

Modellizzazione della resistenza di diodi a giunzione PN per alte correnti di lavoro

L. Ciucci(*) S. Bruzzesi(*) M. Romagnoli(*) M. Alighieri(*)
B. Tomelleri(*)

2020/11/01

1 INTRODUZIONE

L'alta resistività intrinseca al silicio, di cui è composta la giunzione bipolare, comporta la presenza di una sua componente resistiva: questa risulta sempre meno trascurabile agli effetti del passaggio di corrente attraverso il diodo, all'aumentare della tensione ai suoi capi e della sua temperatura. Per poter modellare la componente -ohmica- di un diodo percorso da correnti alte si propone un modello semplice di resistenza parassita in serie, in grado di descriverne gli effetti, verificandone sperimentalmente la validità.

2 CENNI TEORICI

Secondo le ipotesi, un diodo reale può essere schematizzato quale un resistore ohmico ed un diodo ideale in serie. Pertanto ci aspettiamo che la corrente che attraversa tali elementi sia la medesima:

$$I = \frac{\Delta V_{\text{resistore}}}{R} = I_0 \left(e^{\frac{\Delta V_{\text{diodo}}}{\eta V_T}} - 1 \right) \quad (1)$$

dunque la relazione che lega la corrente e la tensione ai capi del diodo può essere espressa secondo la legge:

$$\Delta V = \Delta V_d + \Delta V_r = \eta V_T \ln \left(\frac{I + I_0}{I_0} \right) + RI \quad (2)$$

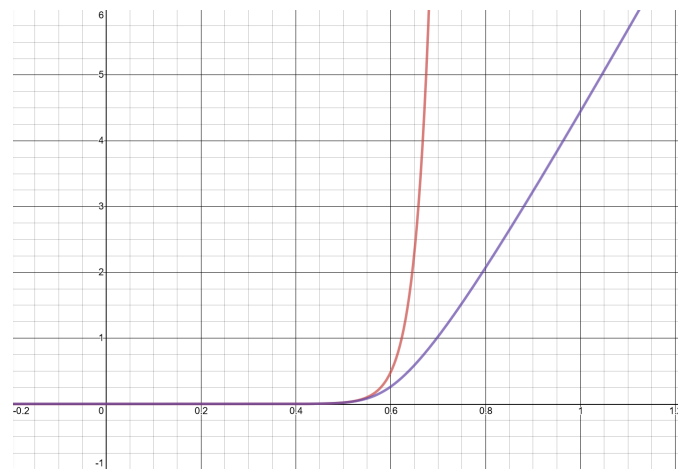


Figura 1: confronto fra curva di Shockley (in rosso) e la legge (2) (in viola)

*Dipartimento di Fisica E. Fermi, Università di Pisa - Pisa, Italy

3 METODO E APPARATO SPERIMENTALE

Il circuito sotto studio lavora con correnti relativamente alte, per evitare sostanziali aumenti di temperatura nei componenti ed eventuali danni agli stessi si imprinono correnti impulsive. La durata degli impulsi è di circa $150 \mu s$ (da misurare meglio) che corrisponde ad un'energia impressa inferiore a 1 mJ dunque ad un aumento della temperatura del semiconduttore inferiore a 0.5 K.

3.1 Apparato

L'apparato sperimentale è costituito da un circuito, realizzato su basetta sperimentale, il cui scopo è generare correnti impulsive attraverso il diodo $D1$ e la resistenza $R1$. La misura della differenza di tensione ai capi di questa permette stima della corrente che attraversa il diodo, per esplorare ampie zone di lavoro del diodo è stata cambiata opportunamente scegliendo tra le seguenti resistenze:

$R1$ nom. $[\Omega]$	$R1$ mult. $[\Omega]$
$0.22 \pm 3\%$	0.226 ± 0.008
$2.2 \pm 5\%$	2.212 ± 0.008
$22 \pm 5\%$	21.86 ± 0.010
$220 \pm 5\%$	216.22 ± 0.07
$2.2k \pm 5\%$	2202.1 ± 0.4
$22k \pm 5\%$	$(21.7 \pm 0.3)10^3$
$0.22M \pm 5\%$	$(217 \pm 3)10^3$

Tabella 1: I valori delle resistenze poste in serie al diodo, riportate in valore nominale e misurate con multimetri digitali.

Nel circuito può essere individuata una parte che si occupa della carica e scarica controllata del condensatore $C1$, il cui scopo è quello di fornire una tensione che possa essere facilmente regolata e sufficientemente stabile durante un impulso. Un'altra parte del circuito si occupa di collegare a comando il condensatore alla serie $R1 - D1$ attraverso un MOS-FET.

Il circuito è alimentato da 2 tensioni fornite da un alimentatore stabilizzato switching e da un Buck Boost Converter, un segnale fornito su $T2$ provoca la carica del condensatore mentre su $T3$ la sua scarica. Un segnale (invertito) su $T1$ innesca l'impulso di corrente sul diodo.

La tensione di $C1$ è misurata attraverso un partitore di tensione collegato ad $OUT3$ ed alla scheda di controllo mentre la tensione ai punti $OUT1$ e $OUT2$ è letta direttamente.

Come MCU per la gestione dell'apparato è stata scelta la scheda **Teensy 3.2**[1], questa si occupa del controllo dei segnali e delle letture analogiche. In particolare **Teensy** permette la lettura analogica sincronizzata differenziale veloce, essendo dotato di due ADC, entrambi con una risoluzione reale di 12 bit. La lettura differenziale è fondamentale per acquisire coppie di dati per le tensioni ai capi del diodo e della resistenza.

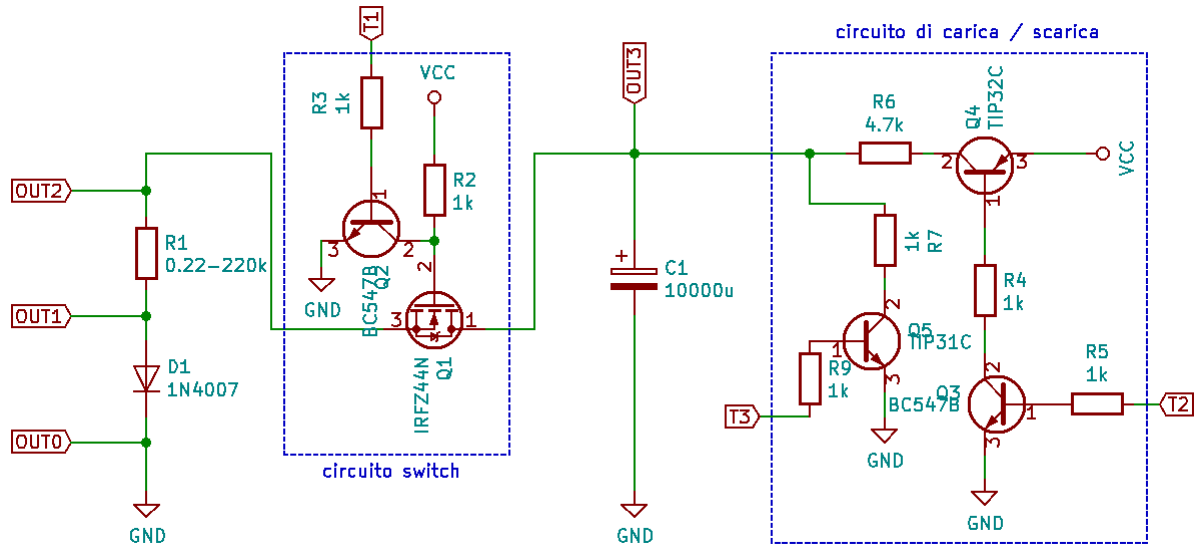


Figura 2: Circuito globale per la gestione del diodo

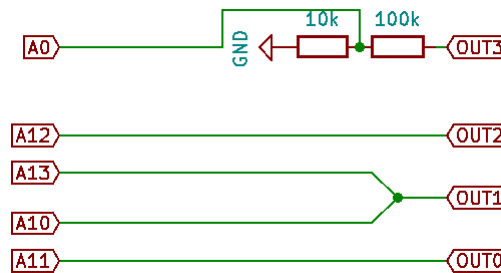


Figura 3: Schema circuitale del sistema di lettura (verso Teensy)

3.2 Calibrazione

I due convertitori ADC della MCU sono stati calibrati.

3.3 Acquisizione dati

Per ogni valore scelto della resistenza $R1$ è stata eseguita una presa dati automatizzata secondo una routine programmata in **Teensy**: il condensatore viene caricato ad una tensione prefissata, una volta raggiunto il valore prestabilito viene inviato il segnale su $T1$, si avvia l'acquisizione sincronizzata e si attende un tempo $50 \mu s$ per permettere al MOS-FET di entrare in conduzione ($R2$ è stata scelta di $1k$, dunque c'è un apprezzabile ritardo tra segnale e impulso) e scartare eventuali segnali spuri; a questo punto si inizia a memorizzare una serie da 100 coppie di letture -sincronizzate- ((in nota) a meno di sfasamenti nell'ordine delle centinaia di nanosecondi?), si ferma l'impulso e si trasmettono i dati al computer; se la tensione al punto $OUT2$ è stata maggiore di $3.3V$ si ferma l'acquisizione per non danneggiare **Teensy**, altrimenti si ricomincia caricando il condensatore ad una tensione più alta.

4 ANALISI DATI E RISULTATI

Il nostro intento è verificare che i dati raccolti siano in accordo con la legge (2). Tramite la legge di Ohm è possibile ricavare la corrente nel diodo, dividendo la caduta di tensione misurate dalla boccia ADC0 per la resistenza R . Si è quindi effettuato un filtraggio volto all'eliminazione degli outliers e dei punti

meno significativi, assumendoli quali variabili indipendenti e di natura gaussiana. Per una discussione dettagliata del sistema di filtraggio dati si rimanda all' [Appendice A: Filtraggio Dati](#).

Successivamente, è stato effettuato un fit sulla base dei dati selezionati. Ingenuamente, avremo dunque potuto pensare di adottare la legge (2) direttamente. Tuttavia ciò comporterebbe il fallimento repentino del fit a causa dei valori negativi o nulli che debitamente si riscontrerebbero entro il logaritmo. È stata quindi adottata una legge alternativa basata sul metodo delle tangenti (o di Newton). Per una trattazione approfondita del modello di fit, si rimanda all' [Appendice B: Metodo di Fit](#).

I dati raccolti con sovrapposta la funzione di fit sono stati posti all'interno del grafico 4. I parametri

Figura 4: La curva caratteristica del diodo ricostruita dai nostri dati, con sovrapposto il modello di best-fit

stimati dal fit risultano essere:

$$\begin{aligned} R_{diodo} &= 47.5 \pm 0.2 \text{ m}\Omega \\ \eta V_T &= 46.40 \pm 0.06 \text{ mV} \\ I_0 &= (3.18 \pm 0.05) \text{ nA} \\ \sigma_{R, \eta V_T} &=? \\ \sigma_{R, I_0} &=? \\ \sigma_{I_0, \eta V_T} &=? \\ \chi^2 / \text{ndof} &= 4483 / 2760 \\ \text{abs_sigma} &= \text{False} \end{aligned}$$

Il χ^2 risulta essere pari a 4483 contro un aspettato di 2760. Successivamente, i dati e la funzione sono stati posti all'interno di un grafico in scala semilogaritmica (grafico 4).

Figura 5: La stessa curva $V - I$ del diodo in scala semilogaritmica, con sovrapposto il modello di best-fit

5 CONCLUSIONI

L'andamento dei dati sperimentali non risulta essere ben descritto dalla legge (2). Ciò può essere spiegato sulla base della non linearità dei convertitori analogici digitali all'interno di **Teensy 3.2**, che ha comportato degli errori difficilmente stimabili e presumibilmente correlati. Tuttavia, si riscontrano delle analogie tra la funzione di fit ed i dati ottenuti. In particolare, dal grafico in scala lineare, si osserva che a correnti alte l'andamento risulta essere pressochè lineare in accordo con l'ipotesi sulla componente resistiva interna. Dal grafico in scala semilogaritmica, inoltre, osserviamo che i dati risultano possedere un andamento simile a quello caratteristico della curva di Shockley, ovvero approssimativamente rettilineo, per poi appiattirsi come un logaritmo al crescere della tensione, esattamente come ci aspetteremo da un andamento lineare. Pertanto potremo aspettarci che i parametri stimati dal fit siano significativi e che la modellizzazione proposta sia una buona approssimazione del diodo reale nelle condizioni di lavoro considerate.

6 APPENDICE A: FILTRAGGIO DATI

6.1 Introduzione

Durante l'esperienza si è raccolto un grande numero di dati, acquisiti in run diversi in base alla resistenza scelta, dunque a zone differenti della curva. Si è posto il problema di eliminare degli outliers in modo indipendente dal modello scelto. Inoltre le varie serie di dati si sovrappongono, dunque è necessario eliminare quei dati che, non aggiungendo informazioni utili, vanno a "sporcare" il grafico.

Il sistema di filtraggio di dati implementato nell'eseguibile si compone di 2 fasi: la prima è l'eliminazione di outliers, la seconda consiste nell'eliminazione di dati non significativi.

6.2 Procedimento

Supponiamo di avere una serie di dati (x, y) e assumiamo che siano indipendenti tra loro. Questo non sarà in generale vero, però quest'ipotesi è tanto più lecita quanto più la correlazione tra le varianze delle misure su x e y è indipendente dai valori assunti dalle x e y stesse e, più sono numerosi i dati racchiusi entro una deviazione lungo x : in questo caso, infatti, la correlazione viene inclusa nella varianza lungo y . Supponiamo inoltre che siano note a priori le $\sigma_x^2 := \text{Var}(x)$ e che la loro distribuzione di probabilità sia normale (le distribuzioni delle componenti sono approssimativamente gaussiane per il convertitore di **Teensy**, perlomeno utilizzando la risoluzione a 12 bit) secondo una matrice di covarianza diagonale nella base $\{x, y\}$. In ogni modo, i nostri dati x e y risultano indipendenti e approssimativamente normali, dunque le assunzioni risultano giustificate. Allora la densità di probabilità che un punto che abbia misurato x si trovi a tale ascissa x_i si ricava banalmente integrando lungo y a x fissata:

$$dP = \frac{1}{\sigma_{x_i} \sqrt{2\pi}} e^{-\frac{1}{2} \frac{(x-x_i)^2}{\sigma_{x_i}^2}} dx.$$

Dunque, ripetendo più volte la stessa misura, la probabilità

$$P\left(|x - x_i| \leq \frac{\varepsilon}{2}\right) = \varepsilon G_{x_i}.$$

dove

$$G_{x_i} := \frac{1}{\sigma_{x_i} \sqrt{2\pi}} e^{-\frac{1}{2} \frac{(x-x_i)^2}{\sigma_{x_i}^2}}.$$

e $\varepsilon > 0$ è piccolo a piacere. Scegliendo allora solo quelle misure x per cui vale $|x - x_i| \leq \frac{\varepsilon}{2}$, queste saranno in numero intorno a:

$$N_i := N_{\text{tot}} \frac{G_{x_i}}{\sum_j G_{x_j}} = N_{\text{tot}} w_i.$$

che definisce implicitamente i pesi w_i con cui si mediano le distribuzioni di probabilità gaussiane G_{x_i} . Allora, detto:

$$G_{y_i} := \frac{1}{\sigma_{y_i} \sqrt{2\pi}} e^{-\frac{1}{2} \frac{(\mu_y - y_i)^2}{\sigma_{y_i}^2}}.$$

Per il principio di massima verosimiglianza siamo quindi interessati a massimizzare la quantità:

$$\mathcal{L} = \prod_{i=1}^n \prod_{j=1}^{N_i} G_{y_i} = \prod_{i=1}^n G_{y_i}^{N_i}.$$

Per la monotonia del logaritmo il problema equivale a massimizzare la quantità:

$$\ln \mathcal{L} = \sum_{i=1}^n \ln G_{y_i}^{N_{\text{tot}} w_i} = \frac{N_{\text{tot}}}{\sum_{j=1}^n G_{x_j}} \sum_{i=1}^n G_{x_i} \ln G_{y_i}.$$

Per cui, a meno di costanti risulta:

$$\ln \mathcal{L} - \text{const.} \propto \sum_{i=1}^n -G_{x_i} \ln \sigma_y - \frac{1}{2} G_{x_i} \left(\frac{y_i - \mu_y}{\sigma_y} \right)^2 \quad (3)$$

Imponendo la condizione di stazionarietà rispetto a μ_y si ottiene dunque:

$$\mu_y = \sum_{i=1}^n y_i w_i \quad (4)$$

Una volta sostituito in (3) quanto appena trovato per μ_y e imponendo la stessa condizione di stazionarietà rispetto a σ_y si ha:

$$\sigma_y^2 = \sum_{i=1}^n (y_i - \mu_y)^2 w_i \quad (5)$$

Infine è possibile ricavare la varianza di μ_y dalla definizione di valore di aspettazione, riconducendola più volte a integrali di gaussiane di altezze e ampiezze diverse:

$$\text{Var}(\mu_y) = \sum_{i=1}^n w_i^2 \sigma_y^2 + \left(\frac{y_i}{\sum_{j=1}^n w_j} \right)^2 \frac{e^{-\frac{(x-x_i)^2}{3\sigma_{x_i}^2}} + \sqrt{3} \left(e^{-\frac{(x-x_i)^2}{\sigma_{x_i}^2}} - \sqrt{2} e^{-3\frac{(x-x_i)^2}{4\sigma_{x_i}^2}} \right)}{2\sqrt{3}\pi\sigma_{x_i}^2} \quad (6)$$

Riassumendo:

Nella (4) prendiamo una media dei campionamenti intorno ad un'ascissa x in esame, pesata sulla distanza che gli x_i hanno da questa; intuitivamente lo interpretiamo come se stessimo applicando un *blur a kernel gaussiano* ai punti acquisiti. Effettivamente quello che stiamo facendo non è molto diverso da una stima di densità di kernel monovariante, dove però nel nostro caso riscaliamo la stima in base al valore assunto da y . Lo stesso ragionamento vale per σ_y^2 , si ha una stima della varianza dei dati la variare di y , pesata sulla distanza dai valori studiati. Dunque $\mu_y \pm \sigma_y$ ci dà una stima della distribuzione dei nostri dati.

6.3 $\text{Var}(\mu_y)$

Mentre σ_y rappresenta la distribuzione dei dati intorno al valor medio μ_y , $\text{Var}(\mu_y)$ ci dà un'idea dell'incertezza che attribuiamo alla miglior stima di y . Questo ci è utile per determinare la convergenza della stima in funzione dei dati acquisiti. Infatti: più la densità dei dati è grande rispetto alla deviazione standard σ_x , più la stima del valore centrale risulta precisa. Graficamente la banda di confidenza è più ristretta dove si concentrano i dati, viceversa tende ad allargarsi dove i dati sono sparsi, a distanze paragonabili a σ_x . Numericamente, si vede dalla seconda somma nell'espressione (6) che la stima del valore centrale è statisticamente significativa solo quando si media su un intervallo campionato con almeno qualche punto ogni deviazione σ_x : altrimenti $\sigma_y \rightarrow 0$ indicando assenza di dati, mentre $\text{Var}(\mu_y)$ tende a $+\infty$ come $\sim e^{x^2}$ indice della stessa insufficienza di dati al fine di stabilire con precisione significativa il valore di μ_y .

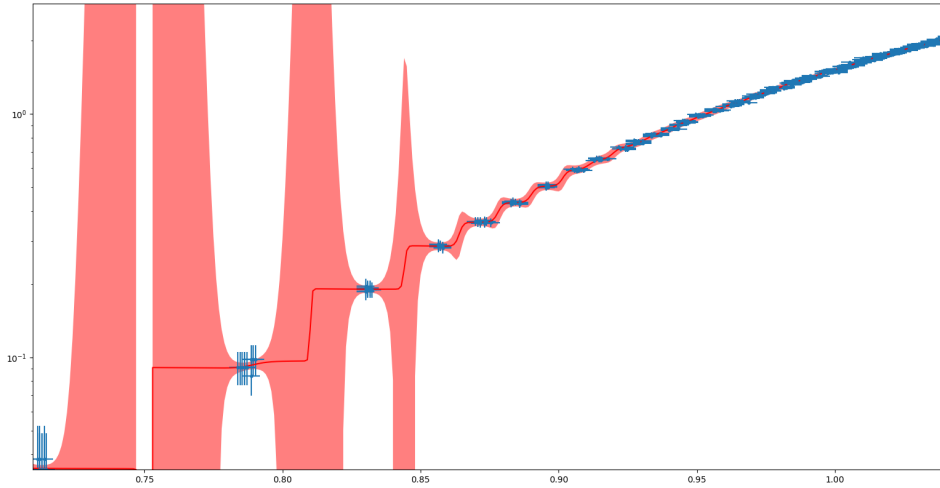


Figura 6: La media μ_y è rappresentata dalla linea rossa, mentre l'area in rosso indica il valore di $\text{Var}(\mu_y)$ al variare dei dati (in blu) lungo x .

Nel caso opposto, in cui i dati sono "densi" (in confronto alle σ_x) la seconda somma, per quanto computazionalmente intensiva, numericamente sembrerebbe piccola in confronto alla prima: in realtà non lo è, ma soprattutto questa non può essere trascurata, poiché è proprio la quantità che descrive la dipendenza dalla densità stessa, dunque la caratteristica convergenza/divergenza della precisione sulla stima centrale fornita.

6.4 Filtro outliers

La parte più semplice nel filtraggio dati consiste nello scartare tutti quei punti che distano da μ_y più di una soglia arbitraria k di deviazioni standard σ_y (nel nostro caso è stato scelto $k = 2$, non critico, trovato dopo una serie di prove). A differenza del classico metodo basato sulla distanza della curva modello di best fit, non siamo influenzati da quest'ultimo. Questo risulta particolarmente utile in simili situazioni di verifica del modello in quanto una selezione basata su un preliminare fit risulterebbe influenzata dalla scelta della funzione in questione e eliminerebbe tutti i dati che non risultano compatibili con essa.

6.5 Filtro dati non significativi

Supponiamo di avere 2 set di dati fatti con diverse resistenze, il primo (A) con una resistenza bassa, il secondo (B) con una alta: Il primo set esplorerà la regione ad alta corrente, mentre il secondo la regione di basse correnti. In generale i dati del primo si sovrapporranno anche nelle zone basse esplorate dal secondo, però senza aggiungere sostanziali informazioni rispetto a quanto farebbe il secondo. Esponiamo dunque il criterio sviluppato per ridurre l'influenza di questi punti meno significativi sulla ricerca dei parametri di best-fit e sulla rappresentazione finale dei dati.

Per capire se in un certo punto i dati di A sono significativi, calcoliamo la misura di significatività che abbiamo sviluppato in (6): $\text{Var}(\mu_y)$ di A e di B . Perciò se $\text{Var}(\mu_y)$ di A è maggiore di $q\text{Var}(\mu_y)$ di B , con q arbitrario (nell'esperienza è stato scelto $q = 3$), questo indica che i dati di A ci stanno dando meno informazioni rispetto a quelli di B . A questo punto è sufficiente controllare tutti i punti scorrendo su tutte le combinazioni di set per eliminare i dati non significativi, che rendono meno immediata l'interpretazione il grafico. Questo è ben visibile in scala logaritmica sulle y dove i punti con grandi incertezze o varianze tendono a disperdersi rapidamente. L'algoritmo è computazionalmente intensivo e richiede una corretta gestione della memoria per evitare bolle di allocazione, dunque è stato implementato in C++ per praticità e richiamato all'interno degli script. Nelle figure di esempio sono mostrati i dati selezionati dall'algoritmo (in nero) ed i dati scartati (in rosso). E' infine mostrato il confronto dei grafici delle $\text{Var}(\mu_y)$ tra due set successivi.

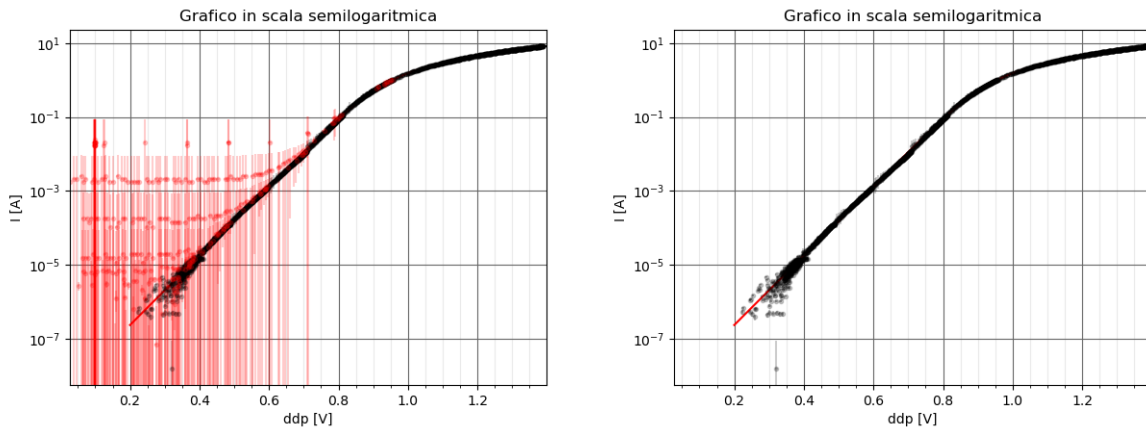


Figura 7: Grafici in scala semilogaritmica dei prima (sinistra) e dopo (destra) del filtraggio dati. I dati scartati sono stati evidenziati in rosso. Per praticità è stato rappresentato un centesimo dei dati raccolti

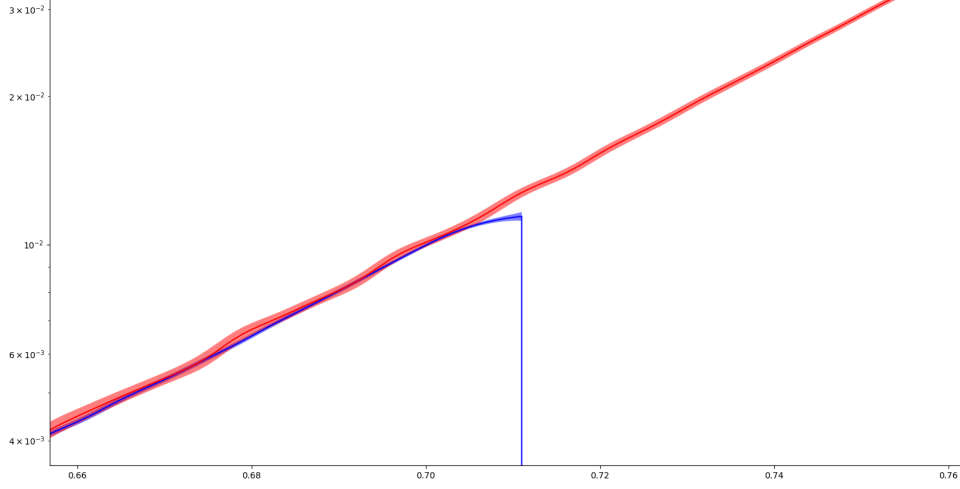
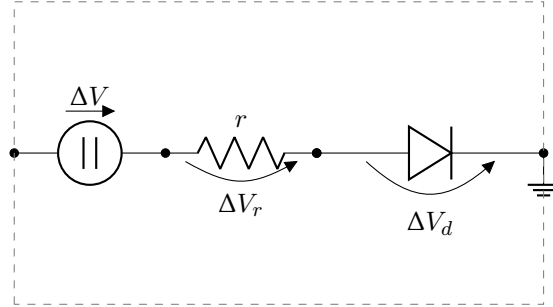


Figura 8: Confronto dei grafici delle $\text{Var}(\mu_y)$ su due set di dati consecutivi.

7 APPENDICE B: METODO DI FIT

Partiamo dall'osservazione che la tensione ai capi del diodo può essere scomposta in ΔV_r e ΔV_d come illustrato in figura:



dove il resistore a sinistra indica proprio la resistenza di un diodo reale, mentre il diodo alla sua destra rappresenta una giunzione bipolare PN, ossia il diodo ideale descritto dall'equazione di Shockley. Dunque ci aspettiamo che la tensione V campionata con Teensy ai capi del diodo rispetti la legge:

$$V = \Delta V = \Delta V_d + \Delta V_r = \eta V_T = \ln \frac{I + I_0}{I_0} + RI \quad (7)$$

All'interno dello script `fit_file.py`, `voltages` rappresenta il vettore delle ascisse dei punti campionati, mentre `currents` rappresenta l'analogo per le ordinate. Le incertezze associate a questi due all'interno dello script prendono i nomi `voltageErrs` e `currentErrs`. Ingenuamente potremmo essere tentati di effettuare una sorta di fit "inverso" (delle x in funzione delle y). Tuttavia una tale operazione non è affatto banale, a causa dei valori negativi/nulli che debitamente si riscontreranno all'interno del logaritmo nella legge (7), che possono facilmente portare al fallimento dell'algoritmo di fit. È per questo che si è dovuto definire il modello "inverso" nella forma di una legge di natura approssimata per ricorsione, grazie al metodo di Newton. Al livello di implementazione si è quindi dichiarata una funzione:

```
def curr(V, I0, nVt, R):
    v = V
    for i in range(Nstep):
```



```

a = deriv_errFun(v, IO, nVt, R)
v = v - errFun(v, V, IO, nVt, R) / a
return (V - v)/R

```

Questa verrà in seguito usata come modello per la corrente in funzione della tensione ai capi del diodo reale nella routine di fit per minimi quadrati. La funzione `curr` cerca di -linearizzare- la relazione tra I e V nell'intorno di un determinato V . Come sappiamo infatti dal metodo di Newton: detta $f(x)$ una funzione tale che $f(x) = 0$ e, dato un valore iniziale tale che $f(x[0]) = \alpha$ generico, sappiamo che la relazione di ricorsione

$$x[0] = \alpha \quad (8)$$

$$x[N+1] = x[N] - \frac{f(x[N])}{f'(x[N])} \quad (9)$$

converge ad un valore approssimato per la nostra x , commettendo un errore sempre più piccolo al crescere di N , indice del livello di ricorsione raggiunto. Nel nostro caso x sarà quella tensione v per cui vale l'identità:

$$I(V) := I_0^{\frac{\Delta V}{\eta V_T}} = \frac{V - v}{R} \quad (10)$$

che identifica il tratto lineare descritto da una retta con coefficiente angolare negativo pari a $-\frac{1}{R}$ ed intercetta $I = v/R$ sull'asse delle ordinate; si tratta proprio della retta di carico del diodo. L'equazione ricorsiva (9) ci permette quindi di determinare il punto di lavoro v per cui la corrente (di lavoro) si può scrivere come $\frac{V-v}{R}$, come si vede dal return nella funzione di fit `curr(V, parametri liberi)`. A questo punto si può già intuire quali siano le altre due funzioni di appoggio richiamate dentro `curr`, di cui riportiamo le definizioni per completezza:

```

def sck(V, IO, nVt):
    return IO*(pylab.exp(V/nVt) - 1)

def errFun(V, V0, IO, nVt, R):
    return sck(V, IO, nVt) + (V - V0)/R

def deriv_errFun(V, IO, nVt, R):
    return IO / nVt * pylab.exp(V/nVt) + 1./R

```

Infatti `errFun` e `deriv_errFun` sono, rispettivamente, la relazione da minimizzare specificata dalla legge (10) e la sua derivata rispetto a V . In conclusione, si è effettuato un fit dei minimi quadrati implementato in Python mediante la funzione `curve_fit` dal modulo `optimize` interno alla libreria `Scipy`[2] adottando come modello `curr` e lasciando liberi tutti i suoi parametri. In particolare si è tenuto conto dell'incertezza sulla variabile indipendente con il metodo "dell'errore efficace": ovvero propagando gli errori `voltageErrs` sulle V , tramite le stime dei parametri ottenute da un fit preliminare, in cui prendiamo `currentErrs` come sole incertezze sulla variabile dipendente I . Lo stesso algoritmo di fit viene iterato più volte, però assumendo come incertezza sulle misure una sorta di errore efficace dato dalla somma in quadratura dei due contributi all'incertezza:

$$\Delta_{\text{eff}_i} := \sqrt{\left| \frac{d \text{curr}}{dV} \right|_{V=V_i}^2 \Delta_{V_i}^2 + \Delta_{I_i}^2}. \quad (11)$$

fin a che non converge ai valori ottimali dei parametri.

RIFERIMENTI BIBLIOGRAFICI

- [1] P. Stoffregen. Teensy 3.2 development board. Sherwood, Oregon, USA. [Online]. Available: <https://www.pjrc.com/teensy/teensy31.html>
- [2] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman,

N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, Í. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors, “SciPy 1.0—Fundamental Algorithms for Scientific Computing in Python,” *arXiv e-prints*, p. arXiv:1907.10121, Jul 2019.