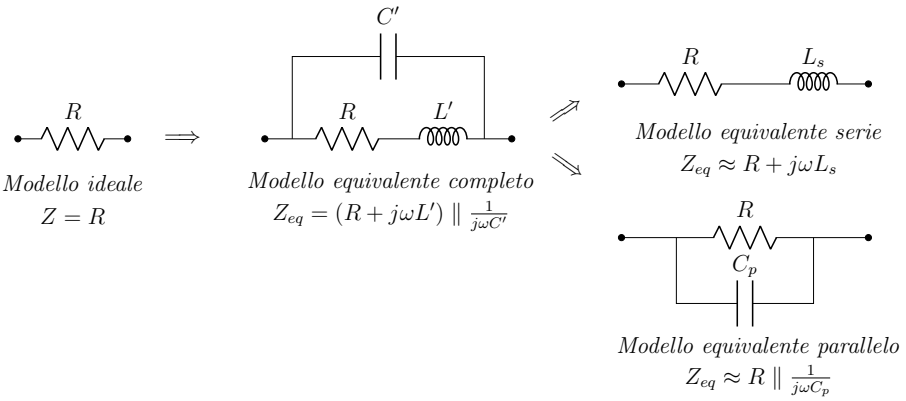


Figura 2.3: Modelli equivalenti del resistore reale.

Normalmente il resistore viene utilizzato in un intervallo di frequenze tale che risultino soddisfatte le condizioni $\omega L' \ll R$ e $\frac{1}{\omega C'} \gg R$. Nel suddetto intervallo si ha inoltre $\omega^2 L'C' \ll 1$, condizione che permette di considerare separabili gli effetti induttivi e capacitivi della rete. Questo consente, mediante opportune approssimazioni, di semplificare l'espressione dell'impedenza equivalente, ottenendo

$$Z_{eq} \approx R + j\omega(L' - R^2 C').$$

Dall'ultima relazione è possibile estrarre il modello serie definendo l'induttanza equivalente come $L_s = L' - R^2 C'$. Analogamente, sviluppando l'espressione in ammettenza

$$Y_{eq} = \frac{1}{Z_{eq}} = \frac{1}{R + j\omega L'} + j\omega C',$$

e applicando le corrispondenti approssimazioni si ottiene $Y_{eq} \approx \frac{1}{R} + j\omega(C' - \frac{L'}{R^2})$ da cui, definendo la capacità equivalente parallela come $C_p = C' - \frac{L'}{R^2}$, si ricava il modello parallelo. La scelta della rappresentazione equivalente più appropriata dipende dalla frequenza di lavoro del componente: alle basse frequenze è preferibile adottare il modello serie, mentre all'aumentare della frequenza gli effetti capacitivi diventano predominanti, rendendo più conveniente l'utilizzo del modello parallelo.

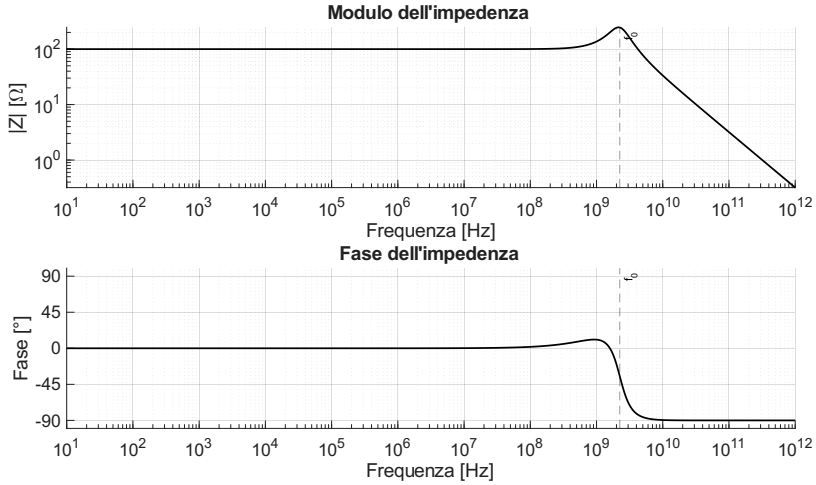


Figura 2.4: Risposta in frequenza del modello completo di impedenza di un resistore reale, con parametri R=100 Ω, L'=10 nH e C'=0.5 pF.

2.2.2 Condensatore Reale

Nel caso di un **condensatore reale**, il suo comportamento elettrico può essere modellato introducendo in parallelo alla capacità C una resistenza R'_p, associata alle perdite nel dielettrico reale. Il tutto è connesso in serie a una resistenza R' e un'induttanza L' che modellano, rispettivamente, le perdite ohmiche del componente e gli effetti induttivi ad esso associati. L'impedenza complessiva, in regime sinusoidale a pulsazione ω, risulta pari a

$$Z_{eq} = R' + j\omega L' + \left(\frac{1}{j\omega C} \parallel R'_p\right) = R' + j\omega L' + \frac{R'_p - j\omega R'^2_p C}{1 + \omega^2 R'^2_p C^2}.$$

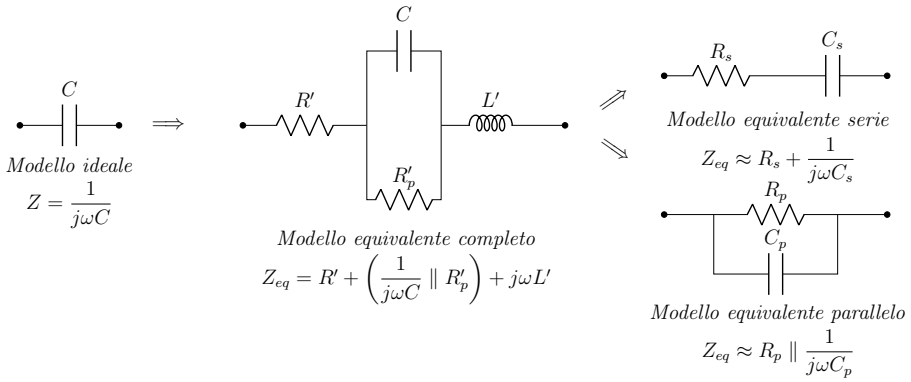


Figura 2.5: Modelli equivalenti del condensatore reale.

Nell'ultima relazione si assume che il condensatore operi a basse frequenze, ovvero che sia verificata la condizione $\omega R'_p C \ll 1$, generalmente soddisfatta nelle applicazioni di interesse. Sotto tale ipotesi l'impedenza può essere ricondotta nella forma:

$$Z_{eq} \approx R' + \frac{1}{\omega^2 C^2 R'_p} + j\omega L' + \frac{1}{j\omega C} = R' + \frac{1}{\omega^2 C^2 R'_p} + \frac{1 - \omega^2 L' C}{j\omega C}$$

Definendo la pulsazione di autorisonanza come $\omega_0 = \frac{1}{\sqrt{L'C}}$, si identificano per confronto con il modello equivalente serie, i relativi parametri caratteristici

$$R_s = R' + \frac{1}{\omega^2 C^2 R'_p}, \qquad C_s = \frac{C}{1 - \left(\frac{\omega}{\omega_0}\right)^2}.$$

Le espressioni evidenziano come per $\omega = 0$ risulta $C_s = C$, mentre, all'aumentare della frequenza, la capacità equivalente C_s assume valori progressivamente crescenti rispetto a C e formalmente, per $\omega \to \omega_0$, la capacità C_s tende a divergere, evidenziando i limiti del modello. I parametri del modello equivalente parallelo si ottengono a partire da quelli del modello serie, imponendo l'uguaglianza tra le rispettive espressioni dell'impedenza equivalente. Si ricavano quindi

$$R_p = R_s(1 + Q^2), \qquad C_p = \frac{C_s}{1 + \frac{1}{Q^2}}$$

dove $Q = \frac{1}{\omega C_s R_s} = \omega C_p R_p$ rappresenta il **fattore di qualità** del condensatore.

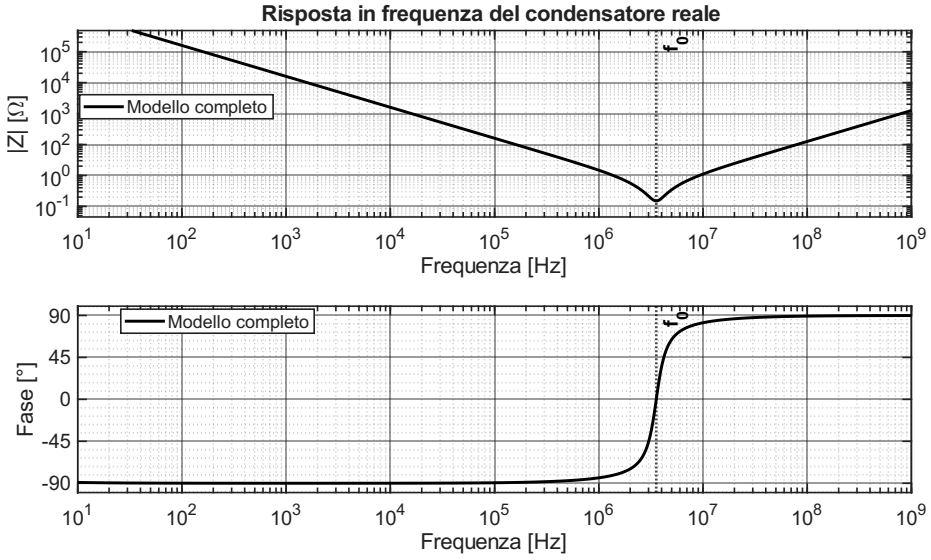


Figura 2.6: Risposta in frequenza del modello completo di impedenza di un condensatore reale, con parametri C=100 nF, R'=0.15 Ω, L'=20 nH e R'_p=10 MΩ.

2.2.3 Induttore Reale

Nel caso di un **induttore reale**, il suo comportamento elettrico può essere modellato introducendo in parallelo all'induttanza ideale L una resistenza R'_p, che rappresenta le perdite di isolamento imperfetto tra le spire, le perdite dovute allo scorrere di correnti parassite e quelle relative a fenomeni di isteresi. Inoltre, l'effetto del campo elettrico generato dalla differenza di potenziale tra le singole spire e i conduttori circostanti viene modellato inserendo in parallelo una capacità C'. Infine, il tutto è connesso in serie ad una resistenza R' che tiene conto della caduta di tensione nel filo conduttore della bobina. L'impedenza complessiva, in regime sinusoidale a pulsazione ω, risulta pari a

$$Z_{eq} = R' + \left(j\omega L \parallel R'_p \parallel \frac{1}{j\omega C'} \right) = R' + R'_p \frac{\omega^2 L^2 + j\omega L R'_p (1 - \omega^2 LC')}{R'^2_p (1 - \omega^2 LC') + \omega^2 L^2}.$$

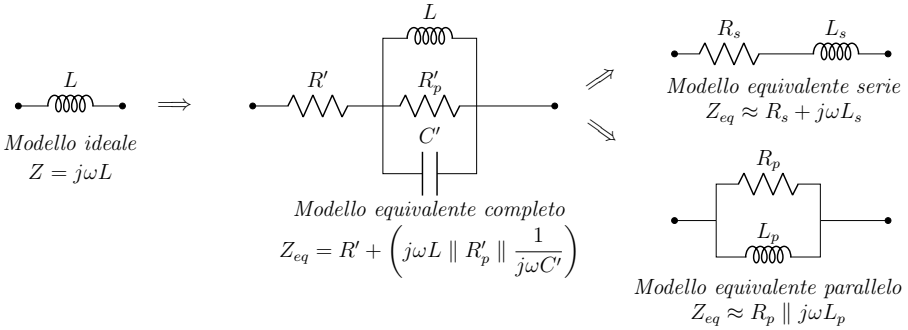


Figura 2.7: Modelli equivalenti dell'induttore reale.

Esaminando l'ultima relazione si può identificare una pulsazione di autorisonanza $\omega_0 = \frac{1}{\sqrt{LC'}}$, alla quale la parte reattiva dell'impedenza si annulla e l'induttore si comporta come una resistenza pura, pari a $R' + R'_p$. Per pulsazioni superiori ad ω_0, poiché prevalgono gli effetti capacitivi, il componente assume un comportamento ohmico-capacitivo, quindi nella pratica esso viene utilizzato per lavorare a pulsazioni almeno dieci volte minori rispetto a quella di risonanza. Pertanto, nelle applicazioni di interesse risulta generalmente verificata l'ipotesi $\omega \ll \frac{1}{\sqrt{LC'}}$, permettendo così di semplificare l'espressione dell'impedenza, mediante le opportune approssimazioni,

come segue:

$$Z_{eq} \approx R' + \frac{\omega^2 L^2 R'_p}{R'^2_p + \omega^2 L^2} + j\frac{\omega L R'^2_p}{R'^2_p + \omega^2 L^2}.$$

Confrontando l'ultima relazione con il modello equivalente serie, è possibile ricavare i valori dei relativi parametri

$$R_s = R' + \frac{\omega^2 L^2 R'_p}{R'^2_p + \omega^2 L^2}, \qquad L_s = \frac{L}{1 + (\frac{\omega L}{R'_p})^2}.$$

Tali espressioni evidenziano che per $\omega L \ll R'_p$ si ha $L_s \approx L$, mentre, all'aumentare della frequenza, l'induttanza equivalente serie L_s diminuisce sempre più rispetto a quella nominale L. Tale andamento risulta valido finché il componente opera a pulsazioni sufficientemente inferiori a quella di autorisonanza, al di sopra della quale il modello perde di validità. I parametri del modello equivalente parallelo si ottengono a partire da quelli del modello serie, imponendo l'uguaglianza tra le rispettive espressioni dell'impedenza. Si ricavano quindi

$$R_p = R_s(1 + Q^2), \qquad L_p = L_s(1 + \frac{1}{Q^2})$$

dove $Q = \dfrac{\omega L_s}{R_s} = \dfrac{R_p}{\omega L_p}$ è il **fattore di qualità** dell'induttore.

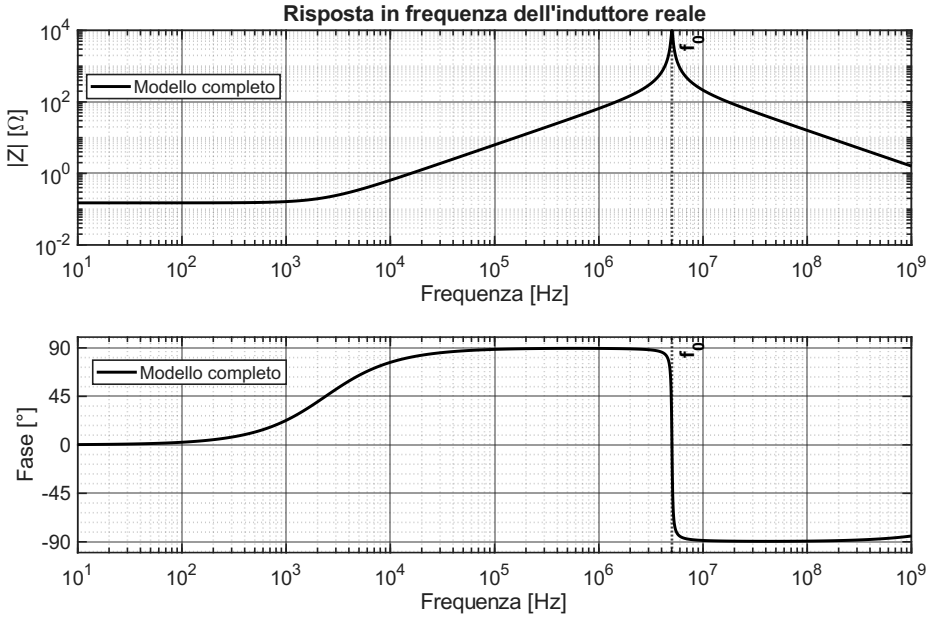


Figura 2.8: Risposta in frequenza del modello completo di impedenza di un induttore reale, con parametri L=10 µH, R'_p=10 kΩ, R'=0.15 Ω e C'=100 pF.

2.3 Metodi per la misura dei parametri *R, C* e *L*

La determinazione dei parametri R, C, e L associati ad un componente passivo può essere effettuata mediante differenti metodi di misura, ognuno fondato su un diverso principio fisico e caratterizzato da uno specifico campo di applicazione, da un certo livello di accuratezza e da un proprio intervallo di frequenze operative.
La scelta del metodo di misura più idoneo dipende dalle caratteristiche del componente da analizzare e dalla precisione desiderata nella stima dei suoi parametri equivalenti. Nel seguito saranno descritti i principali metodi di misura impiegati per la caratterizzazione dei componenti passivi, sia in termini di impedenza sia direttamente attraverso i relativi parametri R, C e L: il **metodo a ponte**, il **metodo della risonanza**, il **metodo della risposta transitoria**, il **metodo voltampermetrico** e il **metodo del ponte auto-bilanciante**.

2.3.1 Metodo a ponte

Il metodo a ponte è basato sul confronto tra il componente incognito da caratterizzare e componenti campione di valore noto, collegati come mostrato nella configurazione più generale in **Figura 2.9**. In tale configurazione, V rappresenta un rivelatore di tensione utilizzato per misurare il segnale di sbilanciamento V_ω, mentre $V_{\omega in}$ un generatore, generalmente alternato e sinusoidale, che alimenta il ponte stesso. Dall'analisi del circuito, è possibile ricavare l'espressione del segnale di sbilanciamento V_ω che risulta pari a:

$$V_\omega = V_{Z_2} - V_{Z_x} = V_{\omega in}(\frac{Z_2}{Z_1 + Z_2} - \frac{Z_x}{Z_3 + Z_x}) = V_{\omega in}\frac{Z_2 Z_3 - Z_1 Z_x}{(Z_1 + Z_2)(Z_3 + Z_x)}.$$

Nell'utilizzo della configurazione a ponte per eseguire misure si è di solito interessati a determinare le condizioni per ottenere lo stato di equilibrio, nel quale $|V_\omega| = 0$. Questa condizione risulta verificata se rispettato il vincolo sulle impedenze

$$Z_2 Z_3 = Z_1 Z_x,$$

da cui è possibile ricavare l'impedenza incognita:

$$Z_x = \frac{Z_2}{Z_1} Z_3.$$

Esistono varie configurazioni di ponte, differenti tra loro per la tipologia di componenti impiegati e per le finalità della misura che rendono possibile:

- **Ponte di Wheatstone**: impiegato per la misura di resistenze, costituito esclusivamente da impedenze puramente reali.
- **Ponte di Maxwell**: impiegato per la misura di induttanze reali, modellate mediante un'induttanza L_x in serie a una resistenza R_x, che rappresenta le perdite dell'avvolgimento.
- **Ponte di de Sauty**: impiegato per la misura di capacità, costituito generalmente da due impedenze puramente capacitive e due impedenze puramente resistive [6].

Il metodo a ponte garantisce generalmente un'elevata accuratezza, risultando particolarmente adatto alle misure di precisione dei componenti. La necessità di raggiungere la condizione di equilibrio ne limita tuttavia la rapidità e la praticità rispetto ad altre tecniche di misura.

Tabella 2.1: Configurazioni dei principali ponti di misura

Ponte	$\mathbf{Z_1}$	$\mathbf{Z_2}$	$\mathbf{Z_3}$	$\mathbf{Z_x}$
Wheatstone	R_1	R_2	R_3	R_x
Maxwell	R_1	R_2	$R_3 \parallel C$	$R_x + j\omega L_x$
De Sauty	R_1	R_2	C_1	C_x

2.3.2 Metodo della risonanza

Il metodo della risonanza si basa sulla realizzazione di circuiti risonanti, in configurazione serie o parallelo, al cui interno viene inserito il componente incognito da caratterizzare. Esprimendo l'impedenza complessiva del circuito nella forma $Z_{eq} = R + jX$, la condizione di risonanza si verifica quando la componente reattiva dell'impedenza della rete risulta nulla, ovvero $X = 0$. Nella sua configurazione più generale, il circuito di misura comprende, oltre al componente da caratterizzare, un generatore di tensione sinusoidale V_{in} che alimenta il circuito ad una determinata frequenza, un

elemento reattivo variabile, il cui valore viene regolato fino al raggiungimento della condizione di risonanza, e un voltmetro, collegato ai suoi capi, utilizzato per verificare tale condizione. Al verificarsi della risonanza, la conoscenza del valore assunto dal componente variabile e della frequenza di eccitazione permette di ricavare indirettamente il parametro incognito del componente in esame. Nel caso di un circuito LC ideale, la condizione di risonanza è descritta dalla relazione

$$\omega_0 = \frac{1}{\sqrt{LC}}$$

attraverso la quale è possibile ricavare uno dei due parametri a partire dalla conoscenza dell'altro. Questo approccio può essere esteso al caso di componenti reali, ricorrendo ai relativi modelli equivalenti, dei quali si possono ottenere informazioni non solo sul parametro nominale, ma anche sui parametri parassiti.

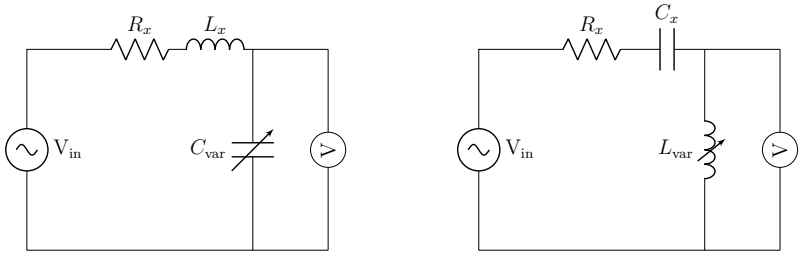


Figura 2.9: Circuiti risonanti utilizzati per la caratterizzazione di un induttore (a sinistra) e di un condensatore (a destra).

Il metodo a risonanza risulta particolarmente indicato per la caratterizzazione di componenti per i quali sia di interesse la determinazione del fattore di qualità e delle perdite. La misura è tuttavia indiretta e richiede una condizione di risonanza sufficientemente definita.

2.3.3 Metodo della risposta transitoria

Il metodo della risposta transitoria permette di determinare i parametri R, C e L di componenti passivi analizzandone la risposta fornita nel dominio del tempo a seguito dell'applicazione di una certa eccitazione, generalmente un segnale a gradino. Il

metodo viene normalmente applicato a circuiti RC e RL del primo ordine, costituiti da un resistore e un elemento reattivo che ne caratterizza la risposta dinamica. Tali circuiti, quando vengono eccitati con un ingresso a gradino di ampiezza V_0, forniscono una risposta descritta dalla soluzione delle seguenti equazioni differenziali, ricavate dalle relazioni costitutive dei componenti e dall'applicazione delle leggi di Kirchhoff ai nodi e alle maglie:

$$RC\frac{dv_C(t)}{dt} + v_C(t) = V_0, \qquad L\frac{di(t)}{dt} + Ri(t) = V_0.$$

Assumendo nulle le condizioni iniziali, si ottengono rispettivamente le soluzioni

$$v_C(t) = V_0(1 - e^{-\frac{t}{RC}}) \qquad i(t) = \frac{V_0}{R}(1 - e^{-t\frac{R}{L}})$$

dalle quali viene definita, per ciascun circuito, la rispettiva costante di tempo, $\tau_{RC} = RC$ e $\tau_{RL} = \dfrac{L}{R}$, che risulta legata ai parametri caratteristici dei componenti utilizzati. Nel caso del circuito RL, dall'espressione della corrente è possibile ricavare quella della tensione ai capi dell'induttore mediante la relazione $v_L(t) = L\dfrac{di(t)}{dt}$ con cui si ottiene:

$$v_L(t) = V_0 e^{-t\frac{R}{L}}$$

La conoscenza analitica delle curve di risposta permette di ricavare la costante di tempo attraverso la misura dell'intervallo di tempo necessario al raggiungimento del 63.2% del valore finale di tensione nel caso RC, oppure alla riduzione al 36.8% del valore iniziale di tensione nel caso RL. In entrambi i casi tale intervallo corrisponde a $\Delta t = \tau$. Pertanto, la misura del tempo necessario al raggiungimento di tale valore consente di determinare direttamente la costante di tempo del circuito. Nel caso del circuito RC, una volta determinata τ_{RC} il valore della capacità può essere ricavato dalla relazione

$$C = \frac{\tau_{RC}}{R},$$

mentre nel caso del circuito RL, dalla misura di τ_{RL} si ottiene il valore dell'induttanza come

$$L = \tau_{RL} R.$$

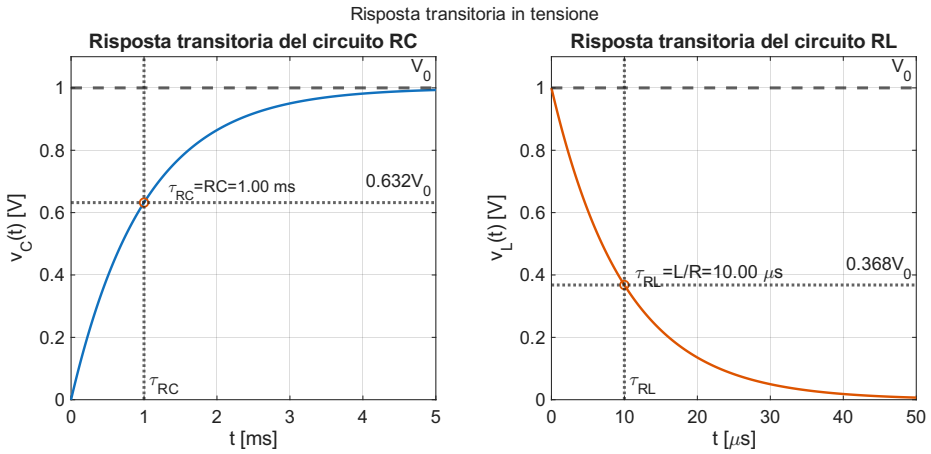


Figura 2.10: Risposta transitoria in tensione di un circuito RC e RL, con R=1 kΩ, C=1 µF e L=10 mH.

Il metodo consente quindi di determinare i parametri dei componenti reattivi a partire dalla misura della risposta transitoria, assumendo noto il valore della resistenza presente nel circuito. Le sue prestazioni sono fortemente legate alla banda operativa e ai tempi di risposta della strumentazione utilizzata.

2.3.4 Metodo voltamperometrico

Il metodo voltamperometrico, realizzato con circuito mostrato in **Figura 2.11** consente di determinare l'impedenza di un componente incognito, eccitato mediante un generatore di tensione sinusoidale V_{in}, a partire dalla misura della tensione e della corrente ad esso associate. La misura della corrente può essere effettuata indirettamente misurando la caduta di tensione ai capi di un resistore di shunt di basso valore, connesso in serie al componente, applicando poi la relazione

$$I = \frac{V_{R_{shunt}}}{R_{shunt}}.$$

Una volta determinate tensione e corrente, l'impedenza incognita viene calcolata come

$$Z_{eq} = \frac{V}{I} = \frac{V}{V_{R_{shunt}}} R_{shunt}.$$

Nella pratica, si preferisce sostituire il resistore di shunt con un trasformatore a basse perdite così da ridurre gli effetti dovuti al suo inserimento, sebbene ciò limiti l'intervallo di frequenze in cui risulta utilizzabile. Questo metodo si distingue per la

semplicità della configurazione e per la possibilità di determinare direttamente l'impedenza a partire dalle grandezze elettriche misurate. L'accuratezza della misura dipende tuttavia dalle caratteristiche degli strumenti e dagli effetti introdotti con i collegamenti.

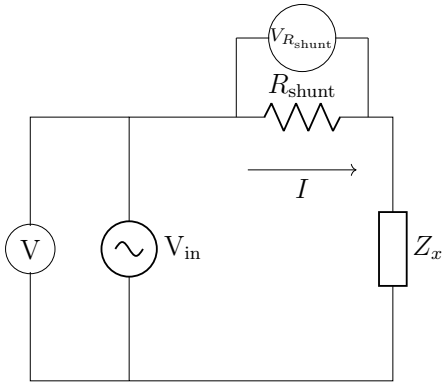


Figura 2.11: Schema della misura voltamperometrica dell'impedenza incognita Z_x.

2.3.5 Metodo a ponte autobilanciante

Il metodo a ponte autobilanciante è basato sullo stesso principio di misura del metodo voltamperometrico, in quanto l'impedenza incognita Z_x del componente viene valutata come rapporto tra la tensione ai suoi capi e la corrente che lo attraversa, dopo aver eccitato il componente mediante un generatore di tensione sinusoidale V_{in}. La differenza tra i due risiede nell'approccio utilizzato per la misura della corrente. Nel metodo precedente, si ricorre all'inserzione di un trasformatore a basse perdite oppure all'utilizzo di un resistore di shunt, mentre nel ponte autobilanciante viene utilizzato un amplificatore operazionale nella configurazione mostrata in **Figura 2.12**. Tale configurazione consente di ridurre al minimo l'alterazione delle condizioni di misura e dell'impedenza del componente.

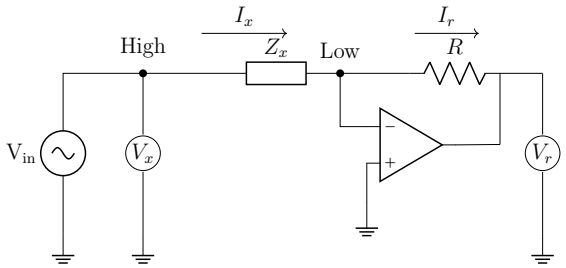


Figura 2.12: Metodo del ponte autobilanciante per la misura dell'impedenza incognita Z_x.

Questa impostazione fa si che il punto "Low" sia posto a massa virtuale attraverso l'ingresso invertente dell'amplificatore operazionale, quindi permettendo di imporre l'uguaglianza tra le correnti I_x e I_r, dalla quale si ricava

$$\frac{V_x}{Z_x} = I_x = I_r = \frac{V_r}{R},$$

e, conseguentemente, manipolando la relazione precedente, si ottiene

$$Z_x = R\frac{V_x}{V_r}.$$

Pertanto, misurando la tensione ai capi dell'impedenza Z_x, indicata con V_x, e quella ai capi della resistenza nota R, indicata con V_r, è possibile ricavare l'impedenza incognita del componente attraverso l'inserzione di due voltmetri, oppure uno solo dotato di un selettore.

Il ponte autobilanciante consente di ottenere misure accurate e automatizzate su un ampio intervallo di impedenze, rappresentando una soluzione particolarmente adatta alla strumentazione moderna per la misura di componenti RLC. La maggiore complessità circuitale e gli effetti parassiti diventano tuttavia aspetti rilevanti all'aumentare della frequenza.

2.4 Strumenti per la misura dei parametri R, C e L

I metodi di misura descritti nei paragrafi precedenti sono alla base delle differenti architetture degli strumenti utilizzati per caratterizzare i parametri R, C e L di un

componente o in generale per descriverne l'impedenza. Tali strumenti posso essere suddivisi in tre categorie principali: ***LCR* meter**, **analizzatori d'impedenza** e **analizzatori di reti vettoriali (*VNA*)**.
È tuttavia possibile effettuare la caratterizzazione dei parametri R, C e L anche mediante strumenti non espressamente progettati per tale scopo. Attraverso opportune configurazioni e l'interfacciamento di differenti funzionalità di misura, strumenti multifunzione possono essere impiegati per determinare l'impedenza di un componente e ricavarne successivamente i relativi parametri equivalenti. Un esempio è rappresentato dal Moku:GO, che, mediante la sua configurazione da Frequency Analyzer, può essere utilizzato per la caratterizzazione dell'impedenza di un componente, come verrà approfondito nelle sezioni successive.

2.4.1 *LCR* meter

Gli LCR meter sono strumenti progettati per fornire la stima diretta dei parametri R, C e L del componente analizzato. Molti strumenti appartenenti a questa categoria, così come numerosi analizzatori di impedenza, basano il loro principio di funzionamento sul metodo del ponte autobilanciante, nel quale l'impedenza complessa incognita viene determinata mediante la misura della tensione ai suoi capi e della corrente che lo attraversa. Partendo dall'analisi dell'impedenza complessa, è possibile ricavare i parametri relativi alle diverse rappresentazioni equivalenti con cui il componente può essere modellato. Tra i principali parametri normalmente forniti dallo strumento si possono individuare:

- R_s**,** L_s **e** C_s, associati alla rappresentazione equivalente serie;
- R_p**,** L_p **e** C_p, associati alla rappresentazione equivalente parallelo;
- $|Z|$ **e** φ, che rappresentano rispettivamente il modulo e la fase dell'impedenza;
- Q, fattore di qualità, e D, fattore di dissipazione (dissipation factor).

Le architetture più moderne e avanzate consentono inoltre di eseguire automaticamente una serie di misure al variare della frequenza di eccitazione, permettendo di

caratterizzare l'impedenza del componente nell'intervallo di frequenze di interesse e di determinarne la risposta in frequenza. In funzione dello strumento utilizzato, tale intervallo può estendersi da pochi hertz fino a frequenze dell'ordine dei megahertz, mentre l'ampiezza del segnale di test può essere opportunamente impostata in funzione del tipo di componente e delle condizioni di misura.

2.4.2 Analizzatori d'impedenza

Gli analizzatori di impedenza sono strumenti di misura progettati appositamente per fornire una descrizione dell'impedenza complessa di un componente in funzione della frequenza. Rispetto ai tradizionali LCR meter, generalmente orientati alla determinazione dei parametri equivalenti R, L e C a una o più frequenze di misura, gli analizzatori di impedenza sono maggiormente orientati alla caratterizzazione completa del comportamento del dispositivo nell'intervallo di frequenze di interesse.
La grandezza fondamentale ottenuta dalla misura è l'impedenza complessa

$$Z(f) = |Z(f)|e^{j\varphi(f)},$$

dalla quale possono essere ricavate, in funzione della frequenza, la componente reale e immaginaria dell'impedenza, il modulo $|Z|$ e la fase φ.
A partire da tali grandezze è inoltre possibile determinare i parametri equivalenti del componente, analogamente al caso degli LCR meter.
Una delle principali caratteristiche degli analizzatori di impedenza è la possibilità di eseguire automaticamente uno sweep in frequenza, acquisendo un numero elevato di punti di misura e permettendo quindi di ottenere la risposta in frequenza del dispositivo analizzato. Questa modalità di misura consente di individuare fenomeni come risonanze, antirisonanze e variazioni dei parametri parassiti, non osservabili mediante una singola misura effettuata ad una specifica frequenza.
Le prestazioni dello strumento sono determinate principalmente dall'intervallo di frequenza analizzabile, dalla gamma di impedenza misurabile, dal livello del segnale di eccitazione e dall'accuratezza della misura. Tali caratteristiche risultano par-

ticolarmente importanti nella scelta dello strumento in relazione al dispositivo da caratterizzare e all'intervallo di frequenze di interesse.

2.4.3 Analizzatori di reti vettoriali

Gli analizzatori di reti vettoriali sono strumenti principalmente progettati per la caratterizzazione di dispositivi e reti elettriche operanti ad alte frequenze e nel campo delle microonde, dove sono rilevanti gli effetti propagativi del segnale e l'adattamento delle impedenze. Differiscono dagli strumenti precedentemente descritti, in quanto consentono la misura delle grandezze associate a segnali incidenti, riflessi e trasmessi. Tali misure vengono effettuate al variare della frequenza, permettendo una caratterizzazione compoleta dei dispositivi analizzati attraverso la determinazione dei relativi parametri di scattering, dai quali è possibile ricavare l'impedenza complessa. Questo permette di estendere la caratterizzazione dei parametri R, L e C, ricavati dall'analisi dell'impedenza, a frequenze generalmente più elevate rispetto a quelle tipiche degli LCR meter e degli analizzatori di impedenza.

Capitolo 3

Confronto tra strumenti

Al fine di valutare le prestazioni e i risultati ottenuti mediante differenti metodi e strumenti per la caratterizzazione dei parametri R, C e L dei componenti elettronici, è stato condotto un confronto tra due diversi banchi di misura: il primo realizzato con un **LCX200 di Rhode&Schwarz**, mentre il secondo basato su un **Moku:GO di Liquid Instrument**. Tale confronto è stato effettuato attraverso una campagna di misure condotta su un insieme di dodici componenti passivi, costituito da sei induttori e sei condensatori caratterizzati da differenti valori nominali:

$$C_{nom} = \{0.1\,;\, 0.22\,;\, 0.47\,;\, 0.68\,;\, 2.2\,;\, 10\}\mu\text{F}$$

$$L_{nom} = \{0.001\,;\, 0.033\,;\, 0.150\,;\, 1\,;\, 10\,;\, 20\}\text{mH}$$

I medesimi componenti sono stati utilizzati per entrambe le configurazioni di misura, in modo da rendere direttamente confrontabili i risultati ottenuti mediante i due sistemi. Nel seguito verranno descritti i due banchi di misura, illustrandone l'implementazione e le relative metodologie adottate. I risultati ottenuti saranno successivamente analizzati e posti a confronto, assumendo l'LCX200 come strumento di riferimento.

3.1 Banco di misura basato su LCX200 di R&S

Il primo banco di misura è stato realizzato utilizzando l'**LCX200 di R&S**, un *LCR* meter da laboratorio, al quale è stato collegato l'apposito *test fixture* per la connessione dei componenti da caratterizzare. In particolare, è stato impiegato il modello LCX-Z1, un adattatore per componenti con terminali assiali o radiali, basato su una configurazione di misura a quattro terminali. Tale configurazione consente di effettuare la misura dell'impedenza del dispositivo in prova riducendo l'influenza delle resistenze e delle induttanze associate ai collegamenti. Lo strumento è stato inoltre interfacciato con un PC esterno mediante collegamento USB, consentendo l'esecuzione automatizzata delle misure e la successiva acquisizione dei dati. Il controllo dello strumento è stato realizzato in ambiente MATLAB, tramite l'apposito *Instrument Control Toolbox*, utilizzando il protocollo di comunicazione SCPI (*Standard Commands for Programmable Instruments*).
Attraverso tale interfaccia è stato possibile configurare da remoto i parametri di misura e gestire l'acquisizione della risposta in frequenza.



Figura 3.1: Banco di prova con LCX200.

3.1.1 Procedura di misura

In seguito all'allestimento del banco di misura, sono state effettuate le misurazioni mirate alla caratterizzazione della risposta in frequenza dell'impedenza complessa, valutata in modulo $|Z|$ e fase φ, di ciascun componente campione.

Al fine di rendere le misure ripetibili e uniformi per tutti i componenti, l'intera procedura è stata automatizzata e gestita mediante uno script sviluppato in MATLAB. In particolare, l'acquisizione della risposta in frequenza è stata effettuata nell'intervallo compreso tra 100 Hz e 100 kHz, considerando 4096 punti in frequenza distribuiti logaritmicamente e impostando il valore della tensione di eccitazione a 1 Vrms. Lo script è stato quindi strutturato in modo da gestire le diverse fasi della misura. Inizialmente viene instaurata la comunicazione con l'LCX200 mediante interfaccia VISA; successivamente vengono definiti i punti di frequenza e configurati i parametri di misura, per procedere infine all'acquisizione e all'elaborazione dei dati. La procedura descritta è stata implementata mediante lo script MATLAB riportato di seguito.

```
clear;
clc;
d = visadevfind("ResourceName","USB0::0x0AAD::0x0197::3629.8856k03
    -102448::0::INSTR");
if isempty(d)
    dev = visadev("USB0::0x0AAD::0x0197::3629.8856k03-102448::0::INSTR");
else
    dev = d;
end
configureTerminator(dev,"LF");
f = logspace(2,5,4096);

writeline(dev,"FUNCtion:IMPedance:TYPE ZTD");
writeline(dev,"VOLT 1");
[Z,theta] = misuraLCX200(dev,f);
```

Script 3.1: Script MATLAB per l'acquisizione della risposta in frequenza dell'impedenza mediante LCX200

La prima parte dello script è dedicata all'instaurazione della comunicazione tra MATLAB e l'LCX200. A tale scopo viene utilizzata l'interfaccia VISA, attraverso la

quale lo strumento viene identificato mediante il relativo *Resource Name*. La funzione *visadevfind* consente di verificare l'eventuale presenza di una connessione già attiva, qualora questa non sia disponibile, viene instaurata una nuova connessione mediante la funzione *visadev*. Infine, viene configurato il carattere di terminazione utilizzato per lo scambio dei messaggi tra MATLAB e lo strumento. Nella seconda parte viene definito, tramite il comando *logspace*, il vettore delle frequenze contenente i 4096 punti distribuiti logaritmicamente nell'intervallo 100 Hz e 100 kHz. Successivamente, vengono configurati i parametri di misura dell'LCX200 mediante l'invio dei relativi comandi SCPI. In particolare, vengono impostate la misura d'impedenza, in termini di modulo e fase, e la tensione di eccitazione, fissata a 1 Vrms. L'acquisizione effettiva del vettore contenente la risposta in frequenza viene realizzata mediante uno sweep in frequenza, questo implementato attraverso la funzione *misuraLCX200()*, riportata di seguito.

```
function [val1,val2] = misuraLCX200(dev,f)

    val1 = zeros(1,length(f));
    val2 = zeros(1,length(f));
for k = 1:length(f)
    writeline(dev,"FREQ " + string(f(k)));
    writeline(dev,"INIT");
    pause(0.05);

    writeline(dev,"FETCh?");
    data = readline(dev);
    val = str2double(split(data,','));
    val1(k) = val(1);
    val2(k) = val(2);
end
end
```

Script 3.2: Script MATLAB contenente la funzione misuraLCX200()

La funzione *misuraLCX200()* realizza l'acquisizione della risposta in frequenza per le frequenze definite nello script principale. In ingresso riceve la connessione con lo strumento e il vettore delle frequenze di misura, mentre restituisce due vettori contenenti, rispettivamente, il modulo dell'impedenza e la fase.

Inizialmente vengono allocati i vettori destinati alla memorizzazione dei risultati. Successivamente, mediante un ciclo for, la misura viene ripetuta per ciascun punto del vettore delle frequenze. Per ogni iterazione viene impostata sull'LCX200 la frequenza corrispondente e viene avviata la misura mediante il comando INIT. Dopo una breve pausa, necessaria a consentire il completamento dell'acquisizione, il comando *FETCh?* richiede allo strumento i risultati della misura.

I dati restituiti dall'LCX200 vengono quindi letti tramite MATLAB e convertiti in valori numerici. Le due grandezze ottenute vengono infine memorizzate nei rispettivi vettori, mantenendo la corrispondenza con il punto di frequenza al quale sono state acquisite. Una volta salvati i vettori relativi al modulo e alla fase dell'impedenza, questi vengono elaborati nel workspace di MATLAB per ricavare i parametri della rappresentazione serie associata al componente, mediante le relazioni

$$R_s = |Z|cos(\theta_z), \quad X_s = |Z|sin(\theta_z).$$

In funzione del segno della reattanza X_s, viene quindi determinata la capacità equivalente serie C_s, nel caso $X_s < 0$, oppure l'induttanza equivalente serie L_s, nel caso $X_s > 0$, tramite le espressioni

$$C_s = \frac{-1}{2\pi f X_s}, \quad L_s = \frac{X_s}{2\pi f}.$$

Infine, i parametri ottenuti vengono rappresentati graficamente in funzione della frequenza, al fine di confrontarli con i risultati dell'altro banco di misura. La procedura descritta è stata implementata mediante uno script MATLAB, riportato di seguito, e ripetuta per tutti i dodici campioni disponibili. Per ciascuna misura è stato quindi sostituito il componente in esame, memorizzando i relativi vettori e generando i corrispondenti grafici.

```
% Variabile per distinguere i componenti
induttore = true;
% Conversione della fase in radianti
theta_rad = deg2rad(Theta_LCX200);

% Parte reale e immaginaria dell'impedenza
Rs_LCX200 = Z_LCX200 .* cos(theta_rad);      % Resistenza serie [Ohm]
Xs_LCX200 = Z_LCX200 .* sin(theta_rad);      % Reattanza serie [Ohm]

% Controllo effettuato sul segno di Xs
Cs_LCX200 = NaN(size(Xs_LCX200));
Ls_LCX200 = NaN(size(Xs_LCX200));
indL = Xs_LCX200 > 0;
indC = Xs_LCX200 < 0;
Ls_LCX200(indL) = Xs_LCX200(indL) ./ (2*pi*f(indL));              %
    Induttanza serie [H]
Cs_LCX200(indC) = -1 ./ (Xs_LCX200(indC) .* (2*pi*f(indC)));      % Capacita
    ' serie [F]

% plotting risultati
figure('Name','Risposta in frequenza (LCX200)','Color','w');
    subplot(3,2,1);
    semilogx(f, Z_LCX200,'LineWidth',1.2);
    grid on;
    xlabel('Frequenza [Hz]');
    ylabel('|Z| [\Omega]');
    title('Modulo dell''impedenza');
    subplot(3,2,2);
    semilogx(f, Theta_LCX200,'LineWidth',1.2);
    grid on;
    xlabel('Frequenza [Hz]');
```

```
30      ylabel('$\theta$ [$^\circ$]');
31      title('Fase dell''impedenza');
32      subplot(3,2,3);
33      semilogx(f, Rs_LCX200,'LineWidth',1.2);
34      grid on;
35      xlabel('Frequenza [Hz]');
36      ylabel('R_s [\Omega]');
37      title('Resistenza serie');
38      subplot(3,2,4);
39      semilogx(f, Xs_LCX200,'LineWidth',1.2);
40      grid on;
41      xlabel('Frequenza [Hz]');
42      ylabel('X_s [\Omega]');
43      title('Reattanza serie');
44      if induttore
45          subplot(3,2,[5 6]);
46          semilogx(f, Ls_LCX200,'LineWidth',1.2);
47          grid on;
48          xlabel('Frequenza [Hz]');
49          ylabel('L_s [H]');
50          title('Induttanza serie');
51      else
52          subplot(3,2,[5 6]);
53          semilogx(f, Cs_LCX200,'LineWidth',1.2);
54          grid on;
55          xlabel('Frequenza [Hz]');
56          ylabel('C_s [F]');
57          title('Capacita' serie');
58      end
```

Script 3.3: Script MATLAB per il calcolo dei parametri serie

3.2 Frequency Response Analyzer del Moku:GO M2

Il secondo banco di misura è stato realizzato utilizzando il **Moku:GO di Liquid Instrument**, uno strumento multifunzione programmabile che integra diverse funzionalità di misura e analisi. L'architettura dello strumento si basa su un FPGA Zynq, che consente l'elaborazione dei segnali in tempo reale e la riconfigurazione dinamica delle funzionalità di misura. Nel caso in esame, lo strumento è stato configurato nella sua modalità di **Frequency Analyzer**, tramite la quale è possibile determinare la risposta in frequenza di un sistema, in termini di modulo e fase, mediante lo sweep in frequenza di un segnale sinusoidale. I principali parametri del segnale di eccitazione possono essere configurati in funzione delle caratteristiche del componente da analizzare, impostando in particolare l'ampiezza del segnale, l'intervallo di frequenze all'interno del quale effettuare lo sweep e il numero di punti di misura. Al fine di permettere un confronto diretto con i risultati ottenuti attraverso il banco di misura basato sull'LCX200, è stato mantenuto lo stesso intervallo di frequenza e numero di punti di misura, rispettivamente pari a

$$\Delta f = (100\,\mathrm{Hz}; 100\,\mathrm{kHz}), \qquad N = 4096.$$

Il Moku:GO dispone di quattro canali dotati di connettori BNC, di cui due configurati come uscite e due come ingressi. I canali di output vengono utilizzati per generare il segnale sinusoidale di eccitazione, mentre quelli di input consentono di acquisire i segnali ai capi degli elementi del circuito di misura. Il circuito di misura è stato realizzato su una breadboard collegando in serie il componente da caratterizzare a un resistore di valore noto, utilizzato come resistenza di shunt. Il valore della resistenza di shunt è stato opportunamente scelto in funzione del valore nominale del componente in esame, in modo da ottenere una risposta del circuito adeguatamente sensibile nell'intervallo di frequenze considerato. La configurazione delle connessioni tra i canali di ingresso e di uscita del Moku consente di misurare, a seconda della disposizione delle sonde, il rapporto tra la tensione ai capi del componente in esame e il segnale di eccitazione, oppure il rapporto tra la tensione ai capi del resistore di shunt e

il medesimo segnale. Tali grandezze costituiscono i dati di ingresso utilizzati nella successiva elaborazione per la determinazione dell'impedenza del componente.

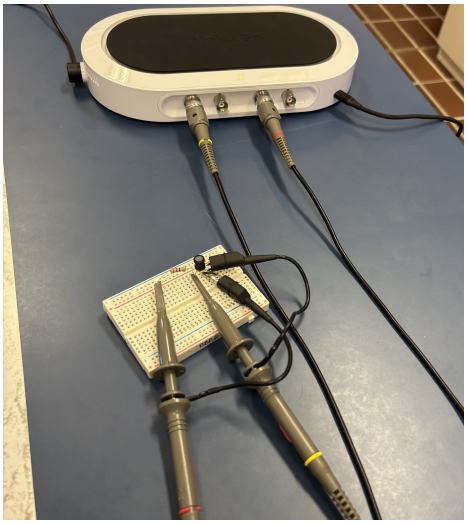


Figura 3.2: Banco di prova con Moku:GO.

3.2.1 Procedura di misura

La scelta della resistenza di shunt è stata effettuata tenendo conto della diversa natura dei componenti sottoposti a misura. In particolare, sono stati analizzati complessivamente sei condensatori e sei induttori, per i quali la presenza della resistenza di shunt in serie al componente determina un circuito equivalente a un filtro del primo ordine. A seconda della tipologia di componente e della configurazione dei canali di misura, è quindi possibile ottenere una risposta assimilabile a quella di un filtro passa-basso o passa-alto.

Per ciascuna delle due tipologie di componente, il valore della resistenza di shunt è stato scelto in modo da posizionare la frequenza di taglio del circuito in prossimità di 5 kHz. Tale frequenza è stata individuata come valore rappresentativo dell'intervallo di misura considerato, compreso tra 100 Hz e 100 kHz, in modo da disporre di un'adeguata estensione della risposta sia al di sotto sia al di sopra della frequenza di taglio.

Per i circuiti costituiti dal resistore e dal condensatore, la frequenza di taglio è espressa dalla relazione

$$f_c = \frac{1}{2\pi R_{\text{shunt}} C},$$

da cui, fissato f_c e noto il valore nominale della capacità, è stato ricavato il valore teorico della resistenza di shunt:

$$R_{\text{shunt}} = \frac{1}{2\pi f_c C}.$$

Analogamente, per i circuiti costituiti dal resistore e dall'induttore, la frequenza di taglio è data da

$$f_c = \frac{R_{\text{shunt}}}{2\pi L},$$

da cui

$$R_{\text{shunt}} = 2\pi f_c L.$$

I valori così calcolati sono stati confrontati con i valori commerciali disponibili e, per ciascun componente, è stata selezionata la resistenza nominale più prossima a quella teoricamente richiesta. Il valore effettivo di ciascun resistore di shunt è stato quindi misurato mediante l'LCX200 e utilizzato nella successiva elaborazione dei dati, così da tenere conto del valore reale della resistenza anziché di quello nominale.

Tabella 3.1: Valori dei resistori di shunt utilizzati

C_{nom} [µF]	R_{shunt} [Ω]	R_{nom} [Ω]	R_{real} [Ω]	L_{nom} [mH]	R_{shunt} [Ω]	R_{nom} [Ω]	R_{real} [Ω]
0.1	318.31	330	321.9	0.001	0.03	1	1.01
0.22	144.69	150	147.9	0.033	1.04	1	1.01
0.47	67.73	68	67.63	0.15	4.71	4.7	4.69
0.68	46.81	47	46.57	1	31.42	22	21.75
2.2	14.47	10	9.83	10	314.16	330	322.3
10	3.18	3.3	3.3	20	628.32	470	462.2

Una volta definita la resistenza di shunt, per ciascun componente sono state effettuate tre acquisizioni indipendenti della risposta in frequenza, mantenendo invariati i parametri di eccitazione e la configurazione del circuito. In funzione della tipologia di componente e della disposizione dei canali di misura, il Frequency Analyzer è stato configurato per acquisire il rapporto tra la tensione ai capi del componente e la tensione di eccitazione oppure il rapporto tra la tensione ai capi della resistenza di shunt e la tensione di eccitazione.

Per ciascuna acquisizione sono stati quindi registrati il modulo, espresso in dB, e la fase della funzione di trasferimento $H(f)$ nell'intero intervallo di frequenze considerato. Le tre acquisizioni relative a ciascun componente sono state successivamente elaborate mediante uno script sviluppato in MATLAB. Per ogni frequenza è stato calcolato il valore medio delle tre misure di modulo e di fase, ottenendo così una singola funzione di trasferimento rappresentativa del componente. Per valutare la ripetibilità della misura, le tre acquisizioni sono state confrontate a coppie, calcolando per il modulo e per la fase la differenza tra le rispettive funzioni di trasferimento. Per ciascun confronto sono stati quindi determinati il valore medio e la deviazione standard dell'errore sull'intero intervallo di frequenze. Valori medi prossimi a zero e deviazioni standard ridotte indicano una buona concordanza tra le acquisizioni e, quindi, una buona ripetibilità della procedura di misura.

A partire dal valore medio della funzione di trasferimento è stata quindi ricostruita la corrispondente grandezza complessa $H(f)$. Il modulo, inizialmente espresso in decibel, è stato convertito in valore lineare mediante

$$|H| = 10^{H_{\mathrm{dB}}/20},$$

mentre il valore complesso della funzione di trasferimento è stato ottenuto combinando modulo e fase:

$$H = |H|e^{j\varphi}.$$

La configurazione del circuito di misura è stata adattata in funzione del componente analizzato. Nel caso degli induttori, essa è mostrata in **Figura 3.3**.

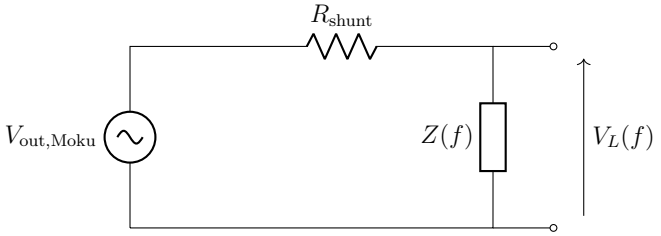


Figura 3.3: Circuito di misura degli induttori.

In questa configurazione, la tensione misurata dalla sonda di ingresso corrisponde alla tensione ai capi dell'induttore, $V_L(f)$. È stata quindi determinata la funzione di trasferimento

$$H(f) = \frac{V_L(f)}{V_{\text{out,Moku}}(f)},$$

dalla quale è possibile ricavare indirettamente l'impedenza complessa del componente. Considerando il circuito come un partitore di tensione, si ha infatti

$$H(f) = \frac{Z(f)}{R_{\text{shunt}} + Z(f)},$$

da cui

$$H(f)\left(R_{\text{shunt}} + Z(f)\right) = Z(f) \quad \Longrightarrow \quad Z(f) = R_{\text{shunt}}\frac{H(f)}{1 - H(f)}.$$

Nel caso dei condensatori, il circuito di misura è riportato in **Figura 3.4**.

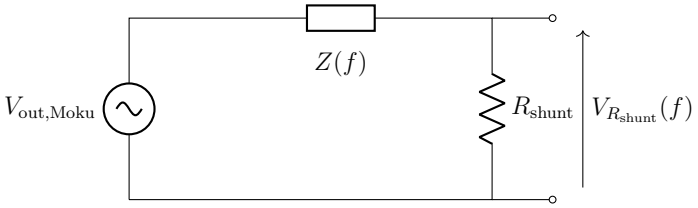


Figura 3.4: Circuito di misura dei condensatori.

In questa configurazione, la tensione misurata dalla sonda di ingresso corrisponde alla tensione ai capi della resistenza di shunt, $V_{R_{\text{shunt}}}(f)$. La funzione di trasferimento misurata dallo strumento è pertanto

$$H(f) = \frac{V_{R_{\text{shunt}}}(f)}{V_{\text{out,Moku}}(f)}.$$

Analogamente al caso degli induttori, considerando il circuito come un partitore di tensione, si ottiene

$$H(f) = \frac{R_{\text{shunt}}}{R_{\text{shunt}} + Z(f)}.$$

Da questa relazione segue

$$H(f)\,(R_{\text{shunt}} + Z(f)) = R_{\text{shunt}} \quad \Longrightarrow \quad Z(f) = R_{\text{shunt}} \frac{1 - H(f)}{H(f)}.$$

Le formulazioni ricavate sono state implementate direttamente in MATLAB utilizzando il valore reale della resistenza di shunt precedentemente misurato. Dall'impedenza ottenuta sono stati successivamente determinati il modulo $|Z(f)|$ e l'angolo di fase $\theta_Z(f) = \arg Z(f)$. In seguito, sono stati calcolati i parametri della rappresentazione serie del componente come nel caso del primo banco di misura:

$$\begin{cases} R_s = |Z| cos(\theta_z) \\ X_s = |Z| sin(\theta_z) \end{cases} \qquad \begin{cases} C_s = \dfrac{-1}{2\pi f X_s}, & X_s < 0 \\ L_s = \dfrac{X_s}{2\pi f}, & X_s > 0 \end{cases}$$

Per la rappresentazione grafica dei risultati, i dati sono stati sottoposti, dove necessario, a una procedura di filtraggio per ridurre il rumore presente nelle misure. In particolare, è stato inizialmente applicato un filtro a mediana mobile, seguito da un filtro di Savitzky–Golay, utilizzato per ottenere uno smoothing delle grandezze calcolate preservandone l'andamento locale.

```
% Variabili di calcolo
induttore = false;
solo_grafici = true;
filtro_denoising = true;
Rshunt = 321.9;  % Valore della resistenza di shunt utilizzata.

% Caricamento dati delle 3 misurazioni effettuate
load("a.csv");
load("b.csv");
load("c.csv");

```

```
12 f = a(:,1);
13
14 A{1} = a(:,2);
15 A{2} = b(:,2);
16 A{3} = c(:,2);
17 fase{1} = a(:,3);
18 fase{2} = b(:,3);
19 fase{3} = c(:,3);
20
21 % Calcolo della ripetibilita' delle misure
22 mediaA_moku = mean([A{1} A{2} A{3}],2);
23 stdA_moku   = std([A{1} A{2} A{3}],0,2);
24
25 % Escursione tra le misure ripetute
26 misureA = [A{1} A{2} A{3}];
27 escursione_A = max(misureA,[],2) - min(misureA,[],2);
28
29 % Errore relativo calcolato come:
30 % (escursione tra le misure / valore medio)x100
31 errore_relativo_A = 100 * escursione_A ./ abs(mediaA_moku);
32 mediaF_moku = mean([fase{1} fase{2} fase{3}],2);
33 stdF_moku   = std([fase{1} fase{2} fase{3}],0,2);
34
35 nomiA = {'errA12','errA13','errA21','errA23','errA31','errA32'};
36 nomiF = {'errF12','errF13','errF21','errF23','errF31','errF32'};
37
38 % Calcolo degli errori di modulo e fase per ogni possibile configurazione.
39 % Per ogni coppia di indici (i,j) viene valutata la differenza tra i vettori
40 % di ampiezza e fase, escludendo i casi in cui i=j
41
42 k = 1;
43 for i = 1:3
```

```
44      for j = 1:3
45          if i == j
46              continue
47          end
48              errA{k} = A{i} - A{j};
49              errF{k} = mod(fase{i} - fase{j} + 180,360) - 180;
50              assignin('base', nomiA{k}, errA{k});
51              assignin('base', nomiF{k}, errF{k});
52              k = k + 1;
53      end
54  end
55
56  % Generazione tabelle con gli errori e i parametri statistici.
57  if ~solo_grafici
58      T_errori = table(errA{1},errA{2},errA{3},errA{4},errA{5},errA{6}, ...
59                       errF{1},errF{2},errF{3},errF{4},errF{5},errF{6}, ...
60                       'VariableNames',...
61                       {'errA12','errA13','errA21','errA23','errA31','errA32',...
62                        'errF12','errF13','errF21','errF23','errF31','errF32'});
63      writetable(T_errori,...
64      "Vettori_Errore.csv");
65
66      T_stat = table(...
67              f,...
68              mediaA_moku,...
69              stdA_moku,...
70              escursione_A,...
71              errore_relativo_A,...
72              mediaF_moku,...
73              stdF_moku,...
74              'VariableNames',...
75              {'Frequenza',...
```

```
76          'Media_Modulo',...
77          'Std_Modulo',...
78          'Escursione_Modulo',...
79          'Errore_Relativo_Modulo_percentuale',...
80          'Media_Fase',...
81          'Std_Fase'});
82
83      writetable(T_stat,...
84          "Statistiche_Ripetibilita_Moku.csv");
85
86      save('statistiche_moku.mat',...
87          'f',...
88          'mediaA_moku',...
89          'mediaF_moku',...
90          'stdA_moku',...
91          'stdF_moku',...
92          'escursione_A',...
93          'errore_relativo_A');
94
95  save('media_moku.mat', 'f', 'mediaA_moku', 'mediaF_moku');
96  end
97
98  % Grafici ripetibilita'
99  figure;
100 semilogx(f,stdA_moku,'LineWidth',1.5)
101 grid on
102 xlabel('Frequenza [Hz]')
103 ylabel('\sigma modulo [dB]')
104 title('Ripetibilita' della misura del modulo')
105
106 figure
107 semilogx(f,errore_relativo_A,'LineWidth',1.5)
```

```
108 grid on
109 xlabel('Frequenza [Hz]')
110 ylabel('Errore relativo [%]')
111 title('Errore relativo del modulo')
112
113 figure
114 semilogx(f,stdF_moku,'LineWidth',1.5)
115 grid on
116 xlabel('Frequenza [Hz]')
117 ylabel('$\sigma$ fase  [$^\circ$]')
118 title('Ripetibilita' della misura di fase')
119
120 figure
121 semilogx(f,errA{1},'LineWidth',1.2)
122 hold on
123 semilogx(f,errA{2},'LineWidth',1.2)
124 semilogx(f,errA{4},'LineWidth',1.2)
125 grid on
126 xlabel('Frequenza [Hz]')
127 ylabel('Differenza modulo [dB]')
128 legend('M1-M2','M1-M3','M2-M3')
129 title('Differenze tra acquisizioni ripetute')
130
131 A_H_dB = mediaA_moku;        % Modulo in dB.
132 Phi_H_deg = mediaF_moku;     % Fase in gradi.
133
134 A_H = 10.^(A_H_dB / 20);
135 H = A_H .*exp(1j*deg2rad(Phi_H_deg));
136
137 % Calcolo impedenza incognita
138 if induttore
139    Zx = Rshunt .* H ./ (1-H);
```

```
140 else
141    Zx = Rshunt .* (1-H) ./ H;
142 end
143 
144 Modulo_Z = abs(Zx);
145 Theta_Z  = angle(Zx);
146 Theta_Z_unwrap = rad2deg(unwrap(Theta_Z));
147 
148 % Smoothing della fase
149 Theta_Z_f = sgolayfilt(movmedian(Theta_Z_unwrap,5),3,15);
150 Rs = real(Zx);
151 Xs = imag(Zx);
152 
153 % smoothing delle grandezze fondamentali
154 Rs_f = sgolayfilt(movmedian(Rs,5),3,15);
155 Xs_f = sgolayfilt(movmedian(Xs,5),3,15);
156 if induttore
157     Ls = Xs ./ (2*pi*f);
158     Ls_f = Xs_f ./ (2*pi*f);
159     if ~solo_grafici
160     save('Misure_Moku', 'f','Modulo_Z','Theta_Z_unwrap','Rs','Xs','Ls');
161     end
162  else
163     Cs = -1 ./ (2*pi*f.*Xs);
164     Cs_f = -1 ./ (2*pi*f.*Xs_f);
165     if ~solo_grafici
166     save('Misure_Moku', 'f','Modulo_Z','Theta_Z_unwrap','Rs','Xs','Cs');
167     end
168 end
169 
170 % grafici dei risultati ottenuti
171 figure('Name','Risposta in frequenza','Color','w');
```

```
172 if ~filtro_denoising
173     subplot(3,2,1);
174     semilogx(f, Modulo_Z,'LineWidth',1.2);
175     grid on;
176     xlabel('Frequenza [Hz]');
177     ylabel('|Z| [\Omega]');
178     title('Modulo dell''impedenza');
179     subplot(3,2,2);
180     semilogx(f, Theta_Z_unwrap,'LineWidth',1.2);
181     grid on;
182     xlabel('Frequenza [Hz]');
183     ylabel('$\theta$ [$^\circ$]');
184     title('Fase dell''impedenza');
185     subplot(3,2,3);
186     semilogx(f, Rs,'LineWidth',1.2);
187     grid on;
188     xlabel('Frequenza [Hz]');
189     ylabel('R_s [\Omega]');
190     title('Resistenza serie');
191     subplot(3,2,4);
192     semilogx(f, Xs,'LineWidth',1.2);
193     grid on;
194     xlabel('Frequenza [Hz]');
195     ylabel('X_s [\Omega]');
196     title('Reattanza serie');
197
198     if induttore
199         subplot(3,2,[5 6]);
200         semilogx(f, Ls,'LineWidth',1.2);
201         grid on;
202         xlabel('Frequenza [Hz]');
203         ylabel('L_s [H]');
```

```
204         title('Induttanza serie');
205     else
206         subplot(3,2,[5 6]);
207         semilogx(f, Cs,'LineWidth',1.2);
208         grid on;
209         xlabel('Frequenza [Hz]');
210         ylabel('C_s [F]');
211         title('Capacita' serie');
212     end
213 else
214     subplot(3,2,1);
215     semilogx(f, Modulo_Z,'LineWidth',1.2);
216     grid on;
217     xlabel('Frequenza [Hz]');
218     ylabel('|Z| [\Omega]');
219     title('Modulo dell''impedenza');
220     subplot(3,2,2);
221     semilogx(f, Theta_Z_f,'LineWidth',1.2);
222     grid on;
223     xlabel('Frequenza [Hz]');
224     ylabel('$\theta$ [$^\circ$]');
225     title('Fase dell''impedenza');
226     subplot(3,2,3);
227     semilogx(f, Rs_f,'LineWidth',1.2);
228     grid on;
229     xlabel('Frequenza [Hz]');
230     ylabel('R_s [\Omega]');
231     title('Resistenza serie');
232     subplot(3,2,4);
233     semilogx(f, Xs_f,'LineWidth',1.2);
234     grid on;
235     xlabel('Frequenza [Hz]');
```

```
236     ylabel('X_s [\Omega]');
237     title('Reattanza serie');
238     if induttore
239         subplot(3,2,[5 6]);
240         semilogx(f, Ls_f,'LineWidth',1.2);
241         grid on;
242         xlabel('Frequenza [Hz]');
243         ylabel('L_s [H]');
244         title('Induttanza serie');
245     else
246         subplot(3,2,[5 6]);
247         semilogx(f, Cs_f,'LineWidth',1.2);
248         grid on;
249         xlabel('Frequenza [Hz]');
250         ylabel('C_s [F]');
251         title('Capacita' serie');
252     end
253 end
```

Script 3.4: Script MATLAB che implementa la procedura di misura adottata nel banco basato sul Moku:GO

3.3 Analisi dei risultati sperimentali ottenuti con i due banchi di misura

Ripetibilità delle misure effettuate con il Moku:GO

Al fine di caratterizzare la ripetibilità delle misure effettuate mediante il Moku:Go, ciascun componente è stato sottoposto a tre acquisizioni indipendenti della funzione di trasferimento $H(f)$, descritte nella **Sezione 3.2.1**, indicate con $M1$, $M2$ e $M3$. Tali acquisizioni hanno permesso di calcolare alcuni indicatori, attraverso i quali è stata valutata la ripetibilità delle misure ottenute mediante lo strumento.

Tra gli indicatori considerati rientra lo scostamento del modulo di $H(f)$, questo in riferimento alle tre possibili coppie di acquisizioni ($M1 - M2$, $M1 - M3$ e $M2 - M3$). Tale scostamento è stato valutato tramite il calcolo delle differenze puntuali, in modulo, lungo l'intero intervallo di frequenze analizzato.

A partire dalle differenze è stato inoltre definito un indicatore sintetico della variabilità del modulo, denominato errore relativo del modulo, calcolato come rapporto tra l'escursione, ovvero la differenza tra il valore massimo e il valore minimo del modulo delle tre acquisizioni, e il relativo valore medio. Il calcolo è stato effettuato punto per punto in frequenza ed espresso in termini percentuali:

$$errore\,relativo\%(f) = 100\,\frac{\max_i |H_{M_i}(f)| - \min_i |H_{M_i}(f)|}{\frac{1}{3}\sum_{i=1}^{3} |H_{M_i}(f)|}.$$

A completamento di questa analisi, sono state calcolate anche la deviazione standard del modulo e quella della fase così da disporre di una descrizione statistica più completa della variabilità tra le acquisizioni, non limitata al solo modulo della funzione di trasferimento.

L'insieme di questi indicatori ha permesso di costruire, per ciascun valore nominale di capacità e induttanza considerato, una rappresentazione grafica articolata su quattro pannelli: le differenze tra le coppie di misure, l'errore relativo del modulo, la deviazione standard del modulo e la deviazione standard della fase, tutti espressi in funzione della frequenza. La lettura incrociata di questi grafici, riportati in **Figura 3.5** e **Figura 3.6**, permette di individuare gli intervalli di frequenza in cui la

misura presenta una maggiore ripetibilità, corrispondenti alle frequenze più elevate del range analizzato, e quelli in cui la variabilità tra le acquisizioni risulta lievemente maggiore, ovvero alle frequenze più basse. Tali considerazioni sono state ricavate dall'osservazione dell'andamento della deviazione standard del modulo e della fase, nonché dal grafico relativo alla differenza tra le acquisizioni.

L'analisi dei grafici mostra un comportamento sistematico e ricorrente, comune a entrambe le tipologie di componente: le curve di differenza del modulo tra le acquisizioni oscillano attorno allo zero simmetricamente e con un'ampiezza decrescente all'aumentare della frequenza.

Nel caso dei **condensatori**, i risultati mostrano una buona ripetibilità delle misure nella regione centrale dell'intervallo di frequenze, mentre la variabilità tende ad aumentare alle frequenze più elevate. In particolare, all'aumentare del valore di capacità, l'incremento dell'errore relativo tende a comparire a frequenze progressivamente più basse. Tale comportamento è coerente con la dipendenza dell'impedenza capacitiva dalla frequenza e dalla capacità, data da $|Z_C| = \dfrac{1}{2\pi f C}$.

Per i condensatori di capacità maggiore, infatti, l'impedenza del componente assume valori più bassi già a frequenze relativamente contenute e la funzione di trasferimento tende a $H(f) \approx 1$. In tale condizione, piccole variazioni nella misura di $H(f)$ possono determinare variazioni più significative dell'impedenza ricavata mediante la relazione $Z(f) = R_{\text{shunt}} \frac{1-H(f)}{H(f)}$, con conseguente aumento dell'errore relativo.

Alle frequenze più elevate diventano inoltre maggiormente rilevanti gli effetti parassiti del componente e del circuito di misura, come evidenziato dall'aumento della deviazione standard del modulo e della fase.

Nel caso degli **induttori**, i risultati evidenziano una dipendenza della ripetibilità dal valore nominale del componente e dalla frequenza di misura. Per i valori più bassi di induttanza, in particolare 1 µH, 33 µH e, in misura minore, 150 µH, si osservano differenze più marcate tra le acquisizioni, nonché valori elevati dell'errore relativo e della deviazione standard. Tale comportamento è riconducibile al ridotto rapporto segnale/rumore ($\frac{S}{N}$) associato a queste misure: per mantenere la frequenza di taglio

intorno a 5 kHz è infatti necessario utilizzare una resistenza di shunt di valore molto basso e, alle frequenze inferiori, la tensione misurata ai capi dell'induttore risulta particolarmente contenuta e quindi fortemente influenzata dal rumore e dagli effetti parassiti. Questo si ripercuote sulla misura rendendola non affidabile, in particolare quella associata all'induttanza di valore nominale $L_{nom} = 1\,\mu H$, mostrando i limiti del metodo di misura applicato relativamente ai valori nominali dei componenti analizzati. Per le induttanze maggiori alle frequenze più elevate si osserva un incremento della variabilità tra le acquisizioni, probabilmente riconducibile alla crescente influenza degli effetti parassiti del componente e del sistema di misura.

Nel complesso, i risultati ottenuti indicano che la ripetibilità delle misure con il Moku:Go non è costante sull'intera banda di frequenza esplorata: la misura presenta una maggiore affidabilità in una regione intermedia, mentre agli estremi dell'intervallo di frequenza questa tende a degradarsi. Inoltre, dai grafici viene evidenziata anche una forte dipendenza della ripetibilità dal valore assoluto dell'impedenza del dispositivo sotto test rispetto alla dinamica e alla risoluzione dello strumento. In particolare, la ripetibilità risulta massima laddove l'impedenza del componente si colloca in un intervallo intermedio, ben distinguibile sia dal rumore di fondo sia da eventuali limiti di saturazione dello strumento, mentre degrada progressivamente quando l'impedenza tende a valori molto piccoli.

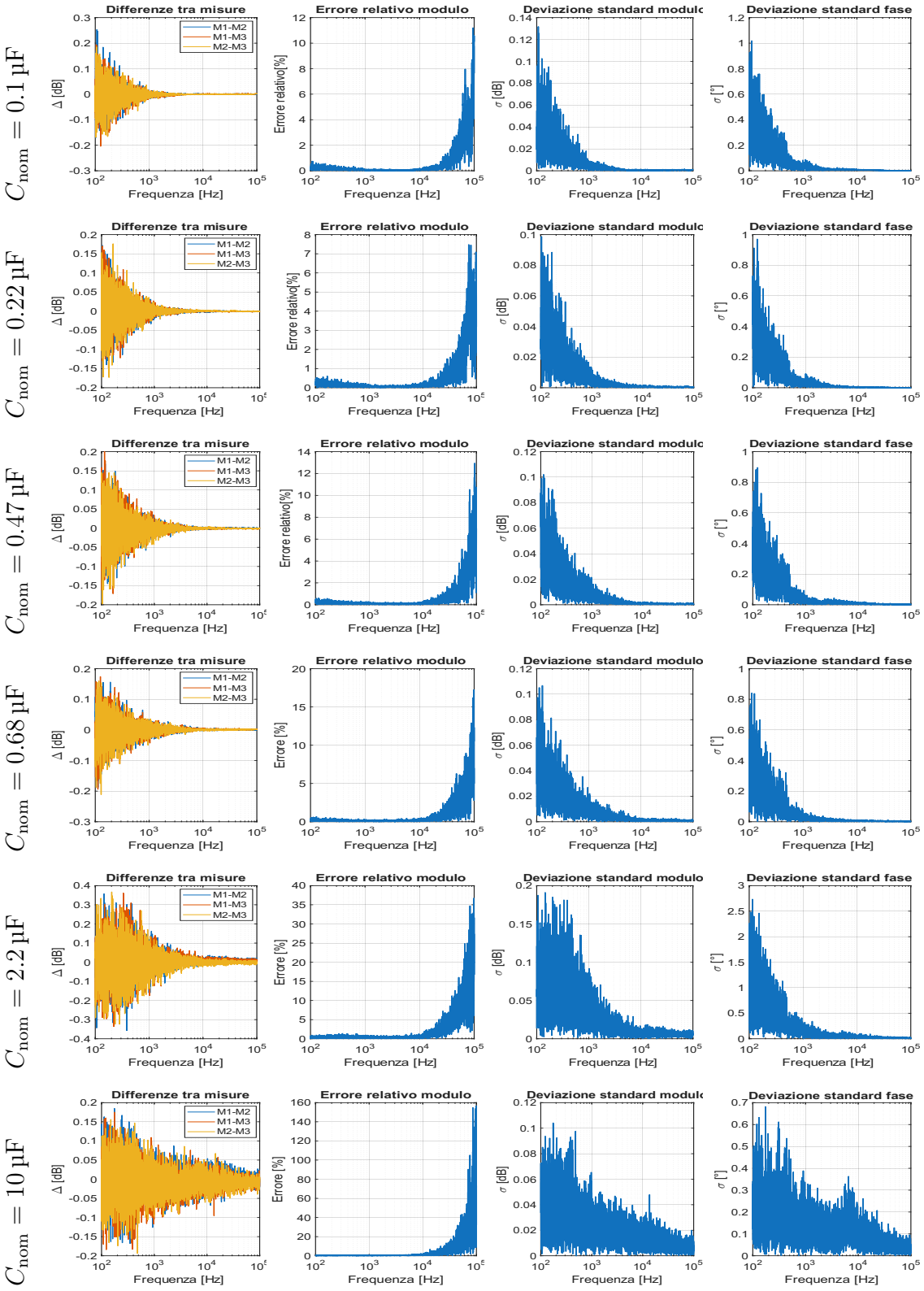


Figura 3.5: Risultati dell'analisi di ripetibilità delle misure sui condensatori.

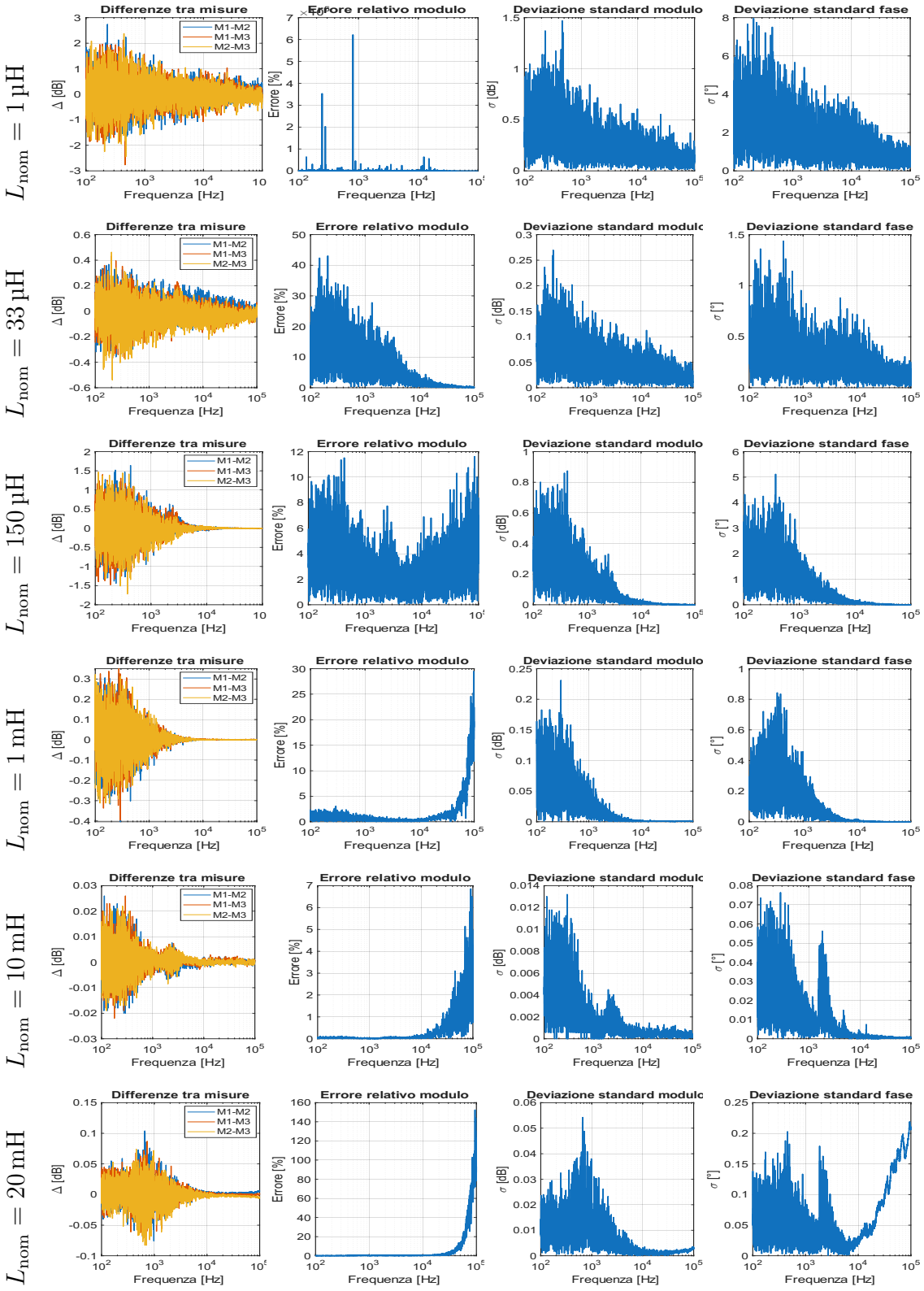


Figura 3.6: Risultati dell'analisi di ripetibilità delle misure sugli induttori.

Confronto sperimentale tra il Moku:GO e l'LCX200 per la misura dei parametri *R, C* e *L*

Il confronto tra le misure, ottenute mediante i due banchi di misura, è stato effettuato assumendo come strumento di riferimento l'LCX200. Attraverso i dati raccolti sul modulo $|Z|$ e fase θ_z dell'impedenza complessa di ciascun componente, sono stati calcolati anche gli opportuni parametri equivalenti serie R_s, C_s e L_s, applicando le relazioni descritte nelle sezioni precedenti. Poiché tali parametri sono stati ricavati indirettamente a partire da modulo e fase dell'impedenza e dipendono dalla specifica rappresentazione equivalente adottata, nel confronto tra i due banchi di misura sono stati considerati direttamente i dati di modulo e fase, al fine di preservare la generalità della trattazione. L'analisi comparativa dei risultati ottenuti è stata implementata in MATLAB tramite lo **Script 3.5**.

```
% Solo i campioni validi per entrambi gli strumenti vengono selezionati
% si evitano valori non finiti o NaN
iZ = isfinite(Modulo_Z) & isfinite(Z_LCX200);
e_Z_fin = Modulo_Z(iZ) - Z_LCX200(iZ);
iTheta = isfinite(Theta_Z_unwrap) & isfinite(Theta_LCX200);
e_Theta_fin = Theta_Z_unwrap(iTheta) - Theta_LCX200(iTheta);
e_Z_percentuale = ((Modulo_Z(iZ)-Z_LCX200(iZ))./Z_LCX200(iZ))*100;
N = length(e_Z_fin);
M = length(e_Theta_fin);

% Errore medio
e_Z_medio = mean(e_Z_fin);
e_Theta_medio = mean(e_Theta_fin);
e_Z_percent_medio = mean(e_Z_percentuale);

% Deviazione standard campionaria
e_Z_dev_std = sqrt(sum((e_Z_fin - e_Z_medio).^2)/(N-1));
e_Theta_dev_std = sqrt(sum((e_Theta_fin - e_Theta_medio).^2)/(M-1));
```

```
e_Z_percent_std = sqrt(sum((e_Z_percentuale-e_Z_percent_medio).^2)...
    /(length(e_Z_percentuale)-1));

% RMSE
RMSE_Z = sqrt(mean(e_Z_fin.^2));
RMSE_Z_percentuale = sqrt(mean(e_Z_percentuale.^2));
RMSE_Theta = sqrt(mean(e_Theta_fin.^2));

% Plotting dei dati
figure;
semilogx(f(iZ),e_Z_fin,'b','LineWidth',1.2);
grid on;
hold on;
yline(e_Z_medio,'r--','LineWidth',1.5);
xlabel('Frequenza [Hz]');
ylabel('Errore modulo [\Omega]');
title('Errore assoluto del modulo: Moku - LCX200');
legend('Errore modulo','Errore medio');

figure;
semilogx(f(iZ),e_Z_percentuale,'b','LineWidth',1.2);
grid on;
hold on;
yline(e_Z_percent_medio,'r--','LineWidth',1.5);
xlabel('Frequenza [Hz]');
ylabel('Errore relativo modulo [%]');
title('Errore relativo del modulo: Moku - LCX200');
legend('Errore relativo','Errore medio');

figure;
semilogx(f(iTheta),e_Theta_fin,'b','LineWidth',1.2);
grid on;
```

```
51 hold on;
52 yline(e_Theta_medio,'r--','LineWidth',1.5);
53 xlabel('Frequenza [Hz]');
54 ylabel('Errore fase [deg]');
55 title('Errore di fase: Moku - LCX200');
56 legend('Errore fase','Errore medio');
```

Script 3.5: Script MATLAB per il confronto grafico dei risultati

Le coppie di vettori, relative al modulo e alla fase dell'impedenza, vengono inizialmente caricate nel Workspace MATLAB così da essere in seguito elaborate per il confronto. L'elaborazione viene effettuata considerando esclusivamente i campioni per i quali risultano disponibili valori finiti per entrambi gli strumenti, escludendo quindi eventuali campioni caratterizzati da valori non numerici o non validi. In particolare, mediante le condizioni logiche *isfinite* vengono individuati separatamente i campioni validi relativi al modulo e fase dell'impedenza. I campioni validati, associati ad un determinato punto in frequenza, vengono utilizzati per il calcolo sia dell'errore assoluto sul modulo dell'impedenza, definito come differenza tra il valore riportato dal Moku:GO e quello ottenuto con l'LCX200, sia dell'errore relativo percentuale, mediante le seguenti relazioni

$$e_Z(f) = |Z(f)|_{\text{Moku}} - |Z(f)|_{\text{LCX200}}, \qquad e_Z(f)\% = \frac{|Z(f)|_{\text{Moku}} - |Z(f)|_{\text{LCX200}}}{|Z(f)|_{\text{LCX200}}} \cdot 100.$$

In modo analogo, l'errore sulla fase viene calcolato come

$$e_\theta(f) = \theta(f)_{\text{Moku}} - \theta(f)_{\text{LCX200}}.$$

L'utilizzo della fase sottoposta preliminarmente all'operazione di unwrap consente di rimuovere le discontinuità artificiali associate ai salti di 360° , rendendo più significativo il confronto tra le due misure lungo l'intero intervallo di frequenza considerato. In seguito vengono valutati gli indicatori statistici associati agli errori: il valore medio dell'errore fornisce un'indicazione dell'eventuale presenza di un offset sistematico rispetto al riferimento, mentre la deviazione standard quantifica la dispersione degli

errori attorno al loro valor medio. Successivamente, viene anche calcolato il *Root Mean Square Error* (*RMSE*), definito come

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_i^2},$$

che fornisce una misura complessiva dell'entità dell'errore, tenendo conto sia dell'eventuale componente sistematica sia della dispersione degli errori. Per il modulo vengono determinati sia l'*RMSE* espresso in ohm sia quello relativo percentuale, mentre per la fase l'*RMSE* è espresso in gradi. I dati raccolti sono stati organizzati nella **Tabella 3.2**.

Tabella 3.2: Indicatori statistici dell'errore di confronto tra Moku:GO e LCX200

Componente	Errore modulo [Ω]			Errore modulo [%]			Errore fase [°]		
	Media	σ	RMSE	Media	σ	RMSE	Media	σ	RMSE
0.1 µF	-34.59	63.24	72.07	0.44	1.79	1.85	0.02	0.19	0.19
0.22 µF	-33.56	59.32	68.15	-0.86	2.06	2.23	0.06	0.32	0.33
0.47 µF	-17.67	33.48	37.86	-0.95	2.03	2.24	-0.03	0.29	0.29
0.68 µF	-9.94	20.21	22.52	-0.21	2.09	2.10	-0.15	0.43	0.46
2.2 µF	0.56	3.13	3.18	2.18	1.45	2.61	-0.12	0.41	0.42
10 µF	1.71	2.47	3.00	3.45	3.12	4.65	-1.41	1.86	2.34
1 µH	19.84	26.02	32.71	70515.98	106958.71	128101.04	3513.42	1715.82	3909.92
33 µH	-1.54	2.29	2.76	-51.66	3.05	51.75	0.60	3.15	3.20
150 µH	0.15	0.32	0.35	1.73	2.78	3.27	-1.69	1.87	2.51
1 mH	1.60	3.49	3.84	1.29	0.70	1.47	-0.09	0.42	0.43
10 mH	407.83	1282.38	1345.52	12.71	23.14	26.40	0.61	0.88	1.07
20 mH	1763.20	8488.37	8668.55	17.48	72.61	74.68	-0.12	0.72	0.73

Infine, i risultati vengono rappresentati graficamente in funzione della frequenza, con l'asse delle ascisse in scala logaritmica. Vengono prodotti tre grafici distinti relativi, rispettivamente, all'errore assoluto del modulo, all'errore relativo percentuale del modulo e all'errore di fase. In ciascun grafico, oltre all'andamento dell'errore punto per punto, viene riportato mediante una linea tratteggiata il corrispondente valore medio. Tale rappresentazione consente di distinguere immediatamente eventuali deviazioni localizzate, variazioni dell'errore con la frequenza e possibili offset sistematici tra le misure del Moku e quelle del LCX200. I grafici sono stati organizzati in funzione del

valore nominale del componente analizzato e suddivisi tra componenti capacitivi e induttivi, come riportato rispettivamente in **Figura 3.7** e **Figura 3.8**.
È importante sottolineare che l'errore così definito è un errore di confronto rispetto al LCX200 e non costituisce, in senso assoluto, una misura dell'incertezza strumentale del Moku. L'analisi permette pertanto di valutare la coerenza delle misure ottenute con i due banchi e di individuare le regioni di frequenza o le condizioni di misura nelle quali il comportamento del sistema basato sul Moku si discosta maggiormente dal riferimento.

I risultati riportati in **Tabella 3.2** permettono di quantificare gli scostamenti tra le misure effettuate mediante il Moku:GO e quelle ottenute con l'LCX200. Per i componenti capacitivi, l'errore relativo del modulo presenta valori di *RMSE* compresi tra circa il 1.85% e il 4.65%, evidenziando una buona coerenza complessiva tra i due sistemi di misura. L'errore assoluto risulta invece maggiormente dipendente dal valore dell'impedenza del componente, passando da 72.07 Ω per il condensatore da 0.1 µF a 3.00 Ω per quello da 10 µF. Tale comportamento evidenzia l'importanza di considerare l'errore relativo, oltre a quello assoluto, per confrontare componenti caratterizzati da valori di impedenza differenti.
Anche l'errore di fase risulta generalmente contenuto per i condensatori, con valori di *RMSE* inferiori a 0.5° per i componenti compresi tra 0.1 µF e 2.2 µF. Un comportamento meno favorevole si osserva per il condensatore da 10 µF, per il quale l'*RMSE* di fase raggiunge 2.34°. Il comportamento degli induttori risulta invece più variabile. Escludendo il caso dell'induttore da 1 µH, che presenta valori anomali, l'errore relativo del modulo mostra *RMSE* compresi tra circa l' 1.47% e il 74.68%, mentre l'errore di fase presenta valori generalmente inferiori a pochi gradi. In particolare, i valori più elevati dell'errore relativo si osservano per gli induttori da 33 µH, 10 mH e 20 mH. È tuttavia opportuno osservare che un errore relativo elevato non implica necessariamente un corrispondente errore assoluto elevato, come evidenziato dal caso del componente da 33 µH, per il quale l'*RMSE* assoluto del modulo è pari a soli 2.76 Ω, a fronte di un *RMSE* relativo del 51.75%.

Questo risultato conferma l'utilità di considerare congiuntamente le due metriche nella valutazione delle prestazioni del sistema.

Per i condensatori, gli errori relativi di modulo risultano generalmente contenuti nelle regioni di frequenza in cui il rapporto segnale/rumore della misura è favorevole, mentre alle basse e alle alte frequenze si osservano maggiori oscillazioni e deviazioni rispetto al riferimento. Tali differenze possono essere ricondotte sia al rumore di misura sia alla crescente influenza delle non idealità dei componenti e del circuito di misura, che diventano particolarmente rilevanti agli estremi della banda considerata. Un comportamento analogo si osserva per l'errore di fase, generalmente più contenuto nelle regioni centrali della banda e caratterizzato da una maggiore variabilità alle frequenze più basse e più elevate.

Per gli induttori, il confronto risulta invece fortemente dipendente dalle condizioni di funzionamento del partitore e, in particolare, dalla scelta della resistenza di shunt impiegata. Il caso più evidente è quello dell'induttore da 1 µH, per il quale si osservano errori di modulo e di fase nettamente superiori rispetto agli altri componenti. Questo risultato non deve essere necessariamente interpretato come una scarsa accuratezza intrinseca del Moku, ma è principalmente riconducibile alle condizioni di misura adottate: per estendere la caratterizzazione dell'induttore fino a 100 kHz è stato necessario utilizzare una resistenza di shunt di valore molto ridotto, in modo da mantenere il partitore sensibile anche alle frequenze più elevate. Tuttavia, alle frequenze più basse l'impedenza dell'induttore da 1 µH diventa estremamente piccola e, conseguentemente, anche il segnale misurato attraverso il partitore risulta molto ridotto rispetto al rumore. In queste condizioni il rapporto segnale/rumore peggiora sensibilmente e l'errore sulla funzione di trasferimento $H(f) = \frac{V_L(f)}{V_{out}(f)}$ può tradursi in un errore molto maggiore sull'impedenza ricavata mediante $Z(f) = R_{\text{shunt}} \frac{H(f)}{1-H(f)}$. La relazione utilizzata introduce infatti una propagazione non lineare dell'errore: quando $H(f)$ assume valori prossimi a zero oppure prossimi all'unità, anche piccole variazioni dovute al rumore o alle incertezze di misura possono produrre variazioni significative

del valore calcolato di $Z(f)$. Il caso dell'induttore da 1 µH rappresenta pertanto una condizione limite della metodologia adottata, nella quale la scelta di un singolo valore di resistenza di shunt per coprire l'intera banda di frequenza porta inevitabilmente a un compromesso tra sensibilità alle basse e alle alte frequenze. Per gli induttori di valore maggiore, l'errore risulta progressivamente più gestibile nelle regioni in cui il partitore opera in condizioni favorevoli, anche se alle frequenze più elevate emergono deviazioni più marcate e andamenti non monotoni. Tali comportamenti possono essere associati non solo all'incertezza della misura, ma anche alla natura non ideale degli induttori reali, per i quali resistenza serie, capacità parassite e fenomeni prossimi alla frequenza di autorisonanza rendono l'impedenza complessa sempre meno rappresentabile mediante il solo modello induttivo ideale $Z = j\omega L$.
In particolare, per componenti con impedenza molto bassa, come l'induttore da 1 µH, sarebbe opportuno adottare una resistenza di shunt ottimizzata per sotto-intervalli di frequenza oppure una configurazione di misura che garantisca un rapporto segnale/rumore più elevato, riducendo così la propagazione dell'errore.

In conclusione, il confronto con l'LCX200 evidenzia una buona coerenza delle misure ottenute mediante il Moku:GO nelle condizioni di funzionamento più favorevoli, con scostamenti generalmente contenuti per la maggior parte dei componenti analizzati. Le principali criticità emergono in corrispondenza di componenti caratterizzati da impedenze particolarmente basse o elevate e alle estremità della banda di frequenza, dove il rapporto segnale/rumore e le non idealità del circuito assumono un ruolo più significativo. Il caso dell'induttore da 1 µH rappresenta la condizione più critica e mette in evidenza i limiti della configurazione adottata, senza tuttavia costituire necessariamente un'indicazione di una scarsa accuratezza intrinseca del Moku:GO. Nel complesso, i risultati confermano quindi la validità della metodologia proposta per la caratterizzazione dell'impedenza complessa, evidenziando al contempo come un'adeguata scelta della resistenza di shunt e della configurazione di misura siano fondamentali per ottenere prestazioni uniformi sull'intero intervallo di frequenza e di impedenza considerato.

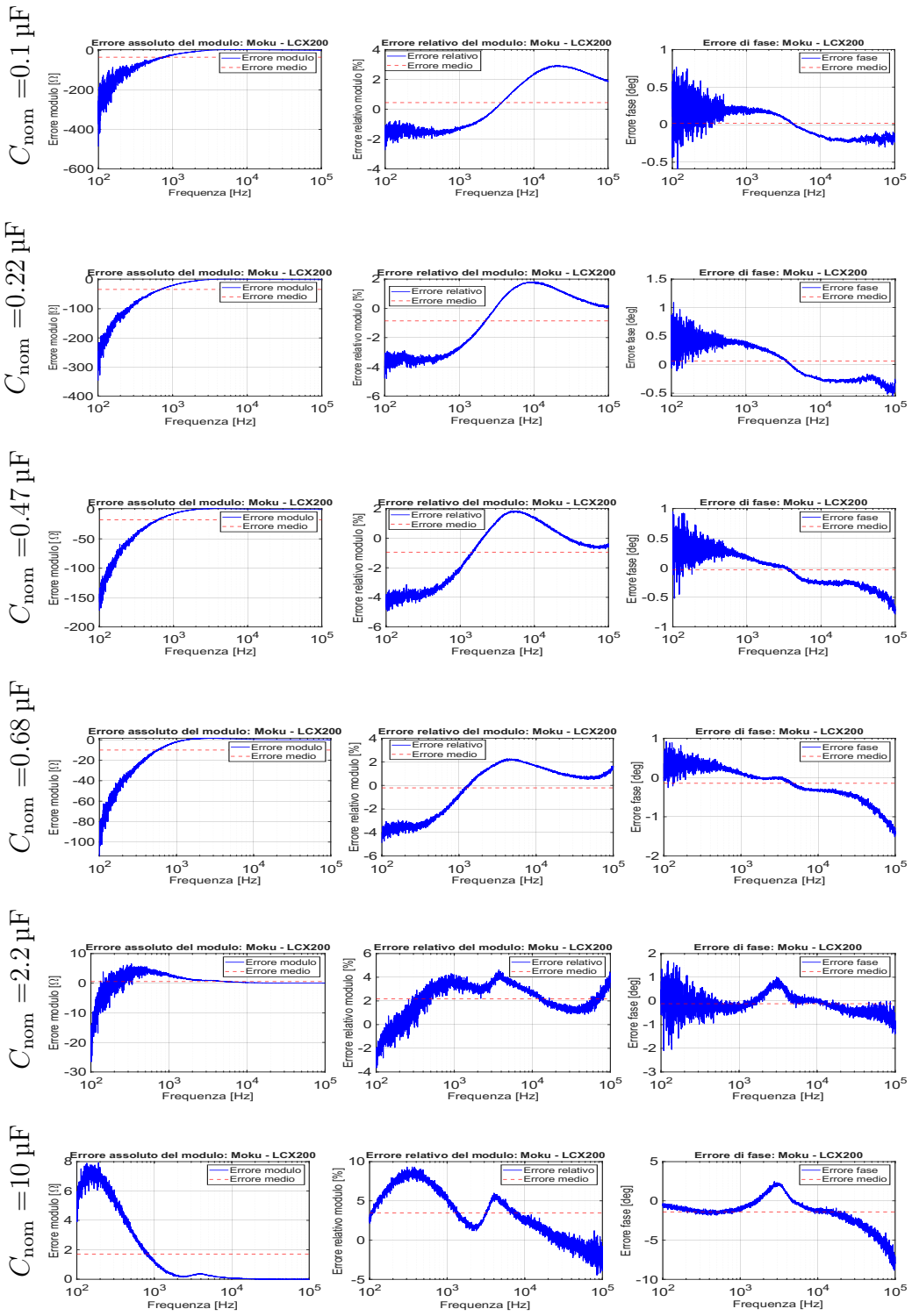


Figura 3.7: Confronto degli errori di misura per i diversi valori nominali di capacità.

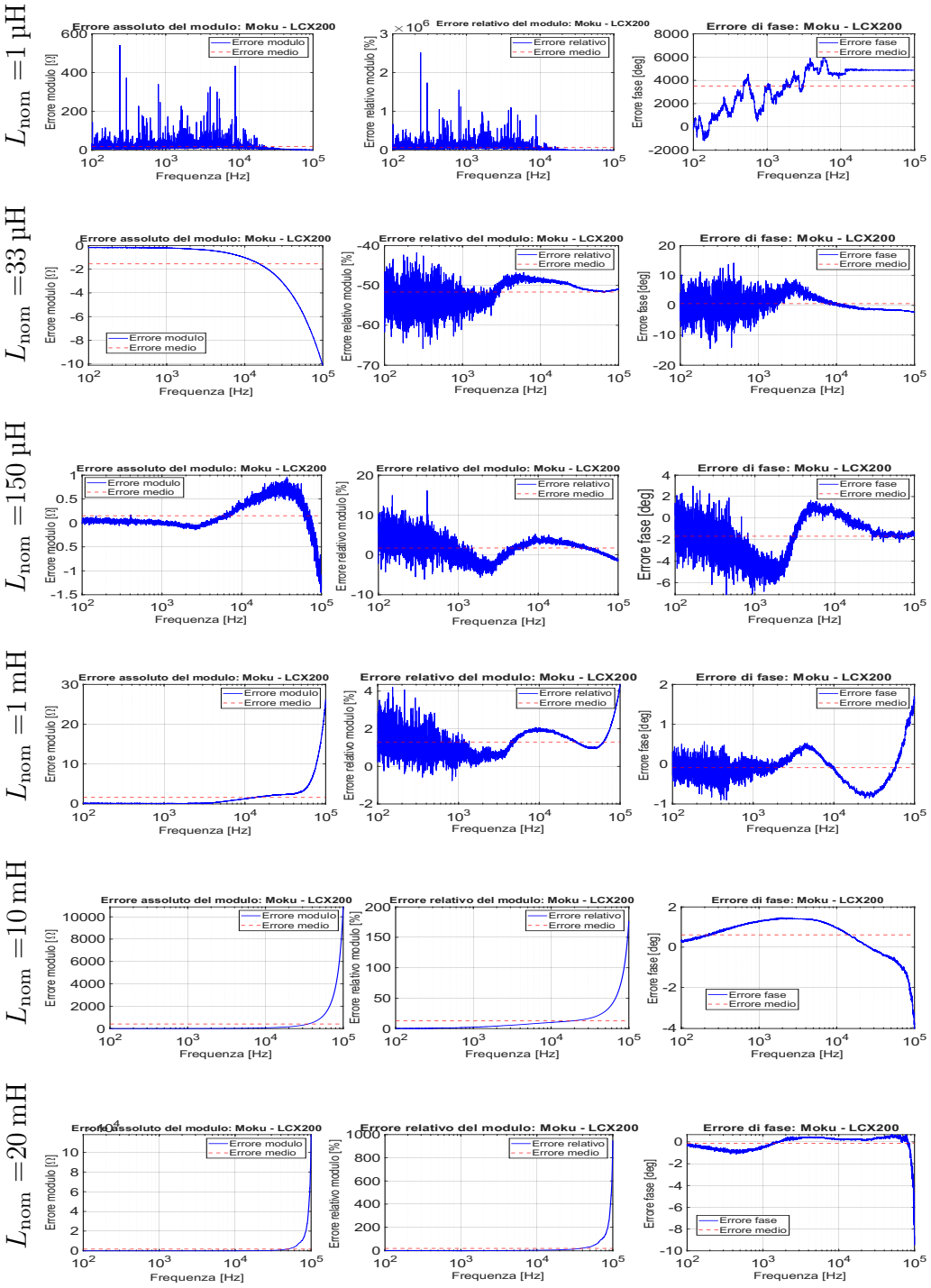


Figura 3.8: Confronto degli errori di misura per i diversi valori nominali di induttanza.

Capitolo 4

Implementazione di un *RCL* meter su ATMega328P

La realizzazione del sistema di misura è stata effettuata prendendo come riferimento il metodo proposto in [4], relativo allo sviluppo di un *RCL* meter a bassissimo costo. L'obiettivo è stato quello di riprodurre sperimentalmente il principio di misura descritto nell'articolo, basato sull'impiego di una scheda Arduino Uno, in particolare del convertitore analogico-digitale (ADC) integrato nel microcontrollore ATmega328P.
Una volta realizzato il sistema, ne sono state valutate sperimentalmente le prestazioni nella determinazione dei parametri elettrici di resistori e condensatori, nonché di reti *RC* nelle configurazioni serie e parallelo.
Infine, per la sola caratterizzazione della resistenza, i risultati ottenuti mediante il metodo proposto sono stati confrontati con quelli derivanti da un metodo alternativo basato sulla misura a ponte. Quest'ultimo utilizza la medesima scheda Arduino Uno, integrata con componenti elettronici esterni, quali resistori e amplificatori operazionali, al fine di realizzare la configurazione circuitale necessaria alla misura.

4.1 Descrizione del metodo

Il metodo proposto sfrutta le porte di ingresso/uscita generiche messe a disposizione dal microcontrollore, la cui configurazione può essere modificata via software in funzio-

ne della condizione di misura. Le caratteristiche elettriche interne delle porte vengono utilizzate per realizzare il circuito di misura, permettendone così l'implementazione senza ricorrere ad ulteriori componenti esterni.

In particolare, tale circuito è costituito dalle due porte analogiche $A0$ e $A3$, indicate rispettivamente come P_{A0} e P_{A1}, tra le quali viene collegato il componente o la rete da caratterizzare. La configurazione delle porte determina il circuito elettrico equivalente utilizzato durante la misura, mentre la tensione presente sulla porta P_{A0} viene acquisita mediante il convertitore analogico-digitale integrato nel microcontrollore.

Le porte I/O possono essere configurate secondo quattro differenti modalità operative, la cui combinazione permette di realizzare le diverse configurazioni richieste dal metodo: ingresso ad alta impedenza (*floating input*), ingresso con resistenza di pull-up interna (*pull-up input*), uscita a livello basso (*low output*) e uscita a livello alto (*high output*).

Per le misure considerate nel presente lavoro, la configurazione di base prevede P_{A0} impostata come ingresso con resistenza di pull-up interna e P_{A1} come uscita a livello basso. In tale configurazione, il componente sotto esame costituisce l'impedenza di carico Z_{LOAD} del circuito, mentre le caratteristiche elettriche interne delle porte contribuiscono a definirne il comportamento complessivo.

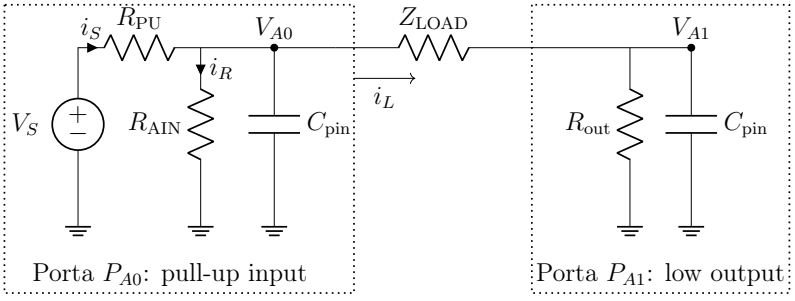


Figura 4.1: Modello elettrico delle porte P_{A0} e P_{A1}.

Il circuito equivalente risultante tiene conto, oltre che dell'impedenza da caratterizzare, dei principali elementi parassiti e delle resistenze interne associate alle porte del microcontrollore. Tra questi rientrano la resistenza di pull-up R_{PU}, la resistenza di uscita R_{OUT}, la resistenza di ingresso analogico R_{AIN} e la capacità associata al pin C_{PIN}. Tali parametri devono essere considerati nell'elaborazione della misura,

in quanto influenzano la relazione tra la tensione acquisita dall'ADC e il parametro elettrico da determinare.

Nel presente lavoro sono state considerate la misura di una resistenza isolata, la misura di una capacità isolata mediante l'analisi del transitorio e la caratterizzazione di reti RC nelle configurazioni serie e parallelo. Sebbene la configurazione circuitale di base rimanga invariata, come quella mostrata in **Figura 4.1**, il parametro da determinare viene ricavato mediante differenti modalità di eccitazione e di acquisizione, in funzione della natura dell'impedenza collegata tra le due porte.

4.1.1 Misura di una resistenza isolata

Nel caso di una resistenza isolata, l'impedenza di carico Z_{LOAD} è costituita dalla sola resistenza R_{LOAD}. La porta P_{A0} viene configurata come ingresso con resistenza di pull-up, mentre P_{A1} viene mantenuta come uscita a livello basso. In queste condizioni, la resistenza di pull-up R_{PU} fornisce la corrente che attraversa il percorso resistivo costituito dalla resistenza di ingresso analogico R_{AIN} e dalla serie tra R_{LOAD} e la resistenza di uscita R_{OUT}. Per la misura in continua, le capacità parassite C_{PIN} possono essere considerate circuiti aperti dopo l'attesa di un intervallo di tempo sufficiente ($\Delta t \geq 5\tau$). La tensione V_{A0} raggiunge pertanto il valore di regime e viene acquisita mediante l'ADC. Indicando con N_{A0} il valore digitale restituito dal convertitore, si ha

$$N_{A0} = \left\lfloor N_{\max} \frac{V_{A0}}{V_S} \right\rfloor .$$

La relazione tra la tensione misurata e la resistenza incognita può essere ricavata mediante l'equivalente resistivo del circuito. Introducendo la variabile ausiliaria

$$K = R_{\mathrm{PU}} \frac{V_{A0}}{V_S - V_{A0}},$$

il valore della resistenza da misurare è ottenuto considerando anche le resistenze interne del microcontrollore:

$$R_{\mathrm{LOAD}} = \frac{R_{\mathrm{AIN}} R_{\mathrm{OUT}} - K\left(R_{\mathrm{AIN}} + R_{\mathrm{OUT}}\right)}{K - R_{\mathrm{AIN}}}.$$

La procedura consente quindi di ricavare il valore della resistenza a partire dalla sola tensione V_{A0} misurata in condizioni stazionarie. La relazione riportata tiene

esplicitamente conto di R_{PU}, R_{AIN} e R_{OUT}, evitando di assumere ideale il circuito interno delle porte.

4.1.2 Misura di una capacità isolata

La misura di una capacità isolata C_{LOAD} viene effettuata attraverso l'analisi del processo di carica del condensatore. Prima dell'acquisizione, le capacità presenti nel circuito vengono completamente scaricate configurando entrambe le porte come uscite a livello basso e mantenendo tale configurazione per un intervallo di tempo sufficientemente lungo. Il lavoro originale prevede due differenti modalità per la misura della capacità isolata: una modalità di acquisizione rapida, indicata come *fast acquisition mode*, e una modalità basata sull'analisi del transitorio, indicata come *transient acquisition mode*. La prima sfrutta la risposta immediata del circuito durante la carica e risulta particolarmente adatta alle capacità più basse; la seconda determina invece il tempo necessario al raggiungimento di una determinata soglia di tensione ed è utilizzata principalmente per capacità più elevate.

Nel presente lavoro viene considerata la misura mediante analisi del transitorio. Dopo la fase di scarica, P_{A0} viene configurata come ingresso con resistenza di pull-up, mentre P_{A1} viene mantenuta come uscita a livello basso. L'applicazione del gradino di tensione determina quindi la carica della capacità attraverso la rete resistiva equivalente del circuito.

L'evoluzione temporale della tensione V_{A0} presenta un andamento esponenziale. In forma generale, la risposta può essere espressa come

$$V_{A0}(t) = V_F + (V_i - V_F)\, e^{-t/\tau},$$

dove V_i e V_F rappresentano rispettivamente la tensione iniziale e quella finale, mentre τ è la costante di tempo del circuito. La procedura implementata nel microcontrollore sfrutta la soglia logica del circuito TTL per individuare un istante caratteristico del transitorio. Viene misurato il tempo $\Delta t = \tau$ trascorso dall'applicazione del gradino fino al raggiungimento della soglia e, nello stesso istante ($t = t_\tau$) viene acquisito il va-

lore V_{A0} mediante l'ADC. Dalla combinazione dei due valori e dal modello equivalente del circuito è quindi possibile determinare la capacità mediante l'espressione

$$C_{\mathrm{LOAD}} = \frac{\Delta t}{R_T \ln(\frac{V_S}{V_S - V_{A0}(t_\tau)})}.$$

dove $V_S \approx 5\,\mathrm{V}$ è la tensione di riferimento fornita da Arduino.

Limiti del metodo

La modalità *transient acquisition* è principalmente adatta a capacità superiori a circa 10 nF e, nel lavoro di riferimento, presenta tipicamente un'accuratezza relativa $a_r < 20\%$ nell'intervallo 10 nF–10 mF, con una incertezza relativa $p_r < 5\%$ da 100 pF a 10 mF. Per capacità inferiori diventano più rilevanti la risoluzione temporale, le capacità parassite e la variabilità della soglia TTL. Il metodo risulta quindi complementare alla modalità *fast*, che è invece più adatta a capacità inferiori a circa 10 nF [4].

4.1.3 Caratterizzazione di una rete *RC* serie

Nel caso di una rete RC serie, l'impedenza di carico è costituita dalla serie tra R_{LOAD} e C_{LOAD}. La caratterizzazione della rete viene effettuata analizzando il valore iniziale della tensione e il successivo transitorio di carica del condensatore. Dopo aver scaricato il condensatore, P_{A0} viene configurata come ingresso con resistenza di pull-up, mentre P_{A1} viene mantenuta come uscita a livello basso. All'istante iniziale, il condensatore può essere considerato come un cortocircuito e la tensione V_{A0} assume quindi un valore V_R determinato dal partitore costituito da R_{PU}, R_{AIN} e dalla serie tra R_{LOAD} e R_{OUT}. A partire dal valore V_R acquisito dall'ADC viene determinato il parametro

$$K = R_{\mathrm{PU}} \frac{V_R}{N_{\max} - V_R},$$

dal quale si ricava il valore della resistenza incognita:

$$R_{\mathrm{LOAD}} = \frac{R_{\mathrm{AIN}}(K - R_{\mathrm{OUT}})}{R_{\mathrm{AIN}} - K}.$$

Una volta determinato R_{LOAD}, è possibile valutare la resistenza equivalente R_T vista dal condensatore durante il transitorio. Tale resistenza è costituita dalla parallela tra R_{PU} e R_{AIN}, in serie con R_{LOAD} e R_{OUT}, ed è quindi espressa da

$$R_T = (R_{\mathrm{PU}} \parallel R_{\mathrm{AIN}}) + R_{\mathrm{LOAD}} + R_{\mathrm{OUT}}.$$

L'applicazione del gradino di tensione determina quindi la carica del condensatore attraverso la resistenza equivalente R_T. Il transitorio della tensione V_{A0} è caratterizzato dalla costante di tempo $\tau = R_T C_{\mathrm{LOAD}}$. Il tempo caratteristico del transitorio viene determinato mediante l'acquisizione della tensione V_{A0} e, una volta ricavata la costante di tempo, il valore della capacità viene ottenuto analogamente al caso isolato mediante

$$C_{\mathrm{LOAD}} = \frac{\Delta t}{R_T \ln(\frac{V_S}{V_S - V_{A0}(t_\tau)})}$$

Limiti del metodo

Nel caso della rete RC serie, l'accuratezza dipende dalla combinazione dei valori di R_{LOAD} e C_{LOAD}. Il lavoro di riferimento evidenzia tre regioni particolarmente critiche: per $R_{\mathrm{LOAD}} > 1\,\mathrm{M\Omega}$ la misura della capacità tende a peggiorare, mentre per $R_{\mathrm{LOAD}} < 100\,\mathrm{k\Omega}$ la determinazione della resistenza risulta meno accurata, soprattutto per $C_{\mathrm{LOAD}} > 1\,\mathrm{nF}$. Per $R_{\mathrm{LOAD}} < 100\,\mathrm{k\Omega}$ e $C_{\mathrm{LOAD}} < 0.1\,\mathrm{nF}$ l'accuratezza di entrambe le grandezze risulta ulteriormente penalizzata. Il metodo presenta pertanto un campo di funzionamento ottimale dipendente dalla posizione della rete nel piano $(R_{\mathrm{LOAD}}, C_{\mathrm{LOAD}})$ [4].

4.1.4 Caratterizzazione di una rete *RC* parallelo

Nel caso di una rete RC parallelo, l'impedenza di carico è costituita dalla combinazione di una resistenza R_{LOAD} e di una capacità C_{LOAD} collegate in parallelo. La procedura di misura sfrutta il valore della tensione raggiunta a regime per determinare la componente resistiva e l'analisi del transitorio di carica per ricavare quella capacitiva. In una prima fase, P_{A0} viene portata a livello basso e successivamente configurata come ingresso con resistenza di pull-up. Prima dell'acquisizione della

tensione di regime viene considerato un valore massimo della capacità $C_{\max}$, utilizzato per stimare la massima costante di tempo del circuito:

$$\tau_{\max} = (R_{\text{AIN}} \parallel R_{\text{PU}})\, C_{\max}.$$

Il circuito viene quindi lasciato evolvere per un intervallo pari a circa $5\tau_{\max}$, al termine del quale la tensione V_F acquisita dall'ADC viene assunta come valore di regime. Il valore di V_F viene utilizzato per determinare la resistenza R_{LOAD}. A partire dalla tensione acquisita viene calcolato il parametro

$$K = R_{\text{PU}} \frac{V_F}{N_{\max} - V_F},$$

dal quale si ricava

$$R_{\text{LOAD}} = \frac{(R_{\text{AIN}} + R_{\text{OUT}})K - R_{\text{AIN}} R_{\text{OUT}}}{R_{\text{AIN}} - K}.$$

Una volta determinato R_{LOAD}, viene calcolata la resistenza equivalente R_T vista dal condensatore durante il transitorio. Essa è data dal parallelo tra la resistenza equivalente di R_{AIN} e R_{PU} e il ramo costituito dalla serie tra R_{OUT} e R_{LOAD}:

$$R_T = (R_{\text{AIN}} \parallel R_{\text{PU}}) \parallel (R_{\text{OUT}} + R_{\text{LOAD}})\,.$$

Il condensatore viene quindi nuovamente scaricato portando P_{A0} a livello basso. Successivamente, la porta viene configurata come ingresso con resistenza di pull-up e viene avviata la misura del transitorio mediante il timer interno del microcontrollore. La costante di tempo τ viene in seguito stimata misurando l'intervallo di tempo, denominato T, necessario affinché la tensione acquisita dall'ADC raggiunga il 63.22% del valore finale V_F. Il valore della capacità viene infine determinato mediante

$$C_{\text{LOAD}} = \frac{T}{R_T}.$$

Limiti del metodo

Per la rete RC parallelo, l'accuratezza dipende dalla corretta determinazione sia della tensione di regime V_F sia della costante di tempo τ. Il metodo richiede inoltre un

tempo di acquisizione dell'ordine di alcune costanti di tempo; nel lavoro di riferimento il valore di regime viene assunto dopo circa 5τ, mentre l'acquisizione complessiva può estendersi fino a circa 10τ. Valori estremi di R_{LOAD} e C_{LOAD} possono quindi ridurre la sensibilità della misura e aumentare l'influenza della quantizzazione dell'ADC e della risoluzione temporale. Anche in questo caso, le migliori prestazioni si ottengono in specifiche regioni del piano $(R_{\mathrm{LOAD}}, C_{\mathrm{LOAD}})$ [4].

4.2 Implementazione dei metodi di misura e analisi dei risultati sperimentali

4.2.1 Resistenza isolata e taratura dei parametri resistivi.

Prima dell'implementazione delle procedure di misura è stata effettuata una fase preliminare di taratura, necessaria per determinare i valori effettivi dei principali parametri elettrici interni alle porte del microcontrollore R_{PU}, R_{OUT} e R_{AIN}.
La procedura di taratura è stata effettuata utilizzando la configurazione impiegata per la misura di una resistenza isolata, mediante il codice Arduino nello **Script 4.4**, dove sono stati considerati solo i valori di N_{A0} forniti dall'ADC. Sono stati selezionati 31 valori nominali di resistenza, utilizzati come campioni di riferimento e caratterizzati preliminarmente mediante un LCX200, così da ottenere per ciascun campione il corrispondente valore reale R_{real}.
Per ciascuna resistenza, il circuito di misura è stato controllato mediante Arduino e i valori restituiti dal convertitore analogico-digitale sono stati acquisiti automaticamente tramite MATLAB attraverso la porta seriale. Per ogni campione sono state effettuate 1000 acquisizioni del valore digitale N_{A0}. A partire dai campioni acquisiti sono stati calcolati il valore medio e la deviazione standard. Il procedimento descritto è stato implementato in MATLAB attraverso lo **Script 4.1**.

```
1 clear;
2 clc;
3 % Cambia COM con quella del tuo Arduino
```

```
port = "COM5";
baud = 115200;
% apertura seriale
s = serialport(port, baud);
% tempo per stabilizzare la comunicazione
pause(2);
% numero campioni
N = 1000;
NA0 = zeros(N,1);
for k = 1:N
    linea = readline(s);
    NA0(k) = str2double(linea);
end
media_NA0 = mean(NA0);
dev_std = std(NA0);
fprintf("Media ADC = %.4f\n",media_NA0);
fprintf("Deviazione standard = %.4f\n",dev_std);
clear s;
```

Script 4.1: Codice MATLAB per la lettura dei valori forniti dall'ADC

La procedura è stata ripetuta per tutti i valori di resistenza considerati, ottenendo così per ciascun componente una terna di valori costituita dal valore reale R_{real} del campione, dal valore medio $\overline{N}_{A0}$ e dalla deviazione standard $\sigma_{N_{A0}}$ (**Tabella 4.1**). Tali dati sono stati successivamente utilizzati per stimare i parametri elettrici interni della scheda.

I tre parametri R_{PU}, R_{OUT} e R_{AIN} sono stati quindi determinati mediante la funzione di ottimizzazione, fornita dall'articolo e descritta nello **Script 4.2**, basata sul metodo dei minimi quadrati ordinari (*Ordinary Least Squares*, OLS).

```
function [RAIN,RPU,ROUT] = fittingOLS2R( NMAX , NA0 , RNOMINAL , guessR )
RAIN = 100E6;
```

```
RPU = 32E3;
ROUT = 600;
    if nargin == 3
    guessR = [ RPU ROUT RAIN ];
    end
func = @(R,A0) ( R(1) .* A0 ./ ( NMAX - A0 ) .* ( R(2) + R(3) ) - R(2) * R
    (3) ) ./ ( R(3) - R(1) .* A0 ./ ( NMAX - A0 ) );
A0 = NA0;
y = RNOMINAL;
OLS = @(R) sum( ( log( func( R , A0 ) ) - log( y ) ) .^ 2 );
opts = optimset( 'MaxFunEvals' , 600000 , 'MaxIter' , 100000 );
[ RFIT , err ] = fminsearch( OLS , guessR , opts ) ;
RPU = RFIT(1);
ROUT = RFIT(2);
RAIN = RFIT(3);
end
```

Script 4.2: funzione MATLAB OLSR

La minimizzazione della funzione obiettivo è stata eseguita numericamente mediante l'algoritmo *fminsearch* di MATLAB, utilizzando come valori iniziali delle stime i valori nominali indicati dal datasheet del microcontrollore.
Dalla funzione di fitting sono stati estrapolati i valori associati ai parametri resistivi

$$R_{\mathrm{PU}} = 35.697\,\mathrm{k\Omega} \qquad R_{\mathrm{OUT}} = 541.9869\,\Omega \qquad R_{\mathrm{AIN}} = 2.4958\,\mathrm{M\Omega}$$

che saranno in seguito utilizzati nelle successive configurazioni di misura, sostituendo i valori nominali dei parametri interni con quelli effettivamente caratteristici della scheda utilizzata. Successivamente, è stato sviluppato il codice Arduino che implementa la misura di resistenza isolata descritta nella **Sezione 4.1.1**, quest'ultimo riportato nello **Script 4.4**. Infine, viene calcolata la resistenza misurata (R_{mis}), l'errore relativo ($a_r = \frac{|R_{mis}-R_{real}|}{R_{real}}$) e l'incertezza relativa ($p_r = \frac{|\frac{dR}{dN}|\sigma_N}{R_{mis}}$). I risultati ottenuti sono riportati in **Tabella 4.1** , mentre dalla rappresentazione grafica, effettuata mediante lo **Script 4.3**, si ottengono le curve di **Figura 4.2**.

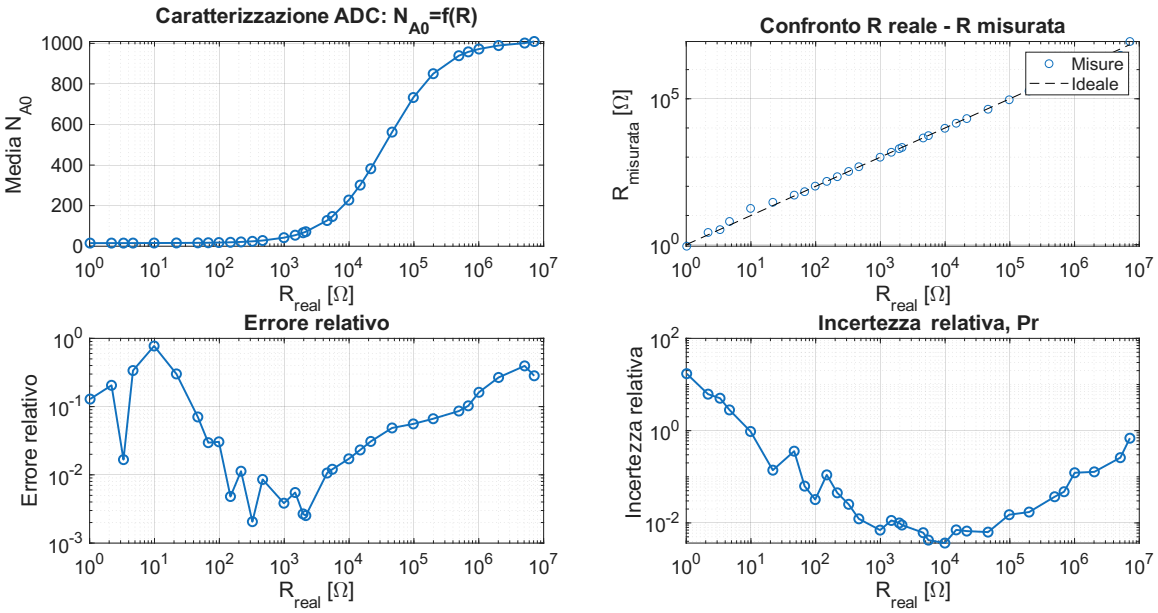


Figura 4.2: Rappresentazione grafica dei dati sui resistori.

Tabella 4.1: Risultati ottenuti dalle misure sui resistori isolati

R_{real}	$\overline{N}_{A0}$	$\sigma_{N_{A0}}$	$R_{\text{mis}} \pm u_{R_{\text{mis}}}$	R_{real}	$\overline{N}_{A0}$	$\sigma_{N_{A0}}$	$R_{\text{mis}} \pm u_{R_{\text{mis}}}$
1.02 Ω	15.23	0.42	0.9±0.5 Ω	2.154 kΩ	71.32	0.48	2.150±0.001 kΩ
2.17 Ω	15.28	0.45	2.6±0.6 Ω	4.585 kΩ	126.70	0.61	4.536±0.001 kΩ
3.33 Ω	15.30	0.46	3.3±0.6 Ω	5.522 kΩ	146.32	0.48	5.455±0.001 kΩ
4.66 Ω	15.38	0.48	6.2±0.6 Ω	9.915 kΩ	227.50	0.61	9.745±0.001 kΩ
9.86 Ω	15.69	0.46	17.5±0.6 Ω	14.834 kΩ	301.25	1.4	14.491±0.003 kΩ
21.78 Ω	15.99	0.11	28.3±0.2 Ω	21.618 kΩ	381.91	1.5	20.951±0.004 kΩ
46.57 Ω	16.58	0.49	49.9±0.6 Ω	46.045 kΩ	562.53	1.6	43.806±0.009 kΩ
67.47 Ω	17.01	0.11	65.5±0.2 Ω	97.853 kΩ	732.96	3.0	92.40±0.05 kΩ
98.38 Ω	18.00	0.09	101.4±0.2 Ω	198.40 kΩ	850.47	2.3	185.2±0.2 kΩ
148.34 Ω	19.27	0.45	147.6±0.6 Ω	493 kΩ	939.14	2.5	450.6±0.6 kΩ
215.80 Ω	21.07	0.26	213.4±0.4 Ω	689 kΩ	958.09	4.8	618±1 kΩ
323.25 Ω	24.05	0.22	322.6±0.3 Ω	1.005 MΩ	972.11	3.0	0.842±0.003 MΩ
464.5 Ω	28.00	0.16	468.5±0.2 Ω	2.020 MΩ	989.35	3.0	1.481±0.006 MΩ
990.1 Ω	41.97	0.18	993.9±0.3 Ω	5.10 MΩ	1001.56	2.7	3.09±0.03 MΩ
1.483 kΩ	54.23	0.42	1.468±0.001 kΩ	7.16 MΩ	1009.18	2.7	9.2±0.2 MΩ
1.954 kΩ	66.37	0.48	1.949±0.001 kΩ				

Analisi dei risultati

I risultati ottenuti mostrano che la procedura di misura presenta un buon accordo con i valori di riferimento nella regione centrale del campo di misura. In particolare,

per valori di resistenza compresi indicativamente tra alcune centinaia di ohm e alcune decine di kΩ, l'errore relativo rimane generalmente contenuto e inferiore al 3%, per la maggior parte dei campioni.

Per valori di resistenza inferiori a circa 50 Ω, l'errore aumenta sensibilmente. Tale comportamento è dovuto principalmente al fatto che il valore medio di N_{A0} rimane prossimo al limite inferiore del campo di misura dell'ADC, riducendo la sensibilità della misura. Un comportamento analogo si osserva per le resistenze di valore più elevato, per le quali N_{A0} si avvicina al fondo scala. In questa regione l'errore aumenta progressivamente, raggiungendo valori dell'ordine del 40% per i valori di resistenza più elevati analizzati.

Anche l'andamento dell'incertezza relativa risulta coerente con quanto osservato per l'errore: i valori minimi si riscontrano nella regione centrale del campo di misura, mentre aumentano in prossimità dei due estremi. Dalla caratteristica $N_{A0} = f(R)$ si osserva inoltre il progressivo aumento del valore dell'ADC all'aumentare della resistenza, fino all'avvicinamento al fondo scala. Il confronto tra R_{mis} e R_{real} mostra una buona corrispondenza nella regione centrale, mentre gli scostamenti aumentano progressivamente verso gli estremi del campo.

Gli andamenti ottenuti sperimentalmente risultano inoltre coerenti, dal punto di vista qualitativo, con quelli riportati nell'articolo di riferimento, sia per la caratteristica dell'ADC sia per l'andamento dell'errore e dell'incertezza relativa.

Nel complesso, la caratterizzazione effettuata ha permesso di individuare il campo nel quale la configurazione adottata presenta le migliori prestazioni e di evidenziarne i principali limiti.

```
clear;
clc;
load('R_DATI_FINALI_taratura.mat');
[RAIN,RPU,ROUT] = fittingOLS2R( 1023, media_NA0_taratura , Rreal);
% Calcolo della Rstimata
NMAX = 1023;
```

```
7  K = (RPU .* media_NA0_taratura) ./ (NMAX - media_NA0_taratura);
8  R_stimata = ((RAIN + ROUT).*K - RAIN*ROUT) ./ (RAIN - K);
9  % Calcolo modulo errore relativo
10 ar = abs(R_stimata - Rreal) ./ Rreal;
11 % Calcolo pr
12 dKdN = RPU*NMAX ./ (NMAX-media_NA0_taratura).^2;
13 dRdK = RAIN^2 ./ (RAIN-K).^2;
14 dRdN = dRdK .* dKdN;
15 sigma_media_NA0 = dev_std_taratura ./ sqrt(1000);
16 sigma_R = abs(dRdN).*sigma_media_NA0;
17 pr = abs(dRdN).*sigma_R ./ R_stimata;
18 %grafici
19 figure;
20 subplot(2,2,1);
21 semilogx(Rreal,media_NA0_taratura,'o-','LineWidth',1.5,'MarkerSize',7);
22 grid on;
23 xlabel('R_{real} [\Omega]');
24 ylabel('Media N_{A0}');
25 title('Caratterizzazione ADC: N_{A0}=f(R)');
26 subplot(2,2,2);
27 loglog(Rreal,R_stimata,'o');
28 hold on;
29 loglog(Rreal,Rreal,'k--');
30 grid on;
31 xlabel('R_{real} [\Omega]');
32 ylabel('R_{misurata} [\Omega]');
33 title('Confronto R reale - R misurata');
34 legend('Misure','Ideale');
35 subplot(2,2,3);
36 loglog(Rreal,ar,'o-','LineWidth',1.5,'MarkerSize',7);
37 grid on;
38 xlabel('R_{real} [\Omega]');
```

```
ylabel('Errore relativo');
title('Errore relativo');
subplot(2,2,4);
loglog(Rreal,pr,'o-','LineWidth',1.5,'MarkerSize',7);
grid on;
xlabel('R_{real} [\Omega]');
ylabel('Incertezza relativa');
title('Incertezza relativa, Pr');
```

Script 4.3: Codice MATLAB per l'analisi grafica dei dati sui resitori

```
#include <SPI.h>
#define PA0 A0
#define PA1 A3
#define NMAX 1023.0
#define RAIN 2.4958E6
#define RPU 3.5697E4
#define ROUT 541.9869
#define RxTx 115200
void setup() {
Serial.begin( RxTx );
 pinMode( PA0, OUTPUT );
digitalWrite( PA0, LOW );
 pinMode( PA1, OUTPUT );
digitalWrite( PA1, LOW );
delay( 1000 );
pinMode( PA0, INPUT_PULLUP );
delay(1);
}
void loop() {
float NA0 = analogRead( PA0 );
float K = ( RPU * NA0 ) / ( NMAX - NA0 );
float R_LOAD = ( ( RAIN + ROUT ) * K - RAIN * ROUT ) / ( RAIN - K );
```

```
23 Serial.print(NA0);
24 // le due linee di codice seguenti sono state commentate
25 // nella fase di taratura, in quanto non utilizzate
26 Serial.print(",");
27 Serial.println(R_LOAD);
28 }
```

Script 4.4: Codice Arduino per l'implementazione di un R meter

4.2.2 Capacità isolata

La procedura di misura di una capacità isolata, descritta precedentemente nella **Sezione 4.1.2** , è stata implementata tramite Arduino Uno mediante lo **Script 4.5**, nel quale sono stati utilizzati i parametri resistivi determinati nel processo di taratura.

```
1  #include <SPI.h>
2  #define PA0 A0
3  #define PA1 A3
4  #define NMAX 1023.0
5  #define RAIN 2.4958E6
6  #define RPU 3.5697E4
7  #define ROUT 541.9869
8  #define RxTx 115200
9  void setup() {
10 Serial.begin( RxTx );
11 pinMode( PA0, OUTPUT );
12 digitalWrite( PA0, LOW );
13 pinMode( PA1, OUTPUT );
14 digitalWrite( PA1, LOW );
15 delay( 1000 );
16 }
17
18 void loop() {
```

```
19 pinMode( PA0 , INPUT_PULLUP );
20 unsigned long TS = micros();
21 int NA0 = 0;
22 do { NA0 = digitalRead(PA0);
23 } while (NA0 < 1.0);
24 unsigned long TE = micros();
25 NA0 = analogRead( PA0 );
26 pinMode( PA0, OUTPUT );
27 digitalWrite( PA0, LOW );
28 float Tau = (TE-TS)*1E-6;
29 float RT = ROUT + ( RAIN * RPU ) / ( RAIN + RPU );
30 float C_LOAD = Tau / ( RT * log( NMAX / ( NMAX - NA0 ) ) );
31 Serial.print(Tau);
32 Serial.print(",");
33 Serial.print(NA0);
34 Serial.print(",");
35 Serial.println(C_LOAD,12);
36 delay( Tau * 1000 );
37 }
```

Script 4.5: Codice Arduino per l'implementazione di un C meter

Il codice riportato implementa la procedura di misura descritta nella **Sezione 4.1.2**. In fase di inizializzazione, le porte P_{A0} e P_{A1} vengono configurate come uscite e mantenute a livello basso per consentire la scarica delle capacità presenti nel circuito. L'intervallo temporale associato al transitorio viene determinato mediante le funzioni *micros()*, rilevando l'istante di superamento della soglia logica attraverso *digitalRead()*.

Successivamente, il valore di N_{A0} viene acquisito mediante il convertitore ADC tramite l'apposito comando *analogRead()*. A partire da queste due grandezze, il programma calcola la costante di tempo τ, la resistenza equivalente R_T e infine il valore di C_{LOAD} secondo l'espressione descritta nella **Sezione 4.1.2**.

I valori di τ, N_{A0} e C_{LOAD} vengono infine trasmessi alla porta seriale per la successiva elaborazione dei dati in MATLAB.

La verifica sperimentale del metodo è stata realizzata analizzando la risposta acquisita dalla misura di 10 condesatori campioni, caratterizzati da differenti valori nominali:

$$C_{\text{nom}} = \{0.1\,\mu\text{F};\ 0.15\,\mu\text{F};\ 0.22\,\mu\text{F};\ 0.33\,\mu\text{F};\ 0.47\,\mu\text{F};\ 0.68\,\mu\text{F};\ 1\,\mu\text{F};\ 2.2\,\mu\text{F};\ 4.7\,\mu\text{F};\ 10\,\mu\text{F}\}$$

La procedura di misura è stata gestita interamente mediante MATLAB tramite lo **Script 4.6**, nel quale sono state effettuate un totale di $N = 1000$ prove ripetute per ciascun campione, al fine di elaborare in seguito i dati raccolti.

```
clear;
clc;
load('Dati_C\Parametri_R_ARDUINO.mat'); % caricamento Rout,Rain e Rpu
C0_nF = 0.09;                    % Capacita parassita a vuoto
NMAX = 1023;
port = "COM5";                   % Porta COM dell' Arduino
baud = 115200;                   % Velocita di comunicazione della seriale
s = serialport(port, baud);      % Apertura della porta seriale
pause(2);
N = 1000;                        % Numero campioni
Tau = zeros(N,1);
NA0 = zeros(N,1);
C_load = zeros(N,1);
k = 1;                          % Acquisizione dati delle prove
while k <= N
    linea = readline(s);
    values = str2double(split(linea,","));
    if length(values)==3         % Controllo sulla presenza dei tre valori
        Tau(k)=values(1);
        NA0(k)=values(2);
        C_load(k)=values(3);
        k=k+1;
    end
```

```
24 end
25 clear s;                              % Chiusura connessione seriale
26 mean_C_load = mean(C_load);           % Valore medio delle N prove
27 std_C_load = std(C_load);             % Deviazione standard delle N prove
28 uA_C_load = std_C_load / sqrt(N);     % Calcolo incertezza di tipo A
29 mean_C_load_nF = mean_C_load * 1e9;
30 std_C_load_nF = std_C_load * 1e9;
31 uA_C_load_nF = uA_C_load * 1e9;
32
33 fprintf('C = %.3f nF +/- %.3f nF\n',mean_C_load_nF-C0_nF,uA_C_load_nF);
34 save('Misura_C.mat', "mean_C_load_nF","mean_C_load","C_load","uA_C_load_nF
    ","C_load","std_C_load_nF","std_C_load","NA0","Tau","uA_C_load_nF","
    uA_C_load");
```

Script 4.6: Codice MATLAB per la gestione della misura di capacità

In seguito all'acquisizione dei dati associati a ciascun campione, analogamente a quanto effettuato per le misure resistive, sono stati calcolati l'errore relativo $a_r = \dfrac{|C_{\text{load}} - C_{\text{nom}}|}{C_{\text{nom}}}$ rispetto al corrispondente valore nominale e l'incertezza relativa $p_r = \dfrac{\sigma_{C_{\text{load}}}}{C_{\text{load}}}$, mediante il codice MATLAB riportato nello **Script 4.7**.
I risultati sperimentali ottenuti sono stati quindi raccolti nella **Tabella 4.2**.
Nel medesimo script sono stati inoltre realizzati i grafici relativi alle principali grandezze analizzate, riportati in **Figura 4.3**.

Tabella 4.2: Risultati ottenuti dalle misure sui condensatori isolati

C_{real}	$\overline{C}_{\text{LOAD}}$	$\sigma_{C_{\text{LOAD}}}$	$C_{\text{mis}} \pm u_{C_{\text{mis}}}$	C_{real}	$\overline{C}_{\text{LOAD}}$	$\sigma_{C_{\text{LOAD}}}$	$C_{\text{mis}} \pm u_{C_{\text{mis}}}$
95.26 nF	93.26 nF	0.42 nF	93.26±0.02 nF	644.51 nF	649.02 nF	1.50 nF	649.02±0.05 nF
144.31 nF	140.960 nF	0.25 nF	140.960±0.008 nF	996.8 nF	1005.91 nF	1.536 nF	1005.91±0.05 nF
217.7 nF	213.65 nF	0.48 nF	213.65±0.02 nF	2.063 µF	2.0912 µF	0.0029 µF	2.0912±0.0001 µF
316 nF	316.69 nF	0.408 nF	316.69±0.02 nF	4.250 µF	4.4891 µF	0.0252 µF	4.4891±0.0008 µF
441.6 nF	449.75 nF	0.832 nF	449.75±0.03 nF	7.100 µF	8.3110 µF	0.0163 µF	8.3110±0.0006 µF

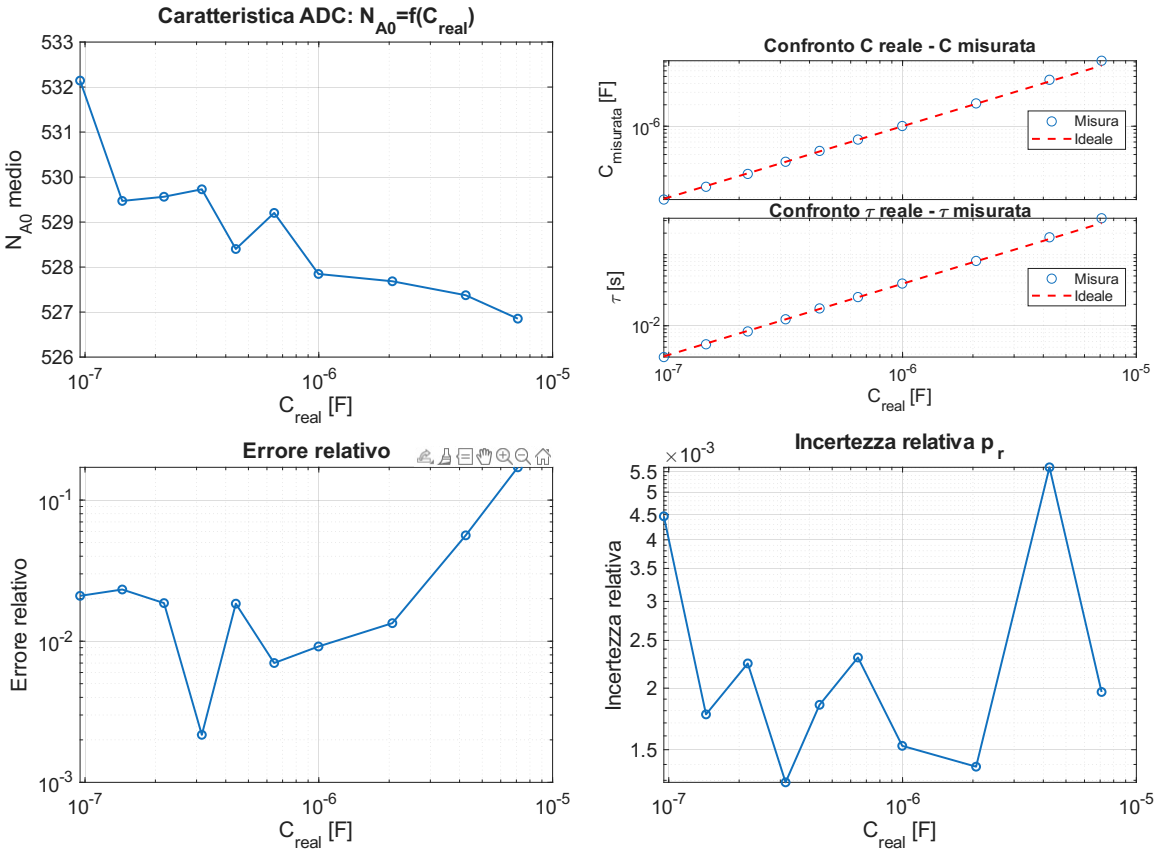


Figura 4.3: Rappresentazione grafica dei dati sui condensatori.

Analisi dei risultati

I risultati ottenuti mostrano una corrispondenza complessivamente buona tra la capacità reale e quella misurata. Lo scostamento relativo rispetto al valore reale varia da circa -2.3% a $+17.1\%$, rimanendo inferiore al 3% per le capacità fino a circa $1\,\mu F$ e aumentando per i valori più elevati, in particolare per i condensatori da $4.25\,\mu F$ e $7.1\,\mu F$. Il confronto tra capacità reale e misurata evidenzia comunque un andamento prossimo a quello ideale, mentre il tempo di transitorio cresce all'aumentare della capacità, come previsto dal modello di carica esponenziale. Nel complesso, il comportamento osservato è in accordo qualitativo con quanto riportato nell'articolo di riferimento per la modalità di acquisizione mediante transitorio, nella quale è atteso un aumento della variabilità dell'accuratezza al variare della capacità. I risultati ottenuti seguono il comportamento atteso dal metodo, pur evidenziando una maggiore deviazione dai valori reali alle capacità più elevate.

```
clear;
clc;
load("Dati_C\Vettori.mat");                    % Caricamento dati N prove
load("Dati_C\Parametri_R_ARDUINO.mat");        % Rout, Rain e Rpu
Vettori_NA0 = table2array(Vettori_NA0);
Vettori_C_load_nF = table2array(Vettori_C_load_nF);
Vettori_Tau_sec = table2array(Vettori_Tau_sec);
NMAX = 1023;
N = 1000;                                      % numero di prove
nC= 10;                                        % numero di C
C_load = zeros(nC,1);
std_C_load = zeros(nC,1);
uA_C_load = zeros(nC,1);

% Elaborazione dati e calcolo parametri statistici
for i = 1:nC
    NA0 = Vettori_NA0(:,i);
    Tau = Vettori_Tau_sec(:,i);
    RT = ROUT + ( RAIN * RPU ) / ( RAIN + RPU );
    C_i = Tau ./ ( RT * log( NMAX ./ ( NMAX - NA0 ) ) );
    C_load(i) = mean(C_i);              % Valore medio di Cload
    std_C_load(i) = std(C_i);                  % Dev. standard di Cload
    uA_C_load(i) = std_C_load(i)/sqrt(N);      % Incertezza di tipo A
end
C_load_nF = C_load * 1e9;
std_C_load_nF = std_C_load * 1e9;
uA_C_load_nF = std_C_load_nF/sqrt(N);
errore_rel = (C_load_nF - Creal_nF)./Creal_nF;  % Errore relativo
errore_rel_abs = abs(errore_rel);              % Modulo err. relativo
P_R = std_C_load_nF ./ C_load_nF;              % Incertezza relativa
% Grafici
```

```
figure('Color','w');
set(gcf,'Units','normalized','Position',[0.05 0.05 0.9 0.88]);
set(groot,'defaultAxesFontSize',15,'defaultTextFontSize',15);
subplot(2,2,1);
semilogx(Creal_nF*1e-9,NA0_mean,'o-','LineWidth',1.5);
grid on; box on;
xlabel('C_{real} [F]'); ylabel('N_{A0} medio');
title('Caratteristica ADC: N_{A0}=f(C_{real})');

subplot(2,2,2);
pos=get(gca,'Position'); delete(gca);
pos(4)=pos(4)/2;
axes('Position',[pos(1),pos(2)+pos(4),pos(3),pos(4)]);
loglog(Creal_nF*1e-9,C_load_nF*1e-9,'o','MarkerSize',7);
hold on;
loglog(Creal_nF*1e-9,Creal_nF*1e-9,'--r','LineWidth',1.5);
grid on; box on;
ylabel('C_{misurata} [F]');
title('Confronto C reale - C misurata');
legend('Misura','Ideale','Location','best');
set(gca,'XTickLabel',[]);
axes('Position',pos);
loglog(Creal_nF*1e-9,RT*C_load_nF*1e-9,'o','MarkerSize',7);
hold on;
loglog(Creal_nF*1e-9,RT*Creal_nF*1e-9,'--r','LineWidth',1.5);
grid on; box on;
xlabel('C_{real} [F]'); ylabel('\tau [s]');
title('Confronto \tau reale - \tau misurata');
legend('Misura','Ideale','Location','best');

subplot(2,2,3);
loglog(Creal_nF*1e-9,errore_rel_abs,'o-','LineWidth',1.5);
```

```
grid on; box on;
xlabel('C_{real} [F]'); ylabel('Errore relativo');
title('Errore relativo');

subplot(2,2,4);
loglog(Creal_nF*1e-9,P_R,'o-','LineWidth',1.5);
grid on; box on;
xlabel('C_{real} [F]'); ylabel('Incertezza relativa');
title('Incertezza relativa p_r');
```

Script 4.7: Codice MATLAB per l'elaborazione dei dati sulla misura di capacità

4.2.3 Rete *RC* serie

La procedura di caratterizzazione di una rete *RC* serie, descritta precedentemente nella **Sezione 4.1.3**, è stata implementata tramite Arduino Uno mediante lo **Script 4.8**, nel quale sono stati utilizzati contemporaneamente i principi alla base delle misurazioni di capacità e di resistenza isolate.

```
#include <SPI.h>
#define PA0 A0
#define PA1 A3
#define NMAX 1023.0
#define RAIN 2.4958E6
#define RPU 3.9E4
#define ROUT 541.2036
#define RxTx 115200
void setup() {
Serial.begin( RxTx );
pinMode( PA0, OUTPUT );
digitalWrite( PA0, LOW );
pinMode( PA1, OUTPUT );
digitalWrite( PA1, LOW );
```

```
15 delay( 1000 );
16 }
17
18 void loop() {
19 pinMode( PA0 , INPUT_PULLUP );
20 int VR = analogRead( PA0 );
21 unsigned long TS = micros();
22 float VREF = NMAX - VR;
23 float VTAU = VR + 0.6322 * VREF;
24 int VC = 0;
25 do {
26  VC = analogRead( PA0 );
27 } while ( VC < VTAU );
28 unsigned long TE = micros();
29 pinMode( PA0, OUTPUT );
30 digitalWrite( PA0, LOW );
31 float Tau = (TE-TS)*1E-6;
32 float K = RPU * VR / ( NMAX - VR );
33 float R_LOAD = ( ( RAIN + ROUT ) * K - RAIN * ROUT ) / ( RAIN - K );
34 float RT = R_LOAD + ROUT + ( RAIN * RPU / ( RAIN + RPU ) );
35 float C_LOAD = Tau / ( RT * log((NMAX - VR) / (NMAX - VC)) );
36 Serial.print(VR);
37 Serial.print(",");
38 Serial.print(VC);
39 Serial.print(",");
40 Serial.print(R_LOAD);
41 Serial.print(",");
42 Serial.print(C_LOAD,12);
43 Serial.print(",");
44 Serial.print(RT);
45 Serial.print(",");
46 Serial.println(Tau,10);
```

```
47 delay( Tau * 1E3 );
48 }
```

Script 4.8: Codice Arduino per l'RC serie meter

Il codice riportato implementa la procedura di misura descritta nella **Sezione 4.1.3**. In fase di inizializzazione, le porte P_{A0} e P_{A1} vengono configurate come uscite e mantenute a livello basso per consentire la scarica del condensatore. Successivamente, P_{A0} viene configurata come ingresso con pull-up e viene acquisito il valore iniziale V_R mediante il convertitore ADC. A partire da tale valore viene determinata la soglia corrispondente al raggiungimento del 63.22% del transitorio, utilizzata per la misura dell'intervallo temporale tramite le funzioni *micros()* e *analogRead()*.
Il programma calcola quindi la costante di tempo τ, la resistenza di carico equivalente R_{LOAD}, la resistenza totale R_T e infine il valore della capacità C_{LOAD} secondo le espressioni riportate nel medesimo paragrafo sopra citato. I valori acquisiti e calcolati vengono infine trasmessi alla porta seriale per la successiva elaborazione dei dati in MATLAB.
La verifica sperimentale del metodo è stata realizzata analizzando la risposta acquisita dalla misura di una rete RC serie, quest'ultima composta dalle combinazioni di 5 condensatori e 6 resistori caratterizzati, mediante un multimetro digitale KAIWEETS HT118E, dai seguenti valori:

$$C_{real} = \{95.26\,\text{nF};\ 441.6\,\text{nF};\ 996.8\,\text{nF};\ 4.25\,\mu\text{F};\ 7.1\,\mu\text{F}\}$$

$$R_{real} = \{9.859\,\Omega;\ 98.382\,\Omega;\ 990.09\,\Omega;\ 9.915\,\text{k}\Omega;\ 97.853\,\text{k}\Omega;\ 1.0049\,\text{M}\Omega\}$$

La procedura di misura è stata gestita, analogamente ai casi precedenti, mediante MATLAB, ricorrendo allo **Script 4.9**. In questo caso sono state effettuate un totale di $N = 300$ prove ripetute per ciascun campione di capacità, variando di volta in volta il valore della resistenza serie associata. La riduzione del numero di prove è stata introdotta al fine di contenere il tempo complessivo necessario per la caratterizzazione.

```
1 clear;
2 clc;
```

```
load('Parametri_R_ARDUINO.mat');                %Rout, Rain e Rpu
NMAX = 1023;
port = "COM5";                  % Porta COM dell' Arduino
baud = 115200;                  % Velocita di comunicazione della seriale
s = serialport(port, baud);                     % Apertura seriale
pause(2);
N = 300;
Tau = zeros(N,1);
VR = zeros(N,1);
R_load = zeros(N,1);
C_load = zeros(N,1);
VC = zeros(N,1);
RT  = zeros(N,1);
k = 1;                  % Acquisizione delle grandezze di interesse
while k <= N
    linea = readline(s);
    values = str2double(split(linea,","));
    if length(values)==6                        % Controllo sulla presenza
    di 6 valori
        VR(k)=values(1);
        VC(k)=values(2);
        R_load(k)=values(3);
        C_load(k)=values(4);
        RT(k)=values(5);
        Tau(k)= values(6);
        k=k+1;
    end
end
clear s;                        % Chiusura della connessione seriale
meanR_load = mean(R_load);                      % Valore medio di Rload
stdR_load = std(R_load);                        % Dev.stnd. di Rload
meanC_load = mean(C_load);                      % Valore medio di Cload
```

```
stdC_load = std(C_load);                          % Dev.stnd. di Cload
uA_R_load = stdR_load / sqrt(N);          % Incertezza di tipo A resistiva
uA_C_load = stdC_load / sqrt(N);          % Incertezza di tipo A capacitiva
Cmedia_nF = meanC_load*1e9;
stdC_nF = stdC_load*1e9;
uA_nF = uA_C_load*1e9;
% Stampa dei valori
fprintf('R = %.3f Ohm +/- %.3f Ohm \n',meanR_load,uA_R_load);
fprintf('C = %.3f nF +/- %.3f nF\n',Cmedia_nF,uA_nF);
save('Misurazione.mat', "RT","Tau","VR","VC","R_load","C_load","meanC_load
    ","uA_C_load","meanR_load","uA_R_load");
```

Script 4.9: Codice MATLAB per la gestione della misura di una rete *RC* serie

Una volta acquisiti i dati relativi alle differenti combinazioni di componenti, sono stati elaborati i grafici riportati in **Figura 4.4**, mediante il codice MATLAB descritto nello **Script 4.10**. Non è stato tuttavia possibile caratterizzare tutte le possibili combinazioni dei componenti, in quanto, per le reti costituite dalle coppie con i valori più elevati, il tempo necessario per ottenere una risposta eccedeva il limite imposto da MATLAB.

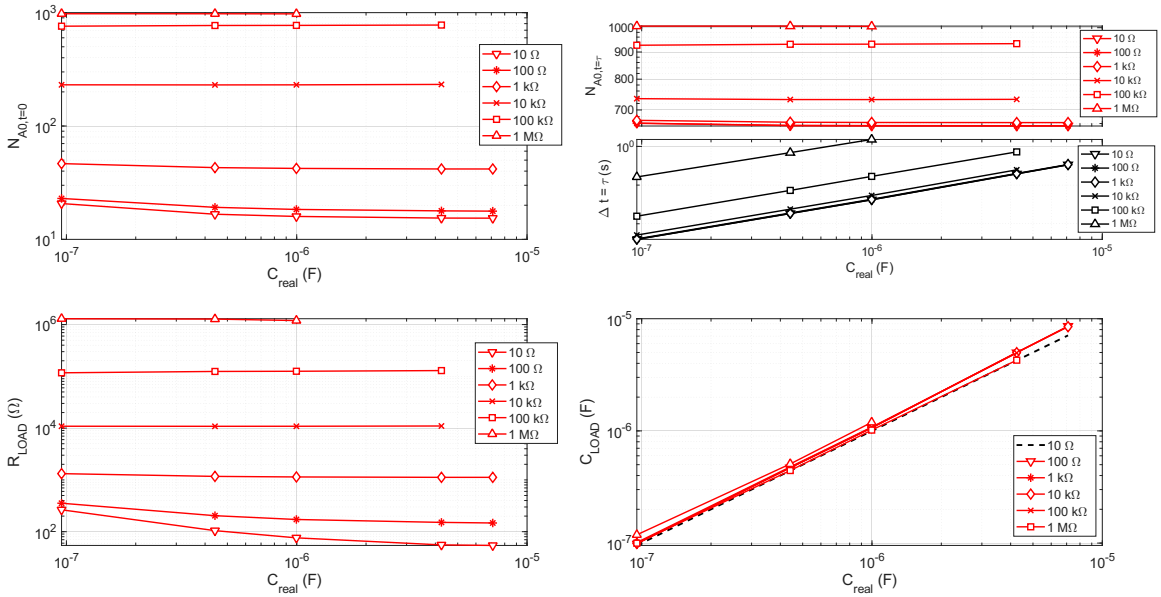


Figura 4.4: Rappresentazione grafica dei risultati riferiti alla rete *RC* serie.

Analisi dei risultati

La caratterizzazione delle reti RC serie, effettuata nell'intervallo di capacità compreso tra 95.26 nF e 7.1 µF e per resistenze comprese tra 9.859 Ω e 1.0049 MΩ, evidenzia gli stessi principali andamenti osservati nel lavoro di riferimento. Il valore di $N_{A0,t=0}$ risulta pressoché indipendente dalla capacità a parità di resistenza, mentre il tempo di transitorio aumenta con C_{real} e con R_{real}, in accordo con la relazione $\tau = RC$. Anche il valore stimato di R_{LOAD} mostra una dipendenza limitata dalla capacità, mentre C_{LOAD} segue nel complesso l'andamento ideale $C_{\text{LOAD}} = C_{\text{real}}$. I risultati ottenuti sono quindi coerenti con il comportamento previsto per una rete RC serie e con le tendenze riportate nell'articolo. Il confronto non comprende tuttavia i limiti della caratterizzazione originale, che si estende da circa 100 pF a 4.7 mF; pertanto, i risultati ottenuti permettono di verificare il corretto comportamento del metodo nel solo intervallo di capacità considerato.

```
clear;
clc;
% Caricamento del file contenente sia i cell array (VR, VC, Tau, Rload e
% Cload) che i vettori riga Creal e Rreal, organizzati in ordine crescente.
% Le righe dei cell array corrispondono a valori crescenti di capacita,
% mentre le colonne a valori crescenti di resistenza R.
load('Dati_reti_RC_serie.mat');
nC=5; nR=6;
meanVR=NaN(nC,nR); meanVC=NaN(nC,nR); meanTau=NaN(nC,nR);
meanR=NaN(nC,nR); meanC=NaN(nC,nR);
stdVR=NaN(nC,nR); stdVC=NaN(nC,nR); stdTau=NaN(nC,nR);
stdR=NaN(nC,nR); stdC=NaN(nC,nR);

for i=1:nC
    meanVR(i,:)=mean(VR{i},1); meanVC(i,:)=mean(VC{i},1);
    meanTau(i,:)=mean(Tau{i},1);
```

```
17      meanR(i,:)=mean(Rload{i},1); meanC(i,:)=mean(Cload{i},1);
18      stdVR(i,:)=std(VR{i},0,1); stdVC(i,:)=std(VC{i},0,1);
19      stdTau(i,:)=std(Tau{i},0,1);
20      stdR(i,:)=std(Rload{i},0,1); stdC(i,:)=std(Cload{i},0,1);
21  end
22  % Grafici
23  marker={'v','*','d','x','s','^'};
24  leg={'10 \Omega','100 \Omega','1 k\Omega','10 k\Omega','100 k\Omega','1 M\
        Omega'};
25
26  figure('Color','w');
27  set(gcf,'Units','normalized','Position',[0.05 0.05 0.9 0.88]);
28  set(groot,'defaultAxesFontSize',15,'defaultTextFontSize',15);
29
30  p1=[0.08 0.56 0.38 0.34]; p2=[0.55 0.56 0.38 0.34];
31  p3=[0.08 0.10 0.38 0.34]; p4=[0.55 0.10 0.38 0.34];
32
33  ax1=axes('Position',p1); hold on; grid on; box on;
34  for k=1:6
35      semilogx(Creal,meanVR(:,k),['-' marker{k}],'Color','r','MarkerFaceColor'
        ,'w','LineWidth',1.5,'MarkerSize',8);
36  end
37  xlabel('C_{real}(F)');ylabel('N_{A0,t=0}');legend(leg);
38
39  ax2=axes('Position',[p2(1) p2(2)+p2(4)/2 p2(3) p2(4)/2-0.02]);
40  hold on; grid on; box on;
41  for k=1:6
42      semilogx(Creal,meanVC(:,k),['-' marker{k}],'Color','r','MarkerFaceColor'
        ,'w','LineWidth',1.5,'MarkerSize',8);
43  end
44  ylabel('N_{A0,t=\tau}'); legend(leg);
45  set(ax2,'XTickLabel',[]);
```

```
46
47 ax3=axes('Position',[p2(1) p2(2) p2(3) p2(4)/2-0.02]); hold on; grid on; box
       on;
48 for k=1:6
49     loglog(Creal,meanTau(:,k),['-' marker{k}],'Color','k','MarkerFaceColor',
       'w','LineWidth',1.5,'MarkerSize',8);
50 end
51 xlabel('C_{real}(F)');ylabel('\Delta t=\tau(s)');legend(leg);
52
53 ax4=axes('Position',p3); hold on; grid on; box on;
54 for k=1:6
55     loglog(Creal,meanR(:,k),['-' marker{k}],'Color','r','MarkerFaceColor','w
       ','LineWidth',1.5,'MarkerSize',8);
56 end
57 xlabel('C_{real}(F)');ylabel('R_{LOAD}(\Omega)');legend(leg);
58
59 ax5=axes('Position',p4); hold on; grid on; box on;
60 loglog(Creal,Creal,'k--','LineWidth',2);
61 for k=1:6
62     loglog(Creal,meanC(:,k),['-' marker{k}],'Color','r','MarkerFaceColor','w
       ','LineWidth',1.5,'MarkerSize',8);
63 end
64 xlabel('C_{real}(F)'); ylabel('C_{LOAD}(F)'); legend(leg);
```

Script 4.10: Codice MATLAB per l'elaborazione dei dati di una rete *RC* serie

4.2.4 Rete *RC* parallelo

La procedura di caratterizzazione di una rete *RC* parallelo, descritta precedentemente nella **Sezione 4.1.4**, è stata implementata tramite Arduino Uno mediante lo **Script 4.11**.

```
#include <SPI.h>
#define PA0 A0
#define PA1 A3
#define NMAX 1023.0
#define RAIN 2.4958E6
#define RPU 3.6E4
#define ROUT 541.9869
#define RxTx 115200
void setup() {
  Serial.begin(RxTx);
  pinMode(PA0, OUTPUT);
  digitalWrite(PA0, LOW);
  pinMode(PA1, OUTPUT);
  digitalWrite(PA1, LOW);
  delay(1000);
}

void loop() {
float Cmax=1000E-9;   // ordine di grandezza della capacita da analizzare
float Rpar=(RAIN*RPU)/(RAIN+RPU);
float Tau_max=Rpar*Cmax;
pinMode(PA0,OUTPUT);
digitalWrite(PA0,LOW);
pinMode(PA0,INPUT_PULLUP);
delay((unsigned long)(5*Tau_max*1000));
float VF=analogRead(PA0);
float Vtarget=0.6322*VF;
pinMode(PA0,OUTPUT);
digitalWrite(PA0,LOW);
delay(50);
pinMode(PA0,INPUT_PULLUP);
```

```
32 unsigned long TS=micros();
33 float VC;
34 do{VC=analogRead(PA0);
35 }while(VC<Vtarget);
36 unsigned long TE=micros();
37 float T=(TE-TS)*1E-6;
38 float K=(RPU*VF)/(NMAX-VF);
39 float R_LOAD=((RAIN+ROUT)*K-RAIN*ROUT)/(RAIN-K);
40 float RT=RAIN*RPU/(RAIN+RPU);
41 RT=RT*(ROUT+R_LOAD)/(RT+(ROUT+R_LOAD));
42 float C_LOAD=T/RT;
43 Serial.print(VF); Serial.print(",");
44 Serial.print(VC); Serial.print(",");
45 Serial.print(R_LOAD); Serial.print(",");
46 Serial.print(C_LOAD, 12); Serial.print(",");
47 Serial.print(RT); Serial.print(",");
48 Serial.println(T, 12);
49 delay(5000 * T);
```

Script 4.11: Codice Arduino per l'RC parallelo meter

L'implementazione della procedura segue direttamente i passaggi descritti per il principio di misura. Inizialmente viene assunto un valore $C_{\max}$ dello stesso ordine di grandezza della capacità da analizzare, utilizzato per stimare la massima costante di tempo del circuito. Il programma configura quindi P_{A0} come ingresso con pull-up e attende un intervallo pari a circa $5\tau_{\max}$, in modo da garantire che il transitorio sia terminato e che la tensione abbia raggiunto il valore di regime V_F. Quest'ultimo viene acquisito tramite l'ADC e utilizzato per il calcolo di R_{LOAD} e della resistenza equivalente R_T.

Successivamente il circuito viene nuovamente scaricato e la misura del transitorio viene avviata registrando l'istante iniziale mediante il timer del microcontrollore. Il programma acquisisce ripetutamente la tensione tramite ADC fino al raggiungimento

della soglia pari al 63.22% di V_F, corrispondente alla tensione raggiunta dal condensatore dopo un tempo pari a una costante di tempo τ. La differenza tra l'istante di raggiungimento della soglia e quello iniziale fornisce quindi direttamente $T = \tau$, dal quale viene infine ricavato il valore di C_{LOAD} attraverso la relazione $C_{\mathrm{LOAD}} = T/R_T$. I risultati della misura vengono infine inviati tramite la porta seriale per la successiva elaborazione in MATLAB. La verifica sperimentale del metodo è stata realizzata analizzando la risposta acquisita dalla misura di una rete RC parallelo, quest'ultima composta dalle combinazioni dei medesimi componenti utilizzati nel caso della misura di una rete RC serie:

$$C_{real} = \{95.26\,\mathrm{nF};\ 441.6\,\mathrm{nF};\ 996.8\,\mathrm{nF};\ 4.25\,\mu\mathrm{F};\ 7.1\,\mu\mathrm{F}\}$$

$$R_{real} = \{9.859\,\Omega;\ 98.382\,\Omega;\ 990.09\,\Omega;\ 9.915\,\mathrm{k}\Omega;\ 97.853\,\mathrm{k}\Omega;\ 1.0049\,\mathrm{M}\Omega\}$$

La procedura di misura è stata gestita mediante MATLAB, ricorrendo allo **Script 4.12**. In questo caso sono state effettuate un totale di $N = 100$ prove ripetute per ciascun campione di capacità, variando di volta in volta il valore della resistenza in parallelo ad esso connessa. La riduzione ulteriore del numero di prove rispetto al caso serie è stata introdotta al fine di permettere la caratterizzazione completa dei componenti esaminati.

```
clear;
clc;
load('Parametri_R_ARDUINO.mat');        % Rout, Rain e Rpu
NMAX = 1023;
port = "COM5";          % Porta COM dell' Arduino
baud = 115200;          % Velocita di comunicazione della seriale
s = serialport(port, baud);     % Apertura seriale
pause(2);
N = 100;
Tau = zeros(N,1);
VR = zeros(N,1);
R_load = zeros(N,1);
```

```
C_load = zeros(N,1);
VC = zeros(N,1);
RT  = zeros(N,1);

k = 1;                  % Acquisizione dati
while k <= N
    linea = readline(s);
    values = str2double(split(linea,","));
    if length(values)==6        % Controllo sulla presenza di 6 valori
        VF(k) = values(1);
        VC(k)=values(2);
        R_load(k)=values(3);
        C_load(k)=values(4);
        RT(k)=values(5);
        Tau(k)= values(6);
        k=k+1;
    end
end
clear s;                % Chiusura della connessione seriale
meanR_load = mean(R_load);      % Valore medio di Rload
stdR_load = std(R_load);        % Dev.stnd. di Rload
meanC_load = mean(C_load);      % Valore medio di Cload
stdC_load = std(C_load);        % Dev.stnd. di Cload
uA_R_load = stdR_load / sqrt(N);        % Incertezza di tipo A resistiva
uA_C_load = stdC_load / sqrt(N);        % Incertezza di tipo A capacitiva
% Stampa dei valori
fprintf('R = %.3f Ohm +/- %.3f Ohm \n',meanR_load,uA_R_load);
fprintf('C = %.3f nF +/- %.3f nF\n',Cmedia_nF,uA_nF);
save('Misurazione.mat', "RT","Tau","VF","VC","R_load","C_load","meanC_load
    ","uA_C_load","meanR_load","uA_R_load");
```

Script 4.12: Codice MATLAB per la gestione della misura di una rete RC parallelo

Una volta acquisiti i dati relativi alle differenti combinazioni di componenti, sono stati elaborati i grafici riportati in **Figura 4.5**, mediante il codice MATLAB descritto nello **Script 4.13**.

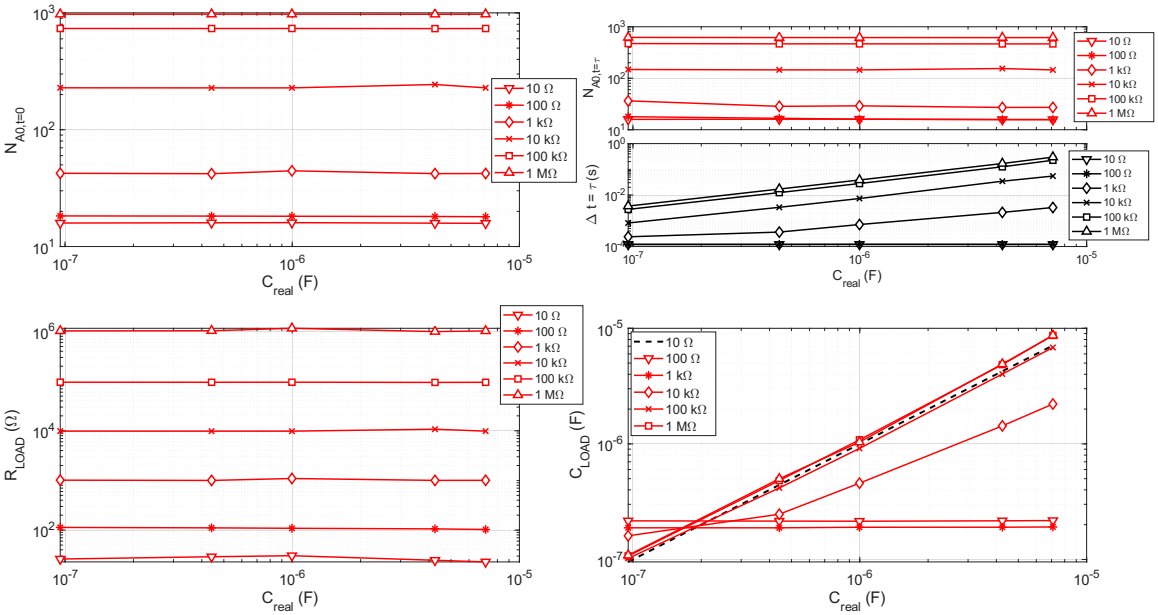


Figura 4.5: Rappresentazione grafica dei dati sulla rete RC parallelo.

Analisi dei risultati

La caratterizzazione delle reti RC parallelo, effettuata nell'intervallo di capacità compreso tra 95.26 nF e 7.1 µF e per resistenze comprese tra 9.859 Ω e 1.0049 MΩ, evidenzia un comportamento coerente con quello atteso e con le tendenze riportate nel lavoro di riferimento. I valori di $N_{A0,t=0}$ e $N_{A0,t=\tau}$ risultano sostanzialmente indipendenti da C_{real} a parità di resistenza, mentre il tempo di transitorio cresce all'aumentare di R_{real} e C_{real}, in accordo con la relazione $\tau = R_T C_{\text{LOAD}}$. La stima di R_{LOAD} risulta inoltre relativamente stabile al variare della capacità. Per C_{LOAD}, invece, si osserva una maggiore dipendenza dal valore di R_{LOAD}: per le resistenze più basse i valori misurati si discostano maggiormente dall'andamento ideale, in particolare le resistenze da 100 Ω e da 1 kΩ, mentre per resistenze più elevate la corrispondenza con $C_{\text{LOAD}} = C_{\text{real}}$ migliora sensibilmente. Questo comportamento è in accordo con quanto osservato nell'articolo, dove le prestazioni della misura di reti RC parallelo dipendono dalla combinazione dei valori di R e C. Nel complesso, i risultati confermano quindi il

corretto andamento del metodo nel range analizzato, evidenziando al contempo una maggiore difficoltà nella determinazione della capacità per i valori più bassi di resistenza.

```
clear;
clc;
% Caricamento del file contenente sia i cell array (VF, VC, Tau, Rload e
% Cload) che i vettori riga Creal e Rreal, organizzati in ordine crescente.
% Le righe dei cell array corrispondono a valori crescenti di capacita,
% mentre le colonne a valori crescenti di resistenza R.
load('Dati_reti_RC_parallelo.mat');
nC=5; nR=6;
meanVF=NaN(nC,nR); meanVC=NaN(nC,nR); meanTau=NaN(nC,nR);
meanR=NaN(nC,nR); meanC=NaN(nC,nR);
stdVF=NaN(nC,nR); stdVC=NaN(nC,nR); stdTau=NaN(nC,nR);
stdR=NaN(nC,nR); stdC=NaN(nC,nR);

for i=1:nC
    meanVF(i,:)=mean(VF{i},1); meanVC(i,:)=mean(VC{i},1);
    meanTau(i,:)=mean(Tau{i},1);
    meanR(i,:)=mean(Rload{i},1); meanC(i,:)=mean(Cload{i},1);
    stdVF(i,:)=std(VF{i},0,1); stdVC(i,:)=std(VC{i},0,1);
    stdTau(i,:)=std(Tau{i},0,1);
    stdR(i,:)=std(Rload{i},0,1); stdC(i,:)=std(Cload{i},0,1);
end
% Grafici
marker={'v','*','d','x','s','^'};
leg={'10 \Omega','100 \Omega','1 k\Omega','10 k\Omega','100 k\Omega','1 M\
    Omega'};

figure('Color','w');
```

```
27 set(gcf,'Units','normalized','Position',[0.05 0.05 0.9 0.88]);
28 set(groot,'defaultAxesFontSize',15,'defaultTextFontSize',15);
29
30 p1=[0.08 0.56 0.38 0.34]; p2=[0.55 0.56 0.38 0.34];
31 p3=[0.08 0.10 0.38 0.34]; p4=[0.55 0.10 0.38 0.34];
32
33 ax1=axes('Position',p1); hold on; grid on; box on;
34 for k=1:6
35     semilogx(Creal,meanVF(:,k),['-' marker{k}],'Color','r','MarkerFaceColor'
       ,'w','LineWidth',1.5,'MarkerSize',8);
36 end
37 xlabel('C_{real}(F)');ylabel('N_{A0,t=0}');legend(leg);
38
39 ax2=axes('Position',[p2(1) p2(2)+p2(4)/2 p2(3) p2(4)/2-0.02]);
40 hold on; grid on; box on;
41 for k=1:6
42     semilogx(Creal,meanVC(:,k),['-' marker{k}],'Color','r','MarkerFaceColor'
       ,'w','LineWidth',1.5,'MarkerSize',8);
43 end
44 ylabel('N_{A0,t=\tau}'); legend(leg);
45 set(ax2,'XTickLabel',[]);
46
47 ax3=axes('Position',[p2(1) p2(2) p2(3) p2(4)/2-0.02]); hold on; grid on; box
        on;
48 for k=1:6
49     loglog(Creal,meanTau(:,k),['-' marker{k}],'Color','k','MarkerFaceColor',
       'w','LineWidth',1.5,'MarkerSize',8);
50 end
51 xlabel('C_{real}(F)');ylabel('\Delta t=\tau(s)');legend(leg);
52
53 ax4=axes('Position',p3); hold on; grid on; box on;
54 for k=1:6
```

```
55      loglog(Creal,meanR(:,k),['-' marker{k}],'Color','r','MarkerFaceColor','w
        ','LineWidth',1.5,'MarkerSize',8);
56  end
57  xlabel('C_{real}(F)');ylabel('R_{LOAD}(\Omega)');legend(leg);
58
59  ax5=axes('Position',p4); hold on; grid on; box on;
60  loglog(Creal,Creal,'k--','LineWidth',2);
61  for k=1:6
62      loglog(Creal,meanC(:,k),['-' marker{k}],'Color','r','MarkerFaceColor','w
        ','LineWidth',1.5,'MarkerSize',8);
63  end
64  xlabel('C_{real}(F)'); ylabel('C_{LOAD}(F)'); legend(leg);
```

Script 4.13: Codice MATLAB per l'elaborazione dei dati di una rete RC parallelo

4.3 Misura di una resistenza isolata mediante metodo a ponte con Arduino Uno

Il sistema sviluppato realizza un ohmetro basato sul principio del ponte resistivo, costituito dalle resistenze note R_1, R_2, R_3 e dalla resistenza incognita R_x. Le resistenze R_1 e R_2 sono state scelte con lo stesso valore nominale, pari a $5\,\mathrm{k}\Omega$, in modo da ottenere, in condizioni ideali, una tensione al nodo centrale pari alla metà della tensione di alimentazione.

Per la resistenza R_3, invece, non è stato scelto un unico valore nominale. Il valore di R_3 influenza infatti l'intervallo di valori della resistenza incognita R_x nel quale l'ohmetro presenta la massima sensibilità, quindi la sua portata. Per questo motivo, R_3 è stata resa variabile all'interno di un insieme di possibili valori nominali, nello specifico:

$$R_3 = \{10\,\Omega;\ 100\,\Omega;\ 1\,\mathrm{k}\Omega;\ 10\,\mathrm{k}\Omega;\ 100\,\mathrm{k}\Omega;\ 1\,\mathrm{M}\Omega\}.$$

I componenti utilizzati nel ponte di misura sono stati in seguito caratterizzati mediante un multimetro digitale KAIWEETS HT118B, dal quale sono stati ottenuti:

$$R_1 = R_2 = 5.1\,\text{k}\Omega, \qquad R_3 = \{10.04\,\Omega;\ 99.84\,\Omega;\ 994.2\,\Omega;\ 9.945\,\text{k}\Omega;\ 100.15\,\text{k}\Omega;\ 1.0139\,\text{M}\Omega\}.$$

Il ponte è alimentato a 5 V e presenta due tensioni ai nodi centrali, corrispondenti ai punti di collegamento tra $R_1 - R_2$ e $R_3 - R_x$. Queste due tensioni vengono applicate agli ingressi degli amplificatori operazionali contenuti nell'LM358, configurati come buffer a guadagno unitario, in modo da non caricare il ponte e da fornire all'Arduino due segnali a bassa impedenza. Le uscite dei buffer sono collegate rispettivamente agli ingressi analogici $A0$ e $A1$ dell'Arduino Uno, che provvede alla loro conversione tramite il convertitore analogico-digitale. Lo schema elettrico che rappresenta il circuito appena descritto è stato riportato in **Figura 4.6**.

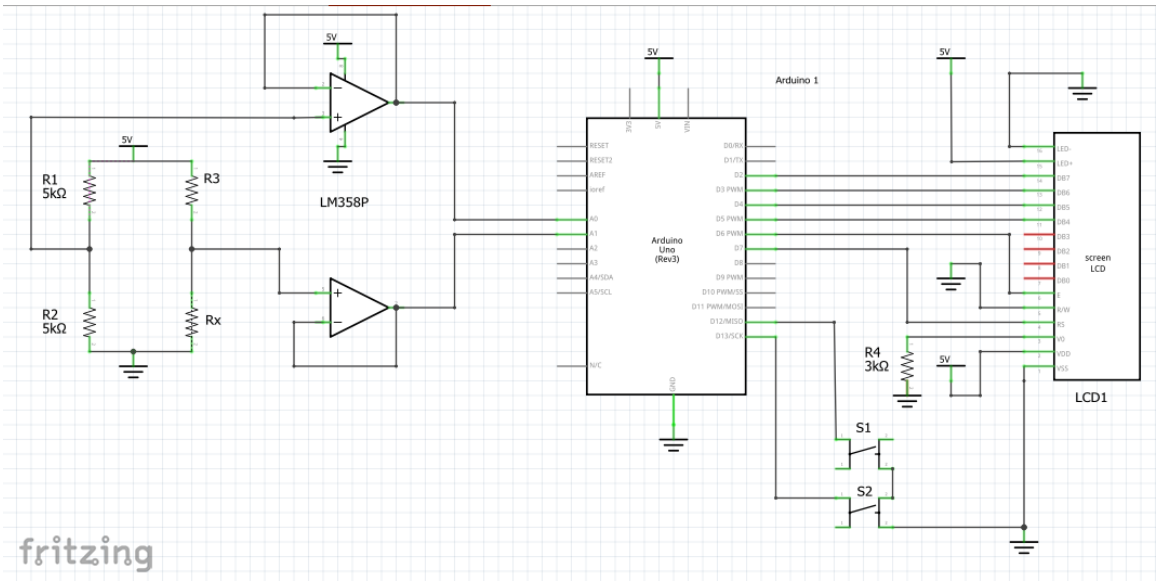


Figura 4.6: Schema elettrico dell'ohmetro con Arduino Uno.

Il codice Arduino che implementa l'ohmetro, basato sul principio della misura a ponte, è riportato nello **Script 4.14**. Il programma acquisisce ripetutamente i due valori digitali N_0 e N_1, proporzionali alle rispettive tensioni V_0 e V_1, calcolandone il rapporto

$$k = \frac{N_1}{N_0} = \frac{(V_1/V_{ref})(2^n - 1)}{(V_0/V_{ref})(2^n - 1)} = \frac{V_1}{V_0}$$

dove $V_{ref} = 5\,\text{V}$ è la tensione di alimentazione fornita da Arduino Uno.

Analizzando il circuito del ponte è possibile ottenere le relazioni $V_1 = V_{ref}\frac{R_x}{R_3+R_x}$ e $V_0 = V_{ref}\frac{R_2}{R_1+R_2}$. Sostituendo tali relazioni nella definizione del parametro k e assumendo $R_1 = R_2$, si ricava il valore della resistenza incognita mediante la relazione $R_x = R_3\dfrac{k}{2-k}$, utilizzata nel codice Arduino sviluppato. Per ridurre l'effetto delle fluttuazioni della lettura di N_1 e N_0, per ogni misura vengono effettuate 10 acquisizioni e ne viene calcolata la media; successivamente vengono eseguite 10 misure di R_x, dalle quali viene determinato il valore medio finale $\overline{R_x}$, la deviazione standard campionaria σ_{R_x} e l'incertezza di tipo A ad essa associata $u_A(R_x)$. Una grandezza fondamentale che caratterizza le misure basate sul metodo a ponte è la sensibilità, definita come la variazione della tensione differenziale $V = V_0 - V_1$ al variare della resistenza incognita R_x:

$$S = \left|\frac{\partial V}{\partial R_x}\right| = \frac{R_3\, V_{DD}}{(R_3 + R_x)^2}.$$

Imponendo $\dfrac{\partial S}{\partial R_3} = 0$ si ottiene la condizione di sensibilità massima

$$\frac{(R_3 + R_x)(R_x - R_3)}{(R_3 + R_x)^4} = 0 \iff R_3 = R_x,$$

in corrispondenza della quale la sensibilità vale

$$S\big|_{R_3=R_x} = \frac{V_{DD}}{4R_3}.$$

Questo risultato giustifica la scelta di suddividere R_3 in sei portate: l'incertezza sulla misura di R_x, legata a quella sulla tensione differenziale tramite $\Delta R_x = \Delta V/S$, è minima quando $R_3 \approx R_x$, mentre cresce all'aumentare dello scostamento $|R_3 - R_x|$. Da qui deriva anche il controllo software implementato nel codice, che segnala all'utente di cambiare portata quando il rapporto tra le tensioni misurate indica che R_3 e R_x sono troppo distanti tra loro. Se R_3 è troppo piccola, viene richiesto di aumentare la portata, mentre se è troppo grande viene richiesto di diminuirla. Poiché nel circuito non è presente un multiplexer o un altro sistema di commutazione automatica, il valore di R_3 deve essere sostituito manualmente in base alla portata desiderata. La sostituzione fisica di R_3 deve essere accompagnata dalla corrispondente selezione a livello software, in modo che il programma utilizzi il corretto valore di R_3 nei calcoli.

La portata può essere modificata tramite i due pulsanti collegati ai pin digitali 12 e 13, che permettono rispettivamente di aumentarla o diminuirla; è inoltre possibile selezionarla tramite la comunicazione seriale. In questo modo il valore memorizzato nel programma rimane coerente con la resistenza R_3 effettivamente inserita nel circuito. Una volta ottenuta una misura valida, il valore di R_x viene visualizzato sia sul display LCD sia sulla porta seriale, utilizzando automaticamente i multipli dell'ohm più appropriati, come kΩ o MΩ.

Valutazione dell'incertezza di tipo B L'incertezza di tipo B su R_x è stata valutata analizzando tre contributi indipendenti, propagati attraverso la relazione

$$R_x = R_3 \frac{k}{2-k}, \qquad k = \frac{N_1}{N_0},$$

mediante la legge di propagazione delle incertezze $u_c(y) = \sqrt{\sum_i \left(\frac{\partial f}{\partial x_i}\right)^2 u^2(x_i)}$.
Contributo del resistore R_3. L'incertezza su R_3 è stata valutata a partire dall'accuratezza dichiarata dal datasheet del multimetro KAIWEETS HT118B, distinta per fondo scala:

$$\Delta R = \pm(1.0\%\,\text{lettura} + 15\,\text{digit}) \quad \text{per le portate } 200\,\Omega,\ 2\,\text{k}\Omega,\ 20\,\text{k}\Omega,\ 200\,\text{k}\Omega,\ 2\,\text{M}\Omega,$$

$$\Delta R = \pm(3.0\%\,\text{lettura} + 25\,\text{digit}) \quad \text{per le portate } 20\,\text{M}\Omega,\ 100\,\text{M}\Omega,$$

dove il valore del digit è calcolato come portata/20000 counts (risoluzione dello strumento). Assumendo per l'errore strumentale una distribuzione di probabilità uniforme, l'incertezza tipo si ottiene dividendo il semiintervallo per $\sqrt{3}$:

$$u(R_{3,i}) = \frac{0.01 \cdot R_{3,i}^{mis} + 15 \cdot (\text{portata}_i/20000)}{\sqrt{3}}.$$

Applicando questa formula alle sei portate nominali impiegate (10 Ω, 100 Ω, 1 kΩ, 10 kΩ, 100 kΩ, 1 MΩ, con valori misurati rispettivamente 10.04 Ω, 99.84 Ω, 0.994 kΩ, 9.944 kΩ, 100.15 kΩ e 1.014 MΩ) sono state ottenute le incertezze riportate in **Tabella 4.3**.

Tabella 4.3: Incertezza tipo B su R_3, per ciascuna portata utilizzata

R_3 nominale	R_3 misurata	$u(R_3)$
$10\,\Omega$	$10.0\,\Omega$	$\pm 0.2\,\Omega$
$100\,\Omega$	$99.8\,\Omega$	$\pm 0.7\,\Omega$
$1\,\mathrm{k}\Omega$	$994\,\Omega$	$\pm 7\,\Omega$
$10\,\mathrm{k}\Omega$	$9.94\,\mathrm{k}\Omega$	$\pm 0.07\,\mathrm{k}\Omega$
$100\,\mathrm{k}\Omega$	$100.2\,\mathrm{k}\Omega$	$\pm 0.7\,\mathrm{k}\Omega$
$1\,\mathrm{M}\Omega$	$1014\,\mathrm{k}\Omega$	$\pm 7\,\mathrm{k}\Omega$

Da questi valori si osserva che, per le portate superiori a $100\,\Omega$, il rapporto $u(R_3)/R_3$ risulta pressoché costante, con

$$\frac{u(R_3)}{R_3} \approx 0.0066,$$

mentre per la portata più piccola ($10\,\Omega$) tale rapporto risulta leggermente superiore a causa della maggiore incidenza del termine in digit rispetto al valore letto. Questa approssimazione ha comunque permesso di semplificare la propagazione dell'incertezza per le portate più utilizzate. Poiché $R_x = R_3\,\dfrac{k}{2-k}$, si ha $\dfrac{\partial R_x}{\partial R_3} = \dfrac{k}{2-k} = \dfrac{R_x}{R_3}$, per cui il contributo di R_3 sull'incertezza di R_x diventa

$$u_{R_3}(R_x) = \left|\frac{\partial R_x}{\partial R_3}\right| u(R_3) = R_x \cdot \frac{u(R_3)}{R_3} \approx R_x \cdot 0.0066,$$

ovvero un'incertezza relativa pressoché costante, pari a circa lo 0.66% del valore misurato di R_x, per le portate superiori a $100\,\Omega$.

Contributo dei resistori R_1 ed R_2. Poiché nel circuito si assume $R_1 = R_2$ (condizione per cui $R_x = R_3\,\frac{k}{2-k}$ e $V_0 = V_{DD}/2$), l'incertezza su questi due resistori è stata valutata considerando il caso peggiore in cui essi si discostano al massimo l'uno dall'altro, anziché sommando in quadratura due contributi indipendenti. Con $R_1 = R_2 = 5100 \pm 38\,\Omega$ (incertezza relativa 0.75%, ottenuta come per R_3 dall'accuratezza del multimetro), il caso peggiore corrisponde a

$$R_1 = 5100 + 38 = 5138\,\Omega, \qquad R_2 = 5100 - 38 = 5062\,\Omega,$$

da cui

$$V_0 = \frac{R_2}{R_1 + R_2}\, V_{DD} = 0.4963\, V_{DD} \quad \text{anziché} \quad V_0 = 0.5\, V_{DD}.$$

Lo scostamento risultante è $\Delta V_0 = (0.5 - 0.4963)\, V_{DD} = 0.0037\, V_{DD}$, ovvero $\Delta V_0\% = 0.37\%$. Assumendo anche qui una distribuzione uniforme, si ottiene

$$u(V_0) = \frac{\Delta V_0}{\sqrt{3}} \approx 0.00214\, V_{DD}.$$

Questo scostamento si ripercuote sul parametro $k = V_1/V_0$ tramite

$$\Delta k = -\frac{V_1}{V_0^2}\,\Delta V_0 = -k\,\frac{\Delta V_0}{V_0} \;\Rightarrow\; \left|\frac{\Delta k}{k}\right| = \left|\frac{\Delta V_0}{V_0}\right|,$$

e quindi, sotto la stessa ipotesi di distribuzione uniforme,

$$u_{R_{1,2}}(k) = k\,\frac{u(V_0)}{V_0}.$$

Tale incertezza su k è stata infine propagata su R_x tramite $\dfrac{\partial R_x}{\partial k} = 2\dfrac{R_3}{(2-k)^2}$, ottenendo il termine $u_{R_{1,2}}$ presente nella somma in quadratura finale.

Contributo della quantizzazione ADC. La conversione a 10 bit di Arduino Uno (riferimento $V_{ref} = 5\,\mathrm{V}$) introduce un LSB pari a $4.88\,\mathrm{mV}$. Assumendo una distribuzione uniforme dell'errore di quantizzazione nell'intervallo $[-\mathrm{LSB}/2\,;\,\mathrm{LSB}/2]$, si ha $\sigma_q = \mathrm{LSB}/\sqrt{12} \approx 1.41\,\mathrm{mV}$, ridotta a $\sigma_q/\sqrt{10} \approx 0.45\,\mathrm{mV}$ mediando su 10 acquisizioni ripetute (errore relativo $\approx 0.009\%$). Propagando questo contributo su $k = N_1/N_0$ e quindi su R_x tramite $\dfrac{\partial R_x}{\partial k} = 2\dfrac{R_3}{(2-k)^2}$, si ottiene il termine u_{ADC}.

Contributo dell'amplificatore operazionale. Le non idealità dell'LM358P (tensione di offset tipica $V_{OS} \approx 2\,\mathrm{mV}$ e corrente di bias tipica $I_B \approx 2\,\mathrm{nA}$) introducono un errore di tensione su entrambi i rami del ponte, proporzionale rispettivamente alle resistenze equivalenti viste dagli ingressi differenziali, $R_{eq,0} = R_1 \| R_2$ e $R_{eq,1} = R_x \| R_3$:

$$u(N_0) = \sqrt{u^2(V_{OS}) + \big(u(I_B)\, R_{eq,0}\big)^2}, \qquad u(N_1) = \sqrt{u^2(V_{OS}) + \big(u(I_B)\, R_{eq,1}\big)^2},$$

con $u(V_{OS}) = V_{OS}/\sqrt{3}$ e $u(I_B) = I_B/\sqrt{3}$ (ipotesi di distribuzione uniforme). Questo contributo, propagato analogamente al precedente, risulta essere il più piccolo tra i tre.

Incertezza combinata. I quattro termini (tipo A + tre di tipo B) sono stati sommati in quadratura:

$$u_c(R_x) = \sqrt{u_A^2 + u_{R_3}^2 + u_{R_{1,2}}^2 + u_{ADC}^2 + u_{OPAMP}^2}.$$

A titolo di esempio, per una misura con R_3 nominale 1 kΩ è stato ottenuto $R_x = 2.057 \pm 0.015\,\text{k}\Omega$, corrispondente a un'incertezza relativa dello 0.714%; scomponendo i contributi (in unità arbitrarie coerenti) si è ottenuto $u_A = 0.559$, $u_{R_3} = 13.58$, $u_{R_{1,2}} = 5.42$, $u_{ADC} = 1.41$ e $u_{OPAMP} = 0.018$, a conferma che il termine dominante è l'incertezza sul resistore di riferimento R_3, seguito da quella su R_1, R_2, mentre i contributi di quantizzazione e dell'operazionale risultano trascurabili.

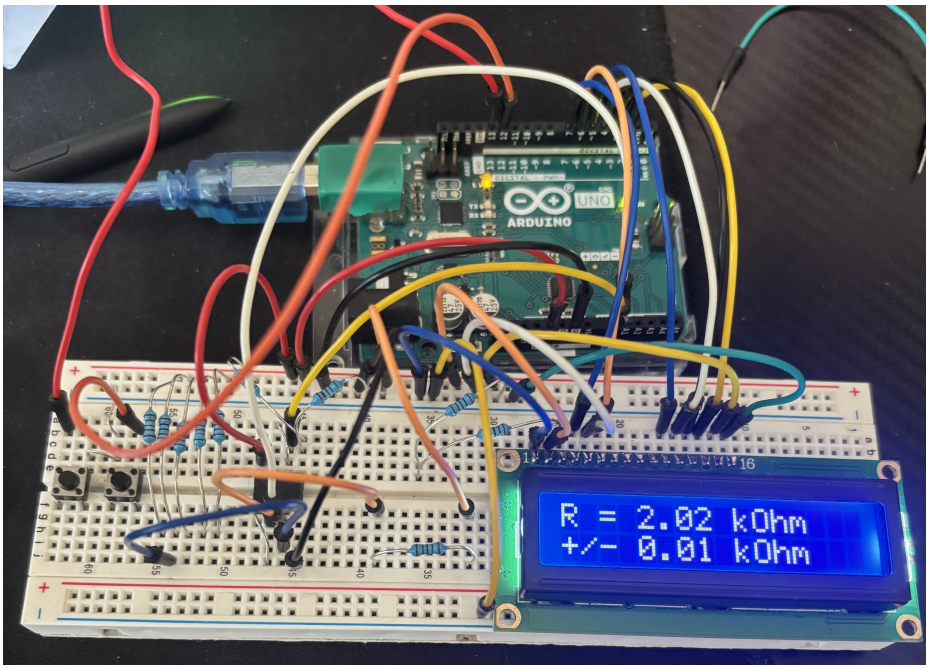


Figura 4.7: Implementazione fisica dell'ohmetro su una breadboard.

Verifica sperimentale del metodo

Per confrontare le prestazioni del metodo a ponte adottato rispetto a quello basato sulla sola scheda Arduino, sono state effettuate delle misurazioni su 25 resistori già caratterizzati mediante il metodo precedente. I risultati della campagna di misure effettuata sono stati riportati nella **Tabella 4.4**, assieme ai valori nominali dei campioni analizzati e ai valori caratterizzati tramite l'LCX200.

Tabella 4.4: Risultati della campagna di misure sui resistori con metodo a ponte

R nominale	R reale	R_{ponte}	u_c	R nominale	R reale	R_{ponte}	u_c
1 Ω	1.02 Ω	3.68 Ω	0.09 Ω	2.2 kΩ	2.154 kΩ	2.15 kΩ	0.02 kΩ
2.2 Ω	2.17 Ω	4.11 Ω	0.03 Ω	4.7 kΩ	4.585 kΩ	4.59 kΩ	0.04 kΩ
3.3 Ω	3.33 Ω	5.79 Ω	0.04 Ω	5.6 kΩ	5.522 kΩ	5.54 kΩ	0.04 kΩ
4.7 Ω	4.66 Ω	6.05 Ω	0.05 Ω	10 kΩ	9.915 kΩ	9.94 kΩ	0.07 kΩ
10 Ω	9.86 Ω	10.37 Ω	0.08 Ω	15 kΩ	14.834 kΩ	14.8 kΩ	0.1 kΩ
22 Ω	21.78 Ω	22.5 Ω	0.2 Ω	22 kΩ	21.618 kΩ	21.6 kΩ	0.2 kΩ
47 Ω	46.57 Ω	48.6 Ω	0.4 Ω	47 kΩ	46.045 kΩ	46.4 kΩ	0.4 kΩ
68 Ω	67.47 Ω	68.4 Ω	0.5 Ω	100 kΩ	97.853 kΩ	97.8 kΩ	0.7 kΩ
100 Ω	98.38 Ω	99.6 Ω	0.7 Ω	1 MΩ	1.005 MΩ	1.005 MΩ	0.007 MΩ
150 Ω	148.34 Ω	149 Ω	1 Ω				
220 Ω	215.8 Ω	216 Ω	2 Ω				
330 Ω	323.25 Ω	332 Ω	3 Ω				
470 Ω	464.5 Ω	465 Ω	4 Ω				
1 kΩ	990.09 Ω	989 Ω	7 Ω				
1.5 kΩ	1.483 kΩ	1.47 kΩ	0.01 kΩ				

```
1  #include <string.h>
2  #include <math.h>
3  #include <LiquidCrystal.h>
4
5  #define RxTx 115200
6  #define R1 5100.0
7  #define R2 5100.0
8  #define VOS 0.002
9  #define IB 20E-9
10 #define R12_ERR 0.00214
11 #define R3_ERR 0.0066
12 #define BTN_UP 12
13 #define BTN_DOWN 13
14 #define N_MEASURE 10
15 #define N_AVG 10
16
17 LiquidCrystal lcd(7,6,5,4,3,2);
18 float Rx[N_MEASURE];
```

```
19 const float R3[]={10.04,99.84,994.2,9945.0,100.15E3,1.0139E6};
20 const float R3_portata[]={1E1,1E2,1E3,1E4,1E5,1E6};
21 int portata=0;
22 // Stampa LCD
23 void stampaLCD(float valore,float incertezza){
24   float fattore; char unita[3];
25   if(valore>=1E6){fattore=1E6;strcpy(unita,"M");}
26   else if(valore>=1E3){fattore=1E3;strcpy(unita,"k");}
27   else{fattore=1;strcpy(unita,"");}
28   valore/=fattore; incertezza/=fattore;
29   lcd.clear(); lcd.setCursor(0,0);
30   lcd.print("R = "); lcd.print(valore,2); lcd.print(" ");
31   lcd.print(unita); lcd.print("Ohm");
32   lcd.setCursor(0,1); lcd.print("+/- "); lcd.print(incertezza,2);
33   lcd.print(" "); lcd.print(unita); lcd.print("Ohm");
34 }
35 // Stampa seriale
36 void stampaRisultato(float valore,float incertezza){
37   char unita[3]; float fattore;
38   if(incertezza<=0){Serial.println("Errore");return;}
39   if(valore>=1E6){fattore=1E6;strcpy(unita,"M");}
40   else if(valore>=1E3){fattore=1E3;strcpy(unita,"k");}
41   else{fattore=1;strcpy(unita,"");}
42   valore/=fattore; incertezza/=fattore;
43   int ordine=floor(log10(incertezza)); float scala=pow(10,ordine-1);
44   incertezza=round(incertezza/scala)*scala;
45   valore=round(valore/scala)*scala;
46   int decimali=0; if(scala<1) decimali=-ordine+1;
47   Serial.print(" ( "); Serial.print(valore,decimali); Serial.print(" +/- ");
48   Serial.print(incertezza,decimali);
49   Serial.print(" ) "); Serial.print(unita); Serial.print("Ohm");
50 }
```

```
51 void setup(){
52   pinMode(BTN_UP,INPUT_PULLUP); pinMode(BTN_DOWN,INPUT_PULLUP);
53   Serial.begin(RxTx); lcd.begin(16,2);
54   lcd.clear(); lcd.print("Ohmetro Pronto");
55   delay(1000); lcd.clear();
56 }
57
58 void loop(){
59   if(Serial.available()){
60     char c=Serial.read(); if(c>='1'&&c<='6') portata=c-'1';
61     while(Serial.available()) Serial.read();
62   }
63   static bool statoUp=HIGH; static bool statoDown=HIGH;
64   bool letturaUp=digitalRead(BTN_UP);
65   bool letturaDown=digitalRead(BTN_DOWN);
66   // Incremento portata con pulsante
67   if(statoUp==HIGH&&letturaUp==LOW){
68     portata++; if(portata>5) portata=5;
69     Serial.print("Portata selezionata: "); Serial.println(portata+1);
70     lcd.clear(); lcd.setCursor(0,0);
71     lcd.print("Portata "); lcd.print(portata+1);
72     lcd.setCursor(0,1); lcd.print("selezionata");
73     delay(1000); while(digitalRead(BTN_UP)==LOW);
74   }
75   // Decremento portata con pulsante
76   if(statoDown==HIGH&&letturaDown==LOW){
77     portata--; if(portata<0) portata=0;
78     Serial.print("Portata selezionata: "); Serial.println(portata+1);
79     lcd.clear(); lcd.setCursor(0,0);
80     lcd.print("Portata "); lcd.print(portata+1);
81     lcd.setCursor(0,1); lcd.print("selezionata");
82     delay(1000); while(digitalRead(BTN_DOWN)==LOW);
```

```
83  }
84  statoUp=letturaUp; statoDown=letturaDown;
85  int n=0; float N0_media=0; float N1_media=0;
86  while(n<N_MEASURE){
87    long sum0=0; long sum1=0;
88    for(int i=0;i<N_AVG;i++){
89      sum0+=analogRead(A0); sum1+=analogRead(A1); delayMicroseconds(200);
90    }
91    float N0=sum0/(float)N_AVG; float N1=sum1/(float)N_AVG;
92    N0_media+=N0; N1_media+=N1;
93    if(N0>0&&N1>0){
94      float K=N1/N0; float R=R3[portata]*K/(2.0-K);
95      if(R>0&&isfinite(R)){Rx[n]=R; n++;}
96    }
97  }
98  N0_media/=N_MEASURE; N1_media/=N_MEASURE;
99  float rapporto=N1_media/N0_media;
100 if(rapporto>1.5){
101   Serial.print("R3 = "); Serial.println(R3_portata[portata]);
102   Serial.println("R3 troppo piccola -> AUMENTARE la portata");
103   lcd.clear(); lcd.setCursor(0,0); lcd.print("R3 BASSA");
104   lcd.setCursor(0,1); lcd.print("AUMENTARE"); lcd.print(" PORTATA");
105 }else if(rapporto<0.5){
106   Serial.print("R3 = "); Serial.println(R3_portata[portata]);
107   Serial.println("R3 troppo grande -> DIMINUIRE la portata");
108   lcd.clear(); lcd.setCursor(0,0); lcd.print("R3 ALTA");
109   lcd.setCursor(0,1); lcd.print("DIMINUIRE"); lcd.print(" PORTATA");
110 }else{
111   float media=0;
112   for(int i=0;i<N_MEASURE;i++) media+=Rx[i];
113   media/=N_MEASURE;
114   float var=0;
```

```
115     for(int i=0;i<N_MEASURE;i++) var+=(Rx[i]-media)*(Rx[i]-media);
116     float sigma=sqrt(var/(N_MEASURE-1));
117     float uA=sigma/sqrt(N_MEASURE);
118     float uR3=R3_ERR*media;
119     float K=N1_media/N0_media;
120     float V0=N0_media/1024.0*5.0;
121     float uR12=2.0*R3[portata]/pow(2.0-K,2)*R12_ERR*K/V0;
122     float uADC=2.0*R3[portata]/pow(2.0-K,2)*K*
123     sqrt((pow(1.0/N0_media,2)+pow(1.0/N1_media,2))/120.0);
124     // Contributo OPAMP
125     float Req0=(R1*R2)/(R1+R2);
126     float Req1=(R3[portata]*media)/(R3[portata]+media);
127     float uVOS=VOS/sqrt(3); float uIB=IB/sqrt(3);
128     float  N0=sqrt(pow(uVOS,2)+pow(uIB*Req0,2));
129     float uN1=sqrt(pow(uVOS,2)+pow(uIB*Req1,2));
130     float uK=K*sqrt(pow(uN1/N1_media,2)+pow(uN0/N0_media,2));
131     float cK=2.0*R3[portata]/pow(2.0-K,2);
132     float uOPAMP=cK*uK;
133     float uc=sqrt(pow(uA,2)+pow(uR3,2)+pow(uR12,2)+pow(uADC,2)+pow(uOPAMP,2)
        );
134
135     // Risultati
136     lcd.clear(); stampaLCD(media,uc);
137     Serial.println("----------------");
138     Serial.print("Portata CORRETTA R3 = "); Serial.print(R3_portata[portata
        ]); Serial.println(" Ohm");
139     Serial.print("Rx = "); stampaRisultato(media,uc); Serial.println();
140     Serial.print("incertezza relativa = "); Serial.print(100.0*uc/media,3);
        Serial.println(" %");
141     Serial.println("----------------");
142     Serial.println("Contributi incertezza:");
143     Serial.print("uA = "); Serial.println(uA,6);
```

```
144     Serial.print("uR3 = "); Serial.println(uR3,6);
145     Serial.print("uR12 = "); Serial.println(uR12,6);
146     Serial.print("uADC = "); Serial.println(uADC,6);
147     Serial.print("uOPAMP = "); Serial.println(uOPAMP,6);
148     Serial.println("----------------");
149     Serial.print("uc = "); Serial.println(uc,6);
150     Serial.println("----------------");
151   }
152   delay(1000);
153 }
```

Script 4.14: Codice Arduino per l'implementazione dell'ohmetro con metodo a ponte

4.4 Confronto tra i due metodi per la misura di resistenza isolata

Il confronto tra il metodo mutuato dall'articolo [4] e quello basato sul principio di misura a ponte descritto nella **Sezione 4.3**, è stato eseguito mediante la misura di 25 resistori di diverso valore, i quali sono stati caratterizzati tramite un LCX200, considerato come strumento di riferimento. I dati raccolti dalle misure svolte tramite i due metodi sono stati organizzati nella **Tabella 4.5**, dove sono stati valutati anche gli errori relativi rispetto al valore misurato con l'LCX200. In seguito alla valutazione degli *errori relativi percentuali* $\epsilon_r[\%] = \dfrac{R_{mis} - R_{real}}{R_{real}} 100$ e delle *incertezze relative percentuali* $u_c[\%] = \dfrac{u_c(R)}{R} 100$, per entrambi i metodi, sono stati realizzati i grafici mostrati in **Figura 4.8**.

Tabella 4.5: Confronto dei risultati ottenuti applicando i due diversi metodi

R_{nom}	R_{real}	R_{ponte}	$u_{c,\text{ponte}}$	R_{articolo}	$u_{c,\text{articolo}}$	$\varepsilon_{\text{articolo}}$	$\varepsilon_{\text{ponte}}$
1 Ω	1.02 Ω	3.68 Ω	0.09 Ω	0.9 Ω	0.6 Ω	−9.8%	−13.3%
2.2 Ω	2.17 Ω	4.11 Ω	0.03 Ω	2.6 Ω	0.6 Ω	+19.9%	+88.9%
3.3 Ω	3.33 Ω	5.79 Ω	0.04 Ω	3.4 Ω	0.7 Ω	+2.2%	+74.2%
4.7 Ω	4.66 Ω	6.05 Ω	0.05 Ω	6.4 Ω	0.6 Ω	+36.5%	+30.1%
10 Ω	9.86 Ω	10.37 Ω	0.08 Ω	10.5 Ω	0.7 Ω	+6.8%	+5.2%
22 Ω	21.78 Ω	22.5 Ω	0.2 Ω	20.5 Ω	0.6 Ω	−5.7%	+3.5%
47 Ω	46.57 Ω	48.6 Ω	0.4 Ω	47.4 Ω	0.6 Ω	+1.8%	+4.3%
68 Ω	67.47 Ω	68.4 Ω	0.5 Ω	65.8 Ω	0.5 Ω	−2.5%	+1.3%
100 Ω	98.38 Ω	99.6 Ω	0.7 Ω	98.3 Ω	0.6 Ω	−0.1%	+1.2%
150 Ω	148.34 Ω	149 Ω	1 Ω	150.2 Ω	0.7 Ω	+1.3%	+0.5%
220 Ω	215.8 Ω	216 Ω	2 Ω	215.7 Ω	0.6 Ω	−0.1%	+0.0%
330 Ω	323.25 Ω	332 Ω	3 Ω	326.7 Ω	0.7 Ω	+1.1%	+2.8%
470 Ω	464.5 Ω	465 Ω	4 Ω	465.5 Ω	0.7 Ω	+0.2%	+0.1%
1 kΩ	990.09 Ω	989 Ω	7 Ω	980.1 Ω	0.7 Ω	−1.0%	−0.1%
1.5 kΩ	1.483 kΩ	1.47 kΩ	0.01 kΩ	1461.0 Ω	0.8 Ω	−1.0%	−0.5%
2 kΩ	1.954 kΩ	1.95 kΩ	0.02 kΩ	1934.0 Ω	0.8 Ω	−1.0%	−0.1%
2.2 kΩ	2.154 kΩ	2.15 kΩ	0.02 kΩ	2133.0 Ω	0.9 Ω	−1.0%	−0.1%
4.7 kΩ	4.585 kΩ	4.59 kΩ	0.04 kΩ	4.522 kΩ	0.002 kΩ	−1.4%	+0.0%
5.6 kΩ	5.522 kΩ	5.54 kΩ	0.04 kΩ	5.431 kΩ	0.002 kΩ	−1.6%	+0.3%
10 kΩ	9.915 kΩ	9.94 kΩ	0.07 kΩ	9.710 kΩ	0.003 kΩ	−2.1%	+0.3%
15 kΩ	14.834 kΩ	14.8 kΩ	0.1 kΩ	14.445 kΩ	0.005 kΩ	−2.6%	−0.2%
22 kΩ	21.618 kΩ	21.6 kΩ	0.2 kΩ	20.899 kΩ	0.008 kΩ	−3.3%	−0.3%
47 kΩ	46.045 kΩ	46.4 kΩ	0.4 kΩ	43.95 kΩ	0.03 kΩ	−4.6%	+0.8%
100 kΩ	97.853 kΩ	97.8 kΩ	0.7 kΩ	93.52 kΩ	0.08 kΩ	−4.4%	+0.0%
1 MΩ	1.005 MΩ	1.005 MΩ	0.007 MΩ	1.018 MΩ	0.007 MΩ	+1.3%	+0.0%

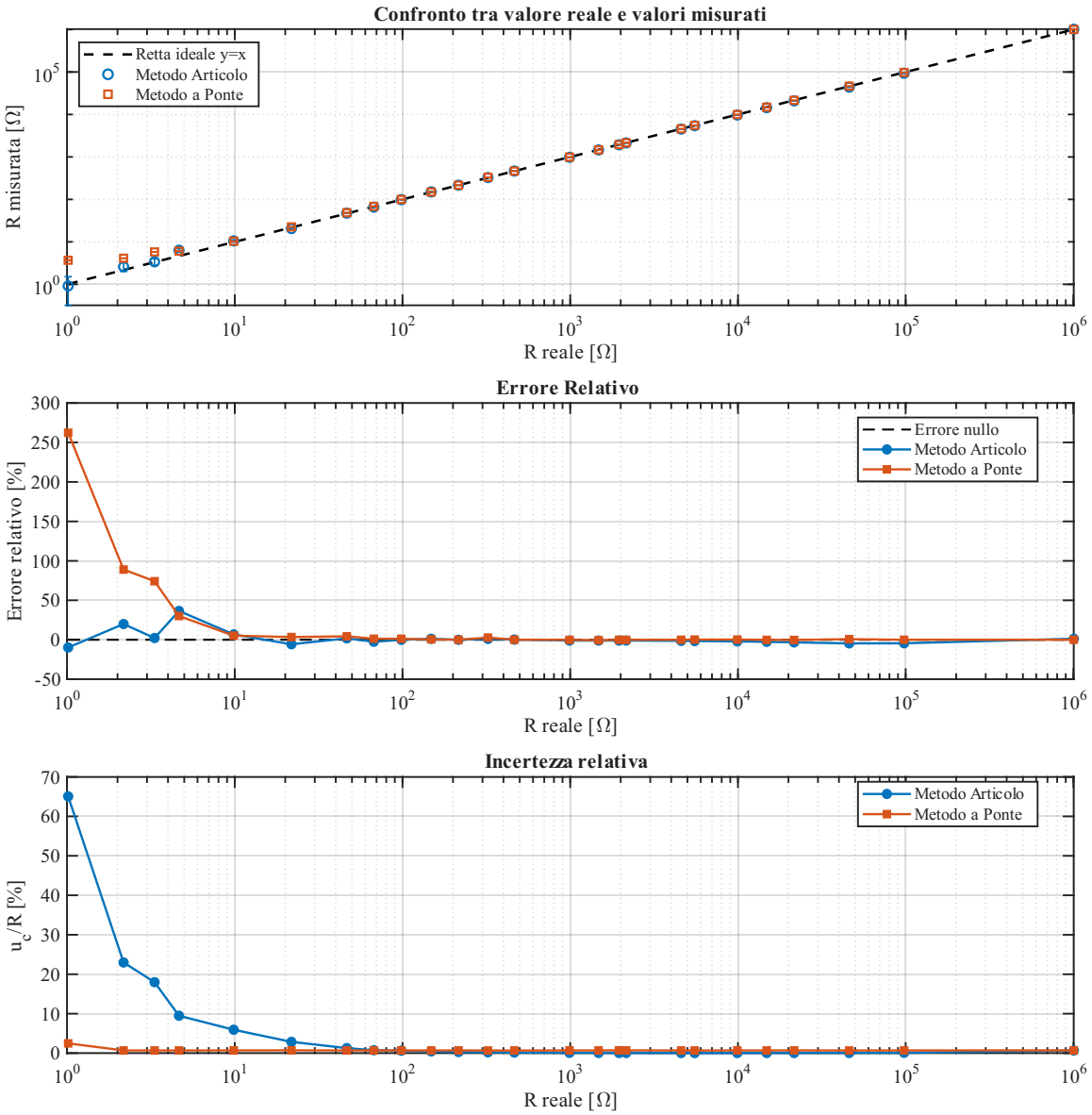


Figura 4.8: Grafici realizzati per il confronto tra i due metodi.

Analisi comparativa dei risultati

Il confronto tra i due metodi evidenzia un comportamento differente in funzione del valore della resistenza misurata. Come mostrato in **Tabella 4.5** e in **Figura 4.8**, per valori di resistenza dell'ordine delle centinaia di ohm e superiori, il metodo a ponte fornisce risultati complessivamente prossimi al valore considerato "reale", ovvero quello caratterizzato mediante l'LCX200.

In particolare, per resistenze comprese tra 68 Ω e 1 MΩ, l'errore relativo del metodo

a ponte risulta compreso, in valore assoluto, tra circa 0.0% e 4.6%. Per contro, alle resistenze più basse si osserva un marcato aumento dell'errore relativo: per $1\,\Omega$, $2.2\,\Omega$, $3.3\,\Omega$ e $4.7\,\Omega$ si ottengono rispettivamente errori relativi pari a -13.3%, $+88.9\%$, $+74.2\%$ e $+30.1\%$. Tale comportamento evidenzia una limitazione del metodo a ponte nella misura di resistenze di basso valore.

Una possibile causa di tale comportamento è legata al valore della resistenza R_3 utilizzato nella misura con la portata più bassa. In tale configurazione è stata utilizzata una resistenza R_3 di circa $10\,\Omega$, posta in serie alla resistenza incognita R_x.

Per valori molto piccoli di R_x, la corrente che attraversa il ramo può quindi assumere valori elevati, essendo approssimativamente determinata da $I = \dfrac{V_{ref}}{R_3 + R_x}$.

Ad esempio, per R_x dell'ordine di pochi ohm e con una tensione di alimentazione di $5\,\mathrm{V}$, la corrente può raggiungere alcune centinaia di milliampere. La conseguente dissipazione di potenza nei resistori può provocarne il riscaldamento e, in presenza di resistori caratterizzati da un TCR (*Temperature Coefficient of Resistance*) non trascurabile, alterare il valore di resistenza durante il processo di misura.

Tale fenomeno può quindi contribuire all'errore osservato alle basse resistenze.

L'ipotesi è inoltre coerente con il fatto che l'errore diminuisce sensibilmente all'aumentare di R_x, quando la corrente nel ramo di misura si riduce.

È inoltre evidente che le prestazioni del metodo a ponte dipendono in maniera significativa dalle caratteristiche dei componenti impiegati nella sua realizzazione. Oltre alla tolleranza dei resistori, assumono particolare rilevanza il coefficiente di temperatura della resistenza (TCR), la stabilità termica e i parametri parassiti del circuito, oltre agli effetti introdotti dallo stadio buffer, quali tensione di offset e correnti di polarizzazione dell'amplificatore operazionale. La scelta e la qualità dei componenti utilizzati influenzano pertanto direttamente l'accuratezza della misura.

Nel metodo proposto nell'articolo, invece, le prestazioni risultano maggiormente legate alle caratteristiche intrinseche della scheda e dell'architettura di acquisizione utilizzata, rendendo il risultato meno dipendente dalla realizzazione di un ramo resistivo esterno. Nel complesso, i risultati mostrano che il metodo a ponte consente di ottenere una buona accuratezza per un ampio intervallo di valori resistivi, ma pre-

senta una degradazione significativa delle prestazioni nella misura delle resistenze più basse. Il metodo considerato nell'articolo presenta invece, nel complesso, una maggiore uniformità dell'errore all'interno dell'intervallo analizzato. Il confronto evidenzia quindi come la scelta dell'architettura di misura debba tenere conto non solo dell'intervallo di resistenze da misurare, ma anche delle caratteristiche elettriche e termiche dei componenti impiegati nella realizzazione del circuito.

Capitolo 5

Conclusioni

Il presente lavoro di tesi ha avuto come obiettivo lo studio e la caratterizzazione di differenti metodologie per la misura dei parametri elettrici R, C e L, con particolare attenzione alla valutazione sperimentale delle prestazioni e dei principali limiti associati a ciascun metodo. L'attività svolta ha permesso di analizzare sia strumenti di misura dedicati, sia soluzioni basate su piattaforme a basso costo e microcontrollori, mettendone in evidenza le caratteristiche in relazione all'intervallo di misura e alle condizioni operative.

Il confronto sperimentale tra il misuratore LCX200 e la piattaforma Moku:Go ha evidenziato come quest'ultima consenta di effettuare la caratterizzazione dei componenti attraverso l'analisi della funzione di trasferimento, fornendo risultati ripetibili in una regione intermedia dell'intervallo di frequenze di interesse. La ripetibilità tende tuttavia a degradarsi agli estremi della banda analizzata e risulta fortemente dipendente dal valore dell'impedenza del componente. In particolare, per i condensatori, l'aumento della capacità comporta uno spostamento verso frequenze inferiori della regione in cui la variabilità della misura tende ad aumentare, mentre per le induttanze di basso valore, soprattutto nel caso di 1 µH, il ridotto rapporto segnale/rumore limita significativamente l'affidabilità della misura. Alle frequenze più elevate assumono inoltre maggiore rilevanza gli effetti parassiti del componente e del circuito di misura.

La realizzazione del sistema RCL basato sul microcontrollore ATmega328P ha permesso di verificare sperimentalmente la possibilità di implementare sistemi di misura

dei parametri elettrici mediante una piattaforma a basso costo. Le prove condotte su resistenze e capacità isolate e su reti RC serie e parallelo hanno consentito di caratterizzare il comportamento del sistema e di valutarne le prestazioni mediante il confronto con valori di riferimento. L'attività ha inoltre evidenziato come le prestazioni di un sistema di questo tipo siano strettamente legate alle caratteristiche dell'architettura di acquisizione, alla risoluzione disponibile e alle modalità con cui vengono elaborate le grandezze misurate.

Particolare attenzione è stata infine dedicata alla realizzazione di un ohmetro basato sul metodo a ponte. Il confronto con il metodo precedentemente implementato e con il valore di riferimento ottenuto mediante LCX200 ha mostrato che il metodo a ponte consente di ottenere risultati soddisfacenti per un ampio intervallo di valori resistivi, in particolare per resistenze dell'ordine delle centinaia di ohm e superiori. Al contrario, per le basse resistenze si osserva una significativa degradazione dell'accuratezza, riconducibile anche alla maggiore influenza delle resistenze parassite, degli effetti termici e delle caratteristiche dei componenti utilizzati nella realizzazione del circuito. Il metodo risulta pertanto fortemente condizionato dalla scelta e dalla stabilità dei componenti impiegati, nonché dalle condizioni operative durante la misura.

Nel complesso, il lavoro ha evidenziato come non esista un'unica metodologia in grado di garantire prestazioni ottimali per qualsiasi intervallo di valori e condizione di misura. La scelta del metodo deve essere effettuata considerando il parametro da misurare, il suo valore, la banda di frequenza di interesse e le caratteristiche dell'architettura di misura. I risultati ottenuti mostrano inoltre come soluzioni basate su piattaforme programmabili possano rappresentare una valida alternativa per la realizzazione di sistemi di misura flessibili e a basso costo, a condizione di considerare attentamente le principali sorgenti di incertezza e i limiti imposti dall'elettronica utilizzata.

Bibliografia

[1] IEEE. Electronic circuits. [Online]. Available: https://technav.ieee.org/topic/electronic-circuits/

[2] P. Horowitz and W. Hill, *The Art of Electronics*, 3rd ed. Cambridge University Press, 2015.

[3] Keysight Technologies, *Impedance Measurement Handbook: A Guide to Measurement Technology and Techniques*, 2016. [Online]. Available: https://www.keysight.com/us/en/assets/7018-06840/application-notes/5950-3000.pdf

[4] P. M. C. Inácio, R. Guerra, and P. Stallinga, "An Ultra-Low-Cost RCL-Meter," *Sensors*, vol. 22, no. 6, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/6/2227

[5] C. K. Alexander and M. N. O. Sadiku, *Fundamentals of Electric Circuits*, 6th ed. McGraw-Hill Education, 2017.

[6] F. Fuso, "Ponti in corrente alternata," Dispense del corso di Laboratorio di Fisica, consultato il 7 agosto 2026. [Online]. Available: https://osiris.df.unipi.it/~fuso/dida/ponti.pdf