



**Politecnico
di Torino**

Corso di Laurea Magistrale in Ingegneria Elettronica (Electronic Engineering)

Tesi di Laurea Magistrale

**Confronto tra differenti approcci
nelle simulazioni a livello
elettromagnetico/circuitale
per convertitori di tipo switching mode
power supply**

Relatori:

Prof. Fabio PARESCHI

Dott. Silvia SIMONE

Ing. Davide LENA

Candidato:

Emanuele PERRIA

Luglio 2024

Sommario

Al giorno d'oggi il tema degli elementi parassiti sta ricoprendo un ruolo di primo piano nell'ambito della progettazione di circuiti in tecnologie emergenti, come ad esempio in dispositivi SiC e GaN. Per le aziende sta diventando sempre più importante e conveniente tenerne conto perché tali non idealità non sono più trascurabili con lo scalamento della tecnologia e l'aumentare delle frequenze di funzionamento. Inoltre, i dispositivi prodotti devono rispettare i limiti dovuti, per esempio, alla compatibilità elettromagnetica, alla signal e power integrity e ad altri vincoli progettuali. Il rischio è che queste non idealità, se trascurate, non facciano più rispettare tali normative, o peggio, danneggino il circuito stesso durante il suo funzionamento. Lo scopo principale di questo studio di Tesi è confrontare i diversi approcci di co-simulazione tra elementi puramente descritti dal loro modello circuitale ed elementi parassiti estratti dalla simulazione elettromagnetica. Sono stati considerati diversi tool attualmente disponibili sul mercato che permettono di eseguire la simulazione elettromagnetica, in particolare Ansys e PathWave, di cui è stato descritto l'algoritmo interno per poterli analizzare e confrontare. Una volta eseguita la simulazione elettromagnetica, sono stati presi in considerazione i diversi formati forniti in output dai tool per poter essere utilizzati successivamente per la co-simulazione all'interno dell'ambiente Cadence Virtuoso. L'approccio descritto è stato applicato a un dispositivo fornito dall'azienda STMicroelectronics, in questo modo è stato possibile osservare come i parametri parassiti influiscono sulle prestazioni finali del dispositivo, rendendo necessaria la loro considerazione già nelle fasi iniziali di progettazione.

La Tesi è così suddivisa: nel primo capitolo sono spiegate le motivazioni che spingono le aziende a prestare particolare attenzione al tema degli elementi parassiti. Nel secondo, terzo e quarto capitolo sono descritti i tool utilizzati in questo studio, i principi fisici sui quali si basano i loro algoritmi interni, con delle simulazioni performate per verificare quanto spiegato. Il quinto capitolo riguarda le cosimulazioni finali nel dominio del tempo, che permettono di capire come effettivamente le non idealità influiscono sul funzionamento di un circuito che è stato preso come esempio. Il sesto ed ultimo capitolo è dedicato alle conclusioni.

Ringraziamenti

Alla mia famiglia che mi ha sempre sostenuto. Vi voglio bene.

Indice

Elenco delle figure	VII
1 Simulazioni elettromagnetiche	1
1.1 L'importanza delle simulazioni elettromagnetiche	1
1.2 Flusso per la cosimulazione	4
1.3 Parametri S	6
1.4 Passivity, causality, stability	8
2 Simulatori utilizzati	11
2.1 Q3D	11
2.1.1 Calcolo dei parametri S con Q3D [10]	13
2.1.2 Calcolo del circuito RLCG ad una certa frequenza con Q3D [10]	15
2.1.3 Calcolo della broadband netlist con Q3D [10]	26
2.2 ADS	26
2.2.1 Momentum RF e Momentum Microwave	27
2.2.2 FEM	33
2.2.3 Calcolo della broadband netlist con ADS	37
2.2.4 Calcolo della broadband netlist con Matlab	37
2.3 EMPro	37
2.4 SPICE	40
3 Analisi in frequenza	43
3.1 Variazione dei parassiti RLCG al variare della frequenza	43
3.1.1 Skin-effect	44
3.1.2 Calcolo di R_{DC} e L_{DC} con Q3D	48
3.1.3 Calcolo di R_{AC} e L_{AC} con Q3D	50
3.1.4 Calcolo di C e G con Q3D	51
3.2 Simulazioni onda quadra in ingresso	51

4	Analisi nel tempo con parametri variabili in frequenza in Cadence	56
4.1	SPICE: simulazione transient dei parametri S, della broadband netlist e del circuito equivalente RLCG ad una sola frequenza	56
4.2	ADS: step per ottenere il file touchstone dei parametri S e la broadband netlist	58
4.3	Q3D: step per ottenere il file touchstone dei parametri S, la broadband netlist e la netlist RLCG definita ad una frequenza	67
4.4	Matlab: generazione delle broadband netlist	71
4.5	Vector fitting - teoria	71
4.5.1	Standard vector fitting	71
4.5.2	Relaxed vector fitting	75
4.5.3	Fast vector fitting	76
4.5.4	Vector fitting - Singular value decomposition	77
5	Cosimulazione: package del convertitore Buck	79
5.1	Buck: brevi cenni di teoria	79
5.2	Simulazione elettromagnetica del package	82
5.3	Netlist equivalenti del package	82
5.4	Simulazioni preliminari	84
5.4.1	Risultati	84
5.5	Cosimulazioni	87
5.5.1	Risultati	91
6	Conclusioni	97
6.1	Prospettive future	97
	Bibliografia	99

Elenco delle figure

1.1	Power-converter in GaN.	3
1.2	Zoom più dettagliato sul power loop di Figura 1.1.	3
1.3	Generico dispositivo a n porte.	6
2.1	Rappresentazione di un generico tetraedro che costituisce la mesh FEM in 3D di una generica struttura circuitale.	12
2.2	Rappresentazione TLine.	14
2.3	Topologia circuitale del modello RLCG a parametri concentrati (Lumped RLCG) utilizzata per il calcolo dei parametri S.	14
2.4	Singolo conduttore con un solo source e un solo sink.	15
2.5	Circuito equivalente RLCG del singolo conduttore avente un solo source e un solo sink.	15
2.6	Singolo conduttore con 2 sources e un solo sink.	16
2.7	Circuito equivalente RLCG del singolo conduttore avente 2 sources e un solo sink.	17
2.8	Esempio di due conduttori separati aventi ciascuno un solo source e un solo sink.	18
2.9	Circuito equivalente RLCG dei due conduttori separati aventi ciascuno un solo source e un solo sink.	19
2.10	Generica topologia circuitale detta T-model.	20
2.11	Shunt subcircuit del T-model nel caso di 3 conduttori distinti.	21
2.12	Esempio di singolo conduttore avente 2 sources e un sink con focus sulle resistenze.	23
2.13	Series subcircuit del T-model equivalente ($R_{21} = R_{12}$).	24
2.14	Generica topologia circuitale detta Ladder-model, avente N celle.	25
2.15	Rappresentazione del meshing sulla superficie conduttiva.	27
2.16	Rappresentazione delle correnti che fluiscono attraverso i bordi delle celle nella mesh. fonte guida di ADS	28
2.17	Modello circuitale LC di una singola cella della mesh.	30
2.18	Esempio di dispositivo che irradia solamente a livello superficiale.	32
2.19	Sistema di coordinate utilizzato.	35

2.20	Rappresentazione di una singola cella cubica.	38
2.21	Esempio di griglia in cui è suddivisa la sfera blu nell'immagine.	38
2.22	Sistema di riferimento in far-field.	39
3.1	Grafici che modellizzano l'andamento in frequenza di R ed L parassite.	44
3.2	Distribuzione di carica all'interno della sezione S di un generico conduttore in DC region e in AC region.	45
3.3	Circuito equivalente per stimare R ed L parassite in transition region. Questo modello circuitale ha un correct limiting behavior.	47
3.4	Layout dell'induttore simulato su EMPro.	52
3.5	Onda quadra a 2MHz in ingresso alla porta dell'induttore.	53
3.6	Confronto di S_{11} della portante a 2MHz con quello della singola sinusoide a 2MHz da sola alla porta dell'induttore su EMPro.	54
3.7	Confronto di S_{11} dell'armonica a 6MHz con quello della singola sinusoide a 6MHz da sola alla porta dell'induttore su EMPro.	54
3.8	Confronto di S_{11} dell'armonica a 10MHz con quello della singola sinusoide a 10MHz da sola alla porta dell'induttore su EMPro.	54
3.9	Confronto di S_{11} dell'armonica a 14MHz con quello della singola sinusoide a 14MHz da sola alla porta dell'induttore su EMPro.	55
4.1	Blocchettino nport, esempio con 4 porte.	57
4.2	Testbench costruito su ADS per generare il file touchstone dei parametri S: Layout dell'induttore + costrutti dedicati.	59
4.3	Opzioni standard del Broadband SPICE Model Generator su ADS.	59
4.4	Opzioni di input avanzate del Broadband SPICE Model Generator su ADS.	60
4.5	Impostazioni per la generazione del circuito RLCG ad una certa frequenza su Q3D.	68
4.6	Opzioni avanzate del Broadband Generator su Q3D.	69
5.1	Schematico teorico di un Buck sincrono con $V_{in} = 5.5V$	80
5.2	Andamento teorico delle tensioni V_{in} e V_{out} del Buck sincrono.	81
5.3	Andamento teorico della tensione V_x del Buck sincrono: onda quadra.	81
5.4	Esempio generico di fili di bonding in un package.	82
5.5	Cadence: testbench costruito per l'analisi AC di R ed L parassite riguardante il filo 5-9 - BROADBAND netlist completa a 9 porte.	84
5.6	Risultati analisi AC in SPICE: confronto della R tra broadband e RLCG netlist a 2MHz - filo 5-9.	85
5.7	Risultati analisi AC in SPICE: confronto della L tra broadband e RLCG netlist a 2MHz - filo 5-9.	85
5.8	Filo 5-9: comportamento in frequenza della R parassita su Q3D.	86
5.9	Filo 5-9: comportamento in frequenza della L parassita su Q3D.	86

5.10	Esempio simbolico di schematico dove viene eseguita la cosimulazione tra due sotto circuiti definiti in formati diversi.	88
5.11	Progetto realistico su Cadence fornito dall'azienda: zoom sulla nport a 9 porte che modella i parametri S dei fili Buck che sono inglobati nel circuito elettrico.	89
5.12	Progetto realistico su Cadence fornito dall'azienda: zoom sulla broadband netlist a 9 porte del Buck che è inglobata nel circuito elettrico.	89
5.13	Progetto realistico su Cadence fornito dall'azienda: zoom sulla netlist RLCG a 2MHz del Buck avente 9 porte che è inglobata nel circuito elettrico.	90
5.14	Modello puramente ideale dei fili di bonding: corto circuiti.	91
5.15	Simulazione transient con l'nport. Tensione di uscita	92
5.16	Andamento tensione VLXpin (pin 9) delle differenti versioni circuitali: zoom a steady-state.	93
5.17	VLXpin (pin 9), zoom ulteriore su periodo alto dell'onda quadra per evidenziare il ringing alla fine del fronte di salita.	93
5.18	VLXpin (pin 9), zoom sul ringing dell'immagine precedente.	94
5.19	Inviluppo dello spettro di un'onda trapezoidale.	95
5.20	Broadband netlist del buck: tensioni sulle due estremità del filo 5-9 (pin 5 e pin 9) a confronto.	96

Capitolo 1

Simulazioni elettromagnetiche

1.1 L'importanza delle simulazioni elettromagnetiche

In fase di progetto, gli schematici di una determinata struttura circuitale indicano quale è la sua funzione di trasferimento, quanto deve valere la tensione in un certo nodo, ma non rendono l'idea di quale sia la geometria del layout (qualunque sia la tecnologia circuitale), e di come i vari componenti interagiscono con i campi elettromagnetici che si generano nella realtà, durante il funzionamento del circuito stesso. Diventa fondamentale associare allo schematico circuitale anche un altro modello che descriva come le non idealità influiscono sul comportamento del circuito nella realtà. Queste non idealità infatti possono provocare comportamenti indesiderati. Se, non viene tenuto conto fin dalle fasi iniziali di progetto di questa problematica, accadrà che questi effetti indesiderati saranno conosciuti solo quando verrà testato il prototipo finale. Negli step intermedi di progetto, non si potranno prendere eventuali contromisure atte proprio ad evitare questi effetti non voluti. Tenere conto fin dall'inizio di queste non idealità aiuta i progettisti a ridurre i tempi di produzione e i tempi di progetto [1]. Queste non idealità vengono modellizzate dai cosiddetti **parametri parassiti** di un certo circuito. Per estrarli bisogna effettuare la **simulazione elettromagnetica**. La simulazione elettromagnetica permette di trovare quali sono i parametri parassiti che, **dal punto di vista elettrico**, influiscono sulle prestazioni del circuito. Se non si tenesse conto sin da subito dei parassiti il rischio sarebbe quello di dover ogni volta produrre nuovi prototipi, quindi ci sarebbe un aumento dei costi perchè si produrrebbero prototipi non correttamente funzionanti e in generale si finirebbe quindi in una sorta di loop

di progetto per via del fatto che ci si accorgerebbe soltanto a fine fabbricazione se un certo prototipo non rispetta le specifiche di affidabilità e performance [1]. Si innescherebbe un circolo vizioso caratterizzato da una sorta di trial and error. Gli obiettivi di affidabilità sono il rispetto dei limiti riferiti alla **compabilità elettromagnetica (EMC)**, all'**interferenza elettromagnetica (EMI)**, la **signal integrity** [2] che vengono regolamentati da un ente denominato **CISPR**, che stabilisce quali sono gli standard da rispettare [3].

In fase di progetto vi è un continuo confronto tra aziende clienti e aziende produttrici riguardante il tema dei parassiti. L'obiettivo è tenerli sotto controllo durante tutti gli step di progetto [1].

L'analisi dei parassiti sta diventando sempre più necessaria anche nel caso di tecnologie emergenti come quelle in **GaN** e **SiC**. Transistor in GaN per esempio, risultano essere molto sensibili ai parassiti e per questo si rende necessario fare una ottimizzazione del layout del PCB su cui andranno inseriti, per non avere un peggioramento delle prestazioni. Per esempio, nel caso di power-converters (Figura 1.1) realizzati usando transistor GaN, sempre su PCB, occorre minimizzare l'induttanza parassita [4]. Questa induttanza parassita, provoca del **ringing** (cioè delle oscillazioni indesiderate) delle tensioni VGS e VDS dei transistor durante le commutazioni. Queste oscillazioni, a loro volta provocano delle sovratensioni non volute. Occorre precisare che l'induttanza di loop in Figura 1.1 è la somma delle induttanze che fanno parte del power loop evidenziato in giallo. Tra queste induttanze si conta anche l'induttanza del PCB e l'induttanza del package [4]. Sempre a proposito di questo esempio, diventa importante anche minimizzare la capacità del loop di gate, in modo da evitare problemi di crosstalk [4].

Nel circuito di Figura 1.1, il transistor $Q2$ è soggetto al problema del ringing appena menzionato. Si ha che la frequenza di ringing della tensione tra drain e source del low-side transistor $Q2$ è definita come:

$$f_{ringing} = \frac{1}{2\pi\sqrt{L_{Loop}C_{oss}}}; \quad (1.1)$$

e l'obiettivo è quello di minimizzarla. L_{Loop} rappresenta l'induttanza di loop mentre C_{oss} rappresenta la capacità di uscita del low-side transistor $Q2$ di Figura 1.1 e 1.2. L'aumentare della frequenza di switching, in generale rende necessario tenere sotto osservazione capacità e induttanze parassite [5].

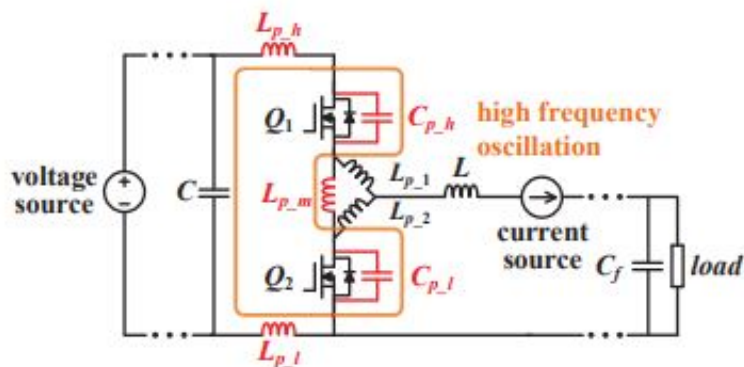


Figura 1.1: Power-converter in GaN.

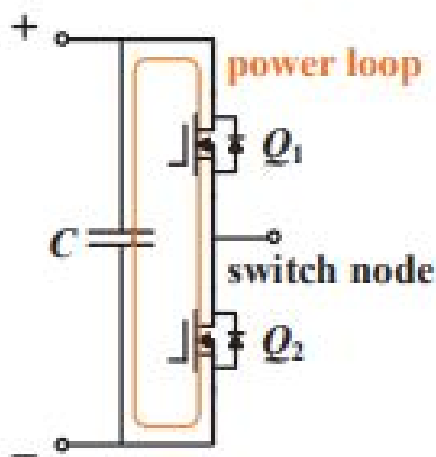


Figura 1.2: Zoom più dettagliato sul power loop di Figura 1.1.

Invece quello che accade nei circuiti costruiti con tecnologia in SiC è che l'induttanza di loop va tenuta sotto controllo soprattutto quando si lavora ad alte tensioni e ad alte frequenze: questa induttanza, col crescere della frequenza, genera degli spike di tensione che possono mandare in breakdown il dispositivo. Per questo motivo in fase di progetto diventa fondamentale ridurre questa induttanza, in modo da abbassare questi spike di tensione [6]. Anche nel caso di power converters in SiC, le alte frequenze di switching al quale lavorano influenzano il funzionamento di questi circuiti per via del fatto che inducono oscillazioni indesiderate di tensioni e correnti. Queste oscillazioni influiscono dal punto di vista dell'interferenza

elettromagnetica, del rumore, provocando anche perdite di potenza aggiuntive. Tutte queste problematiche impongono un limite sulla frequenza di switching che realmente è impostabile per il circuito [7].

Gli esempi appena citati riguardanti le tecnologie in GaN e SiC sono utili per avere un'idea di quanto al giorno d'oggi i parametri parassiti influiscano sulle prestazioni di circuiti in tecnologie sempre più moderne, che garantiscono una sempre maggiore efficienza, anche in termini di potenza. Con l'aumento delle frequenze i parassiti non sono più trascurabili.

Una soluzione che permetta di non preoccuparsi dell'effetto dei parassiti potrebbe essere quella di fare i progetti stando dentro dei margini in modo tale da raggiungere le performance richieste potendo trascurare tutte le non idealità in gioco. Fare ciò non risulta conveniente dal punto di vista economico perchè produrre dispositivi che permettano di considerare dei margini dal punto di vista delle specifiche richiede maggiori costi. Risulta quindi economicamente più sostenibile, per un'azienda, seguire delle specifiche più restrittive di quanto basta per avere comunque le performance richieste per un dato progetto per poter rimanere competitiva sul mercato.

1.2 Flusso per la cosimulazione

In generale, per testare un dato circuito, vi sono degli step che vengono seguiti dai progettisti:

1. **Progettazione dello schematico** del circuito;
2. **Design del layout**;
3. **Simulazione del circuito elettrico** per accertarsi che rispetti le specifiche iniziali;
4. **Simulazione elettromagnetica** e in seguito **estrazione dei parassiti** che riguardano le interconnessioni, che sono modellizzati dai parametri S o da circuiti equivalenti (che saranno visti in seguito in questa trattazione);
5. **Cosimulazione** del circuito elettrico assieme ai parassiti [2].

Grazie all'utilizzo di diversi CAD tool è stato possibile seguire gli step appena elencati su dei circuiti forniti dall'azienda **STMicroelectronics**. Questo studio di Tesi è stato incentrato in particolare **dallo step 4 in poi**, effettuando cioè la **simulazione elettromagnetica** e l'estrazione dei parametri parassiti di un dato circuito.

La **cosimulazione** permette di integrare in un unico tool il circuito puramente elettrico ideale, con i parassiti che sia le interconnessioni che la tecnologia in uso

introducono. Dal risultato della cosimulazione si può quindi osservare come tutte queste non idealità influiscono sulle prestazioni.

Il fine principale tra uno step e l'altro rimane quello di poter tenere sotto controllo e ridurre gli effetti dovuti alle non idealità introdotte dai parassiti per poter apportare, dove necessario, delle modifiche sul layout. Come già accennato, diventa quindi conveniente effettuare diverse tipologie di analisi come ad esempio: signal integrity, electromagnetic compatibility (EMC), power integrity, thermal analysis, DRC (design rule checking) e tante altre. Lo scopo della **signal integrity** è quello di studiare e minimizzare rumori, distorsioni e perdite all'interno dei circuiti che tendono a degradare la qualità di un certo segnale elettrico.

L'**electromagnetic compatibility (EMC)** si occupa di studiare tutti gli aspetti relativi ai disturbi che un certo dispositivo può generare nei confronti di altri sistemi e a come ridurli. Inoltre l'EMC tiene anche conto anche della capacità di un certo circuito di tollerare i disturbi provenienti da altri dispositivi. Come scritto in precedenza, il tutto è anche regolamentato da dei vincoli introdotti dal CISPR [3]. Invece la **power integrity analysis (detta anche power supply analysis)** permette di studiare il comportamento delle reti di distribuzione dell'alimentazione e capire, per esempio se ci sono zone nel layout in cui si hanno cadute di tensione non volute.

Il **DRC** permette di controllare se sono state rispettate le regole di progetto del layout, come ad esempio la minima distanza tra due componenti. Il DRC permette anche di analizzare problemi relativi alla compatibilità elettromagnetica e all'interferenza elettromagnetica.

La **thermal analysis** invece permette di capire in maniera dettagliata come un dato circuito dissipa il calore generato durante il suo funzionamento.

Effettuando il più possibile queste ulteriori analisi si possono prendere in considerazione dei cambiamenti, degli accorgimenti in corso di progetto, in modo da poter apportare delle modifiche in modo consapevole evitando così lo spreco di risorse economiche per la produzione di repliche di prototipo di bassa qualità, aventi scarse prestazioni. I vantaggi sono anche dal punto di vista del time to market, ovvero del tempo di mercato: si evitano in questo modo ritardi sulla commercializzazione del dispositivo [1] e si diminuisce il ciclo di progetto.

Prima di performare le analisi appena menzionate, diventa importante, sempre con l'ausilio di differenti tool, eseguire la **simulazione elettromagnetica**. Gli step da seguire per effettuarla sono:

1. **Design del modello fisico:** cioè definizione della geometria del layout, assegnazione dei materiali e delle loro proprietà fisiche;
2. **Setup delle impostazioni della simulazione elettromagnetica:** cioè definire un frequency plan (start frequency, stop frequency, il tipo di step), le porte interessate, le condizioni al contorno, e altre impostazioni;

3. Esecuzione della simulazione elettromagnetica;
4. Post processing dei risultati: estrazione dei parametri **S**, ed altri parametri di interesse [2].

1.3 Parametri S

Uno dei principali risultati che si ottengono dalla simulazione elettromagnetica di un dato circuito sono i suoi parametri S. Essi modellizzano il comportamento dei parametri parassiti al variare della frequenza. Occorre definire brevemente, in questo capitolo introduttivo sulle simulazioni elettromagnetiche, le principali nozioni teoriche riguardanti i parametri S (scattering, di diffusione): qualsiasi circuito a n porte può essere modellizzato tramite i suoi parametri S come riportato in Figura 1.3: si ha che a_i modella l'onda di potenza progressiva (entrante nella i -esima porta), mentre b_i rappresenta l'onda di potenza regressiva (uscite dalla i -esima porta). Dentro i parametri S è inglobato il comportamento del circuito in termini di tensioni e correnti presenti in ciascuna porta. Per semplicità ciascuna porta ha riferimento a ground e nella maggior parte dei casi per ciascuna porta si assegna un'impedenza di riferimento $R_0 = 50\Omega$. Ogni parametro S_{ij} è funzione della frequenza, cioè $S_{ij} = S_{ij}(f)$.

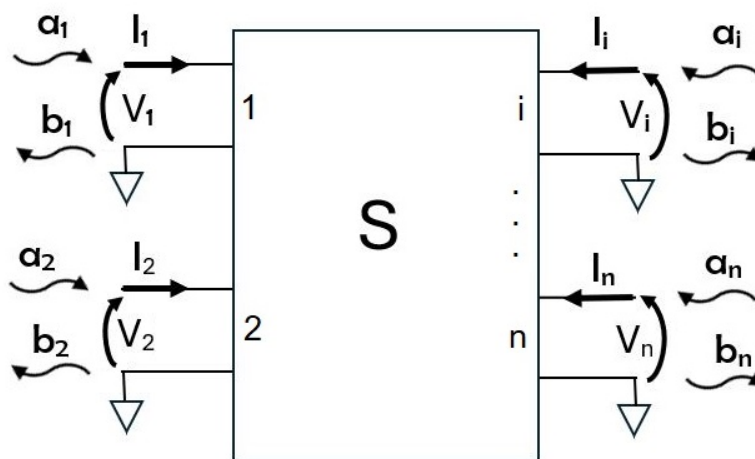


Figura 1.3: Generico dispositivo a n porte.

Le seguenti equazioni mostrano il legame tra la matrice S, le onde di potenza e tensioni/correnti in ciascuna porta:

$$[b] = [S] \cdot [a] \rightarrow \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} & \dots & S_{1n} \\ S_{21} & S_{22} & \dots & S_{2n} \\ \vdots & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \vdots \\ S_{n1} & \dots & \dots & S_{nn} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ \vdots \\ a_n \end{pmatrix} \quad (1.2)$$

$$\begin{aligned} b_1 &= S_{11}a_1 + S_{12}a_2 + \dots + S_{1n}a_n \\ b_2 &= S_{21}a_1 + S_{22}a_2 + \dots + S_{2n}a_n \\ &\vdots \\ &\vdots \\ b_n &= S_{n1}a_1 + S_{n2}a_2 + \dots + S_{nn}a_n \end{aligned} \quad (1.3)$$

$$S_{ij} = \left. \frac{b_i}{a_j} \right|_{a_i=0}; \quad S_{ii} = \left. \frac{b_i}{a_i} \right|_{a_j=0}; \quad \text{con } i \neq j \quad (1.4)$$

$$a_k = \frac{V_k + R_k I_k}{2\sqrt{R_k}}; \quad b_k = \frac{V_k - R_k I_k}{2\sqrt{R_k}} \quad (1.5)$$

$$V_k = (a_k + b_k)\sqrt{R_k}; \quad I_k = \frac{a_k - b_k}{\sqrt{R_k}} \quad (1.6)$$

dove R_k rappresenta l'impedenza di riferimento della porta k-esima che solitamente si pone $R_k = R_0 = 50\Omega$. Nel calcolo di un certo S_{ij} , le porte non interessate sono chiuse sulla loro impedenza di riferimento per avere carico adattato, in modo da avere riflessioni nulle, cioè si impone così $a_i = 0$ per tutte le altre porte.

I parametri S sono stati utilizzati in questo studio di tesi, come **la quantità principale che identifica e caratterizza un dato circuito, dal punto di vista elettrico, a seguito della sua simulazione elettromagnetica.**

Per una data struttura circuitale, ci si è poi serviti dei suoi parametri S per ricavarsi un modello circuitale equivalente che modellizzi il comportamento nel dominio del tempo: per la matrice S, al variare della frequenza, è possibile definire un circuito equivalente corrispondente. Questo fatto è modellizzato dalla cosiddetta **broadband netlist**. A sua volta, la broadband netlist, ad una certa frequenza, è modellizzata da un **circuito RLCG equivalente**.

Lo scopo finale della Tesi è proprio quello di confrontare e capire le differenze tra il modello RLCG (valido ad una sola frequenza) con la

broadband netlist (definita su tutto il range di frequenza di interesse e non solo per una frequenza) e con i suoi parametri **S**. Questi tre formati modellizzano il comportamento dei parassiti di un certo circuito.

Nei capitoli successivi quanto appena accennato sarà dettagliato maggiormente.

1.4 Passivity, causality, stability

In questo studio di Tesi, i circuiti e le relative netlist equivalenti devono essere contemporaneamente **passive**, **causali** e **stabili**. Se queste tre condizioni non sono soddisfatte, ci potrebbero essere problemi di non convergenza durante la cosimulazione in transient delle netlist equivalenti dei circuiti esaminati. Oppure pur convergendo, la cosimulazione potrebbe dare risultati sbagliati. Nel seguito sono riportate le definizioni di **passivity**, **causality** e **stability**.

Passivity

In un circuito passivo (e nella realtà, con perdite) a n porte, la somma delle potenze entranti in tutte le porte è maggiore della somma delle potenze riflesse in tutte le porte. Vale a dire:

$$\sum_{i=1}^N |a_i|^2 > \sum_{i=1}^N |b_i|^2 \quad (1.7)$$

Quindi, la potenza netta totale P sarà definita come:

$$P = \sum_{i=1}^N P_i = \frac{1}{2} \sum_{i=1}^N (|a_i|^2 - |b_i|^2) > 0 \quad (1.8)$$

dove a_i e b_i (già definite nella Sezione 1.3) rappresentano rispettivamente l'onda di potenza progressiva e regressiva della i -esima porta. Invece P_i rappresenta la potenza netta entrante nella i -esima porta del circuito. La potenza totale netta P invece è la somma delle potenze nette di ciascuna porta. Essendo nel caso di dispositivi con perdite, si ha $P > 0$. In caso di dispositivi senza perdite, $P = 0$. **Questo significa che non si può creare energia dal nulla.** L'equazione (1.8) può essere riscritta anche in funzione dei vettori delle onde di potenza $[a]$ e $[b]$:

$$P = \frac{1}{2} \left([a^*]^T [a] - [b^*]^T [b] \right) > 0 \quad (1.9)$$

con $[a]$ e $[b]$ già definiti in (1.2).

Sapendo che $[b] = [S][a]$ e $[b^*]^T = [a^*]^T [S^*]^T$, si avrà che la (1.9) può essere riscritta come:

$$P = \frac{1}{2} \left([a^*]^T [a] - [a^*]^T [S^*]^T [S] [a] \right) > 0 \quad (1.10)$$

se raccogliamo a fattor comune $[a^*]^T$ e $[a]$ otteniamo:

$$P = \frac{1}{2} [a^*]^T \left(I - [S^*]^T [S] \right) [a] > 0 \quad (1.11)$$

Dal quale scaturisce la condizione finale che permette di valutare la passività di un circuito utilizzando i suoi parametri S. **Quindi un circuito è passivo se:**

$$I - [S^*]^T [S] > 0 \quad (1.12)$$

dove $[S^*]^T$ rappresenta la trasposta del complesso coniugato della matrice S. I invece è la matrice identità i cui elementi della diagonale valgono tutti 1 e gli altri elementi hanno valore 0.

Causality [8] [9]

Per spiegare il concetto relativo alla causality si parte da un principio molto semplice: un effetto NON può precedere la sua causa. Un circuito avente risposta all'impulso $h(t)$ è causale se:

$$h(t) = 0 \quad \text{per } t \leq 0 \quad (1.13)$$

dove questa relazione è definita nel dominio del tempo.

Se si passa nel dominio della frequenza si avrà che la risposta all'impulso sarà definita come:

$$H(j\omega) = U(\omega) + jV(\omega) \quad (1.14)$$

Analogamente a quanto illustrato nel dominio del tempo, un sistema avente risposta all'impulso o risposta in frequenza $H(j\omega)$ è causale se:

$$U(\omega) = \frac{1}{\pi} pv \int_{-\infty}^{+\infty} \frac{V(\omega')}{\omega - \omega'} d\omega' \quad (1.15)$$

$$V(\omega) = -\frac{1}{\pi} pv \int_{-\infty}^{+\infty} \frac{U(\omega')}{\omega - \omega'} d\omega' \quad (1.16)$$

dove è bene precisare che:

$$pv \int_{-\infty}^{+\infty} = \lim_{\epsilon \rightarrow 0^+} \left[\int_{-\infty}^{\omega - \epsilon} + \int_{\omega + \epsilon}^{+\infty} \right] \quad (1.17)$$

Le (1.15) e la (1.16) sono dette **relazioni di dispersione Kramers-Kronig**. In queste due relazioni si ha che la parte reale di $H(j\omega)$ **dipende** dalla parte immaginaria di $H(j\omega)$ e viceversa. Queste sono condizioni necessarie e sufficienti per dire che un certo sistema è causale. **Questo discorso vale anche per i parametri S di un certo circuito: se la matrice S è tale che tutti i suoi elementi S_{ij} , che sono valori complessi, rispettano le condizioni (1.15) e (1.16) allora il circuito è causale.**

Stability

Per valutare la stabilità di un certo circuito, si possono analizzare i suoi parametri S. In un circuito stabile, **tutti i poli di tutti gli S_{ij} devono avere parte reale negativa**. In questo studio, la netlist equivalente di un circuito, avrà l' S_{ij} che viene approssimato tramite il metodo del **vector fitting** come:

$$S_{ij}(s) \cong \sum_{n=1}^N \frac{c_n}{s - a_n} + d + sh \quad (1.18)$$

dove gli a_n sono proprio i poli.

Tutta la teoria riguardante il metodo del vector fitting è illustrata nella sezione 4.5.

Quindi dai parametri S si riesce a capire se le condizioni di passivity, causality e stability sono soddisfatte.

Capitolo 2

Simulatori utilizzati

Per effettuare le simulazioni elettromagnetiche e le cosimulazioni sono stati utilizzati i simulatori prodotti dalle aziende ANSYS, Keysight Technologies e Cadence Design Systems che risultano essere i più diffusi e disponibili sul mercato [1]. Per quanto riguarda ANSYS si è utilizzato il simulatore interno **Q3D Extractor**, mentre per quanto riguarda Keysight Technologies si sono utilizzati i simulatori **ADS** e **EMPro**. Si è anche utilizzato, il simulatore **SPICE**. Nel seguito sono descritti nel dettaglio, uno per uno, i vari simulatori appena menzionati e le loro principali funzionalità, utili per i fini di questo studio sui parassiti.

2.1 Q3D

Q3D Extractor è un simulatore quasi-statico 3D [10]. 3D significa che nei calcoli sono considerate le interazioni elettromagnetiche del circuito in tutte e tre le direzioni spaziali [1]. Per quasi-statico invece si intende che l'equazione $l < \lambda/10$ deve essere soddisfatta, dove l è massima distanza tra due punti della geometria del circuito e $\lambda = c/f$ è la lunghezza d'onda riferita alla frequenza f di simulazione [11]. Q3D Extractor è interno al software Ansys Electronics Desktop ed è utilizzato in questo studio per eseguire la simulazione elettromagnetica ed è adatto per l'analisi di circuiti PCB, package IC, e interconnessioni [12]. **Risulta essere utile per le finalità di questa Tesi perchè permette di estrarre 3 tipi di modelli circuitali equivalenti utilizzati che modellizzano il circuito originario che si intende analizzare:**

1. **Modello equivalente in termini dei suoi parametri S**, definito su tutto il range di frequenze;
2. **Broadband netlist equivalente**, definita anch'essa su tutto il range di frequenze;

3. **Circuito equivalente RLCG**, valido però ad una sola frequenza scelta.

Q3D lavora combinando due metodi di calcolo per ricavare le soluzioni relative al campo: **Finite Element Method (FEM)** e **Method of Moments (MoM)**. Nel seguito sono descritti i due metodi appena menzionati.

FEM [10]

In generale Q3D costruisce le mesh dividendo la struttura 3D in elementi a forma di tetraedro: ogni tetraedro rappresenta una regione più piccola alla quale è assegnata una funzione locale che rappresenta il campo. Il simulatore definisce i nodi della mesh assegnando sul perimetro delle facce del tetraedro (al centro di ogni lato) e sui suoi vertici, una certa quantità riguardante il campo (ad esempio il potenziale elettrico). Il valore del campo all'interno del volume di ciascun tetraedro è ottenuto tramite interpolazione dei nodi appena citati. Quanto appena descritto è riassunto nella seguente immagine: Q3D utilizza proprio il metodo FEM per calcolare le resistenze e induttanze parassite in DC region.

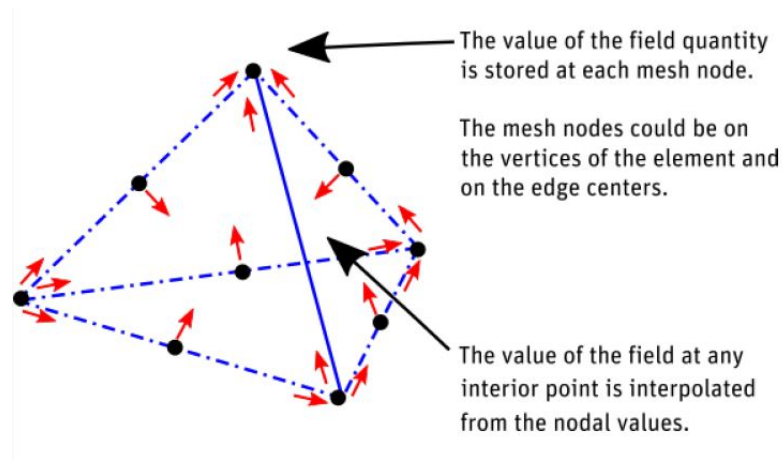


Figura 2.1: Rappresentazione di un generico tetraedro che costituisce la mesh FEM in 3D di una generica struttura circuitale.

Metodo dei momenti [10]

MoM è utilizzato per calcolare capacità e induttanze parassite. Per problemi riguardanti le capacità le mesh sono generate in 2D perchè la quantità di interesse che riguarda il campo è la densità superficiale di carica di un conduttore o di un'interfaccia di un dielettrico. Per questo motivo la superficie interessata viene suddivisa in tanti triangolini. Infine per le induttanze parassite in AC region la geometria è suddivisa in triangoli per il fatto che le quantità relative al campo sono

le correnti superficiali.

Più avanti nella trattazione verrà spiegato nel dettaglio che cosa si intende per resistenze e induttanze in DC e in AC region.

Nella Sezione 2.2.1 verrà spiegato con più dettaglio il metodo dei momenti.

2.1.1 Calcolo dei parametri S con Q3D [10]

Q3D effettua il calcolo dei parametri S secondo due approcci alternativi: il primo è **l'approccio del modello equivalente di linea di trasmissione distribuita**, mentre il secondo è il **metodo del lumped RLCG model (modello RLCG a parametri concentrati)**.

Nell'approccio relativo al modello equivalente di linea di trasmissione distribuita, il simulatore assume che la struttura utilizzata rappresenti fili (o interconnessioni) paralleli di lunghezza simile tra di loro. Questa modalità prevede che i parametri S siano calcolati a partire dall'equazione dei telegrafisti e che la singola linea di trasmissione a n conduttori sia modellizzata usando un set di n linee di trasmissione separate. Come appena scritto, si parte dalle equazioni dei telegrafisti sottostanti:

$$\frac{\partial V(z, t)}{\partial z} = -Ri(z, t) - L \left(\frac{\partial I(z, t)}{\partial t} \right) \quad (2.1)$$

$$\frac{\partial I(z, t)}{\partial z} = -Gv(z, t) - C \left(\frac{\partial V(z, t)}{\partial t} \right) \quad (2.2)$$

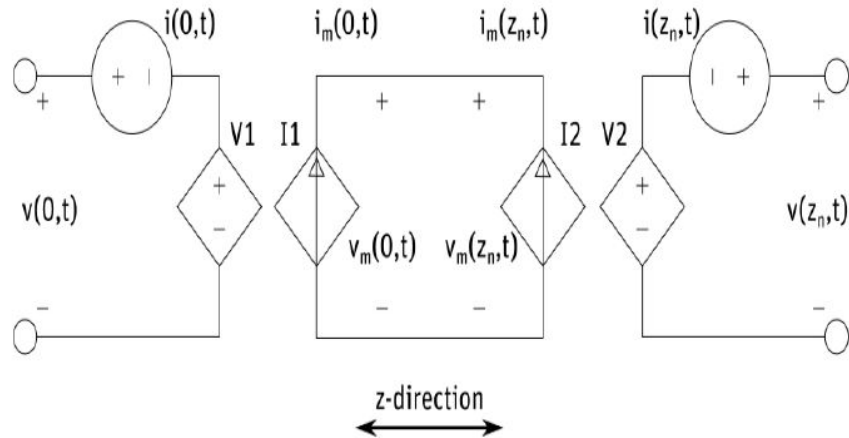


Figura 2.2: Rappresentazione TLine.

In caso i fili/interconnessioni fossero di lunghezza differente (per esempio fili lunghi 50cm accanto a fili di 5mm), i risultati ottenuti potrebbero essere non aderenti alla realtà, e a quel punto diventa conveniente utilizzare il secondo metodo che prende il nome di lumped RLCG model approach, ovvero modello RLCG a parametri concentrati.

Questa seconda metodologia risulta anche utile per strutture 3D arbitrarie. Il modello lumped RLCG dal quale vengono calcolati i parametri S è composto nel modo seguente:

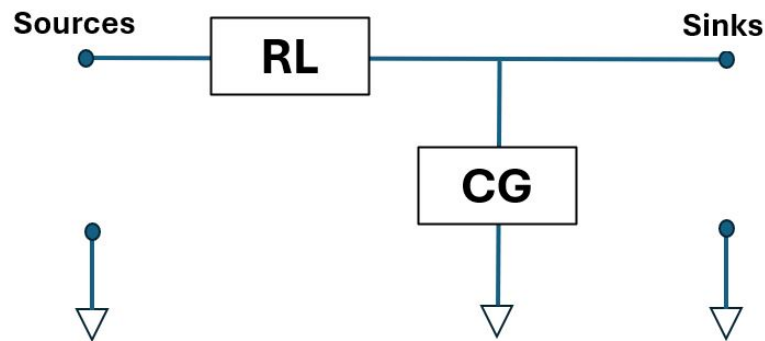


Figura 2.3: Topologia circuitale del modello RLCG a parametri concentrati (Lumped RLCG) utilizzata per il calcolo dei parametri S.

Nelle sezioni 3.1.2, 3.1.3, 3.1.4 è descritto come vengono calcolati i vari parametri R, L, C, G che sono dipendenti dalla frequenza per via dello skin-effect che sarà argomento del prossimo capitolo.

2.1.2 Calcolo del circuito RLCG ad una certa frequenza con Q3D [10]

Q3D, come detto, calcola anche il circuito equivalente RLCG ad una determinata frequenza. **In questa sottosezione verrà chiarito il concetto di circuito RLCG.** Vediamo alcune casistiche di strutture più semplici e come vengono costruite le relative topologie circuitali equivalenti per ciascuna di esse. Partiamo dal caso più semplice possibile, ovvero singolo conduttore avente un solo source e un solo sink, come mostrato sotto:

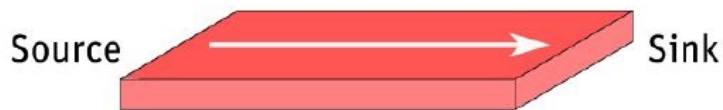


Figura 2.4: Singolo conduttore con un solo source e un solo sink.

I parametri circuitali RLCG rappresentanti questo conduttore sono semplicemente $R = R_{11}$, $L = L_{11}$, $C = C_{11}$, $G = G_{11}$. Il circuito equivalente RLCG costruito da Q3D avrà la seguente topologia:

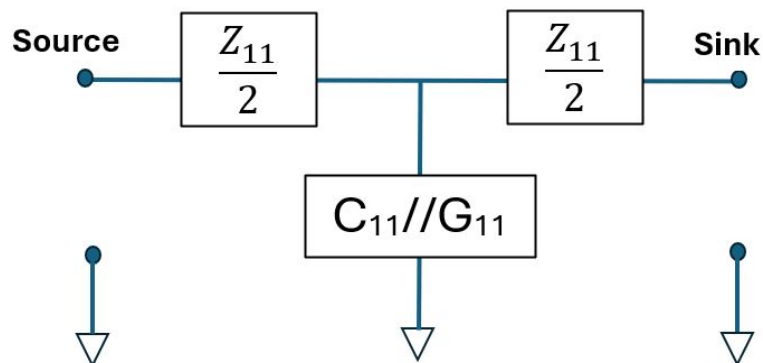


Figura 2.5: Circuito equivalente RLCG del singolo conduttore avente un solo source e un solo sink.

dove:

$$Z_{11} = R_{11} + j\omega L_{11}; \quad \omega = 2\pi f \quad (2.3)$$

quindi l'impedenza Z_{11} è suddivisa a metà lungo i terminali di source e sink e in parallelo il simulatore pone la capacità C_{11} e la conduttanza G_{11} . Nel caso di

materiale dielettrico con perdite la conduttanza G_{11} sarà non nulla, per via del fatto che delle piccole correnti di leakage scorrono in questi tipi di materiali. Invece nel caso di materiale privo di perdite $G_{11} = 0$. G_{11} è posta in parallelo alla capacità C_{11} .

Prendiamo in analisi ora il caso comprendente sempre un singolo conduttore, ma con 2 sources e un sink, come sotto riportato:

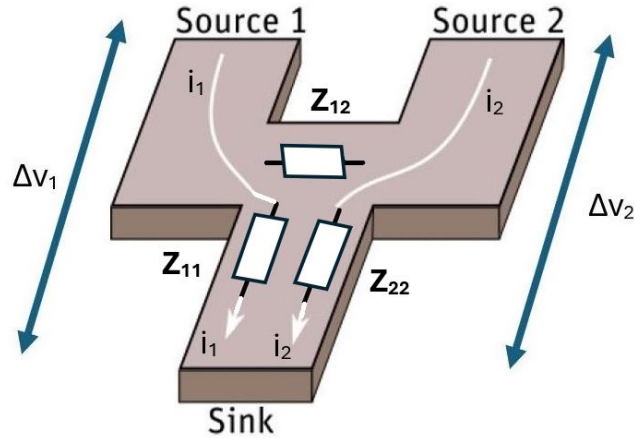


Figura 2.6: Singolo conduttore con 2 sources e un solo sink.

dove la relazione tra tensioni e correnti è data dalla seguente equazione matriciale:

$$\begin{pmatrix} \Delta v_1 \\ \Delta v_2 \end{pmatrix} = \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \end{pmatrix} \quad (2.4)$$

In questo caso i parametri circuitali sono rappresentati dalle matrici:

$$\begin{aligned} R &= \begin{pmatrix} R_{11} & R_{12} \\ R_{12} & R_{22} \end{pmatrix} & L &= \begin{pmatrix} L_{11} & L_{12} \\ L_{12} & L_{22} \end{pmatrix} \\ C &= \begin{pmatrix} C_{11} \end{pmatrix} & G &= \begin{pmatrix} G_{11} \end{pmatrix} \end{aligned} \quad (2.5)$$

dove il modello circuitale, che Q3D genera, sarà il seguente:

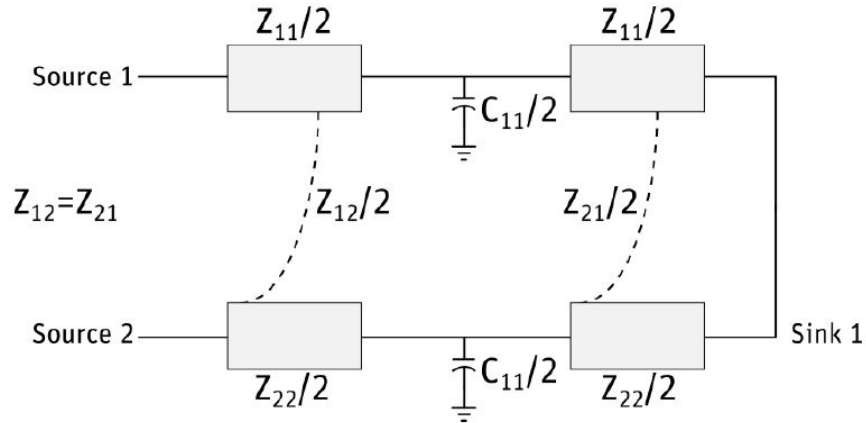


Figura 2.7: Circuito equivalente RLCG del singolo conduttore avente 2 sources e un solo sink.

nel quale:

$$\begin{aligned} Z_{11} &= R_{11} + j\omega L_{11}; \\ Z_{12} &= R_{12} + j\omega L_{12} = Z_{21}; \\ Z_{22} &= R_{22} + j\omega L_{22}; \end{aligned} \tag{2.6}$$

Anche qui, nel caso di materiale dielettrico con perdite dovute a correnti di leakage, G_{11} sarà non nullo: nel circuito equivalente della Figura 2.7, ciascuna capacità di valore $C_{11}/2$ sarà connessa in parallelo con una resistenza pari a $2/G_{11}$ (avente quindi conduttanza $G_{11}/2$). Z_{11} e Z_{22} rappresentano le self-impedances mentre Z_{12} è l'impedenza mutua tra i due sources del conduttore. Tensioni e correnti che scorrono nel conduttore sono proprio regolate dalle impedenze Z_{11} , Z_{22} e Z_{21} .

Un terzo caso da prendere in analisi è quello relativo a due conduttori separati (ciascuno avente un source e un sink), come mostrato nella figura seguente:

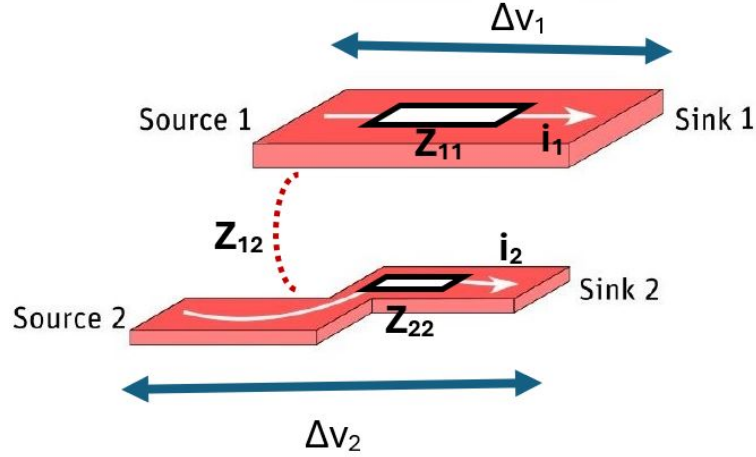


Figura 2.8: Esempio di due conduttori separati aventi ciascuno un solo source e un solo sink.

dove, anche in questo caso, la relazione tra tensioni e correnti è data dalla seguente equazione matriciale:

$$\begin{pmatrix} \Delta v_1 \\ \Delta v_2 \end{pmatrix} = \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \end{pmatrix} \quad (2.7)$$

Le matrici dei parametri circuitali sono:

$$\begin{aligned} R &= \begin{pmatrix} R_{11} & R_{12} \\ R_{12} & R_{22} \end{pmatrix} & L &= \begin{pmatrix} L_{11} & L_{12} \\ L_{12} & L_{22} \end{pmatrix} \\ C^M &= \begin{pmatrix} C_{11}^M & C_{12}^M \\ C_{12}^M & C_{22}^M \end{pmatrix} & G^M &= \begin{pmatrix} G_{11}^M & G_{12}^M \\ G_{12}^M & G_{22}^M \end{pmatrix} \end{aligned} \quad (2.8)$$

Il modello equivalente sarà il seguente:

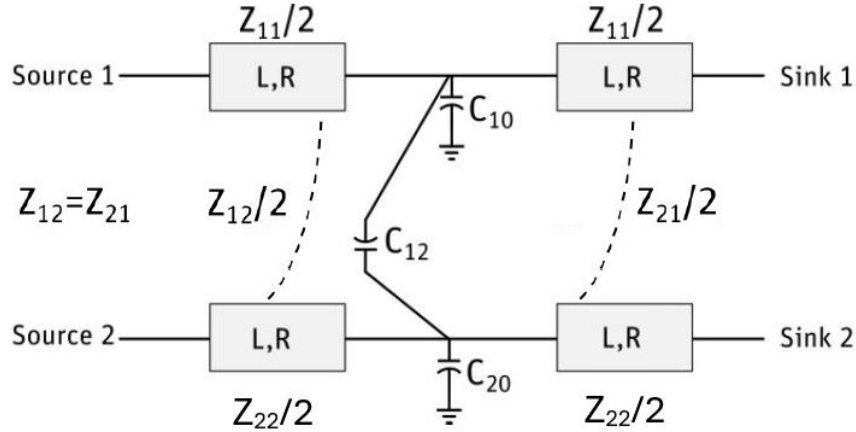


Figura 2.9: Circuito equivalente RLCG dei due conduttori separati aventi ciascuno un solo source e un solo sink.

dove Z_{11} , Z_{12} , Z_{21} , Z_{22} sono definite come in (2.6). Per le capacità si ha che:

$$\begin{aligned}
 C_{10} &= C_{11}^M + C_{12}^M; \\
 C_{20} &= C_{22}^M + C_{12}^M; \\
 C_{12} &= -C_{12}^M;
 \end{aligned} \tag{2.9}$$

La matrice delle capacità C^M è detta **Maxwell matrix**. C_{10} e C_{20} sono le capacità tra il primo conduttore e ground e tra il secondo conduttore e ground rispettivamente. C_{11}^M e C_{22}^M invece sono le self-capacitances di ciascun conduttore, mentre C_{12}^M è la mutual capacitance tra i due conduttori. C_{10} , C_{20} e C_{12} sono le componenti della cosiddetta **Spice Matrix** che è equivalente alla Maxwell matrix C^M . Infatti sia la Maxwell matrix che la Spice matrix descrivono in maniera univoca una certa struttura circuitale. Quindi Q3D calcola due tipi di matrici di capacità (**e anche di conduttanze, allo stesso modo**). Più avanti nella trattazione teorica verrà definita la relazione che in generale lega la Maxwell matrix con la Spice Matrix.

Come per i casi precedenti, in caso i materiali dielettrici siano con perdite dovute alle correnti di leakage, le capacità C_{10} , C_{20} e C_{12} saranno collegate in parallelo rispettivamente con i resistori $1/G_{10}$, $1/G_{20}$ e $1/G_{12}$ che sono modellizzati dalla

matrice delle conduttanze G^M definita in (2.8), in modo da avere:

$$\begin{aligned} G_{10} &= G_{11}^M + G_{12}^M; \\ G_{20} &= G_{22}^M + G_{12}^M; \\ G_{12} &= -G_{12}^M; \end{aligned} \tag{2.10}$$

Questi tre casi semplici appena analizzati sono ottenuti utilizzando la generica topologia circuitale di base che Q3D utilizza, detta T-model che è quella mostrata nell'immagine seguente:

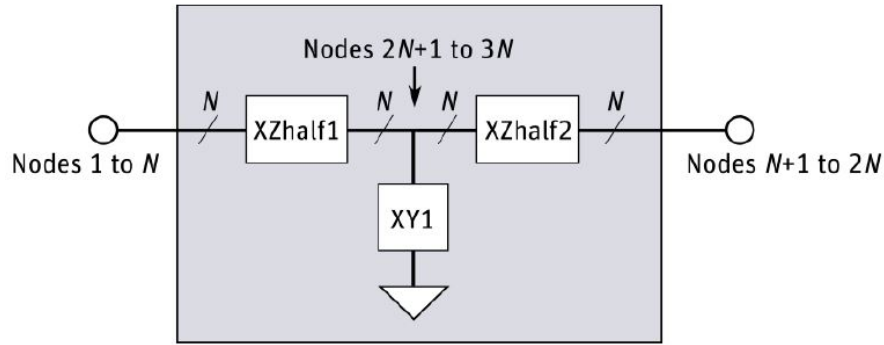


Figura 2.10: Generica topologia circuitale detta T-model.

La topologia circuitale di sopra, rappresenta l'elemento dal quale si costruiscono circuiti più complicati e grandi. Infatti, nei casi ancora più complessi vi sono diverse combinazioni di conduttori. Ma il principio è sempre quello descritto in Figura 2.10. Il circuito finale sarà formato da **series subcircuits** identici tra loro (XZhalf1 e XZhalf2 nella Figura 2.10) e dal **shunt subcircuit** (XY1 nella Figura 2.10). I series subcircuits sono formati da resistenze e induttanze divise per 2 in ciascun ramo, mentre il shunt subcircuit è una rete di capacità (e conduttanze, in caso di materiali dielettrici con perdite) poste in parallelo. Ogni capacità sarà connessa alla sua relativa conduttanza di perdita.

Per esempio, il shunt subcircuit nel caso di 3 conduttori indipendenti, avrà la seguente topologia circuitale:

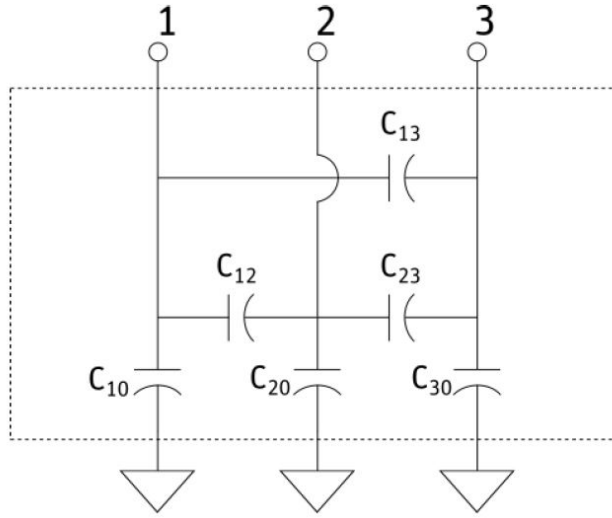


Figura 2.11: Shunt subcircuit del T-model nel caso di 3 conduttori distinti.

dove la Maxwell matrix delle capacità sarà:

$$C^M = \begin{pmatrix} C_{11}^M & C_{12}^M & C_{13}^M \\ C_{21}^M & C_{22}^M & C_{23}^M \\ C_{31}^M & C_{32}^M & C_{33}^M \end{pmatrix} \quad (2.11)$$

Invece la Spice Matrix equivalente, che Q3D deriva dalla Maxwell matrix, è:

$$C^S = \begin{pmatrix} C_{10} & C_{12} & C_{13} \\ C_{12} & C_{20} & C_{23} \\ C_{13} & C_{23} & C_{30} \end{pmatrix} \quad (2.12)$$

da cui si può generalizzare la struttura per n conduttori indipendenti. Anche qui, C_{10} , C_{20} e C_{30} rappresentano le capacità tra il singolo conduttore e ground. C_{12} , C_{13} e C_{23} sono le capacità mutue ($C_{12} = C_{21}$, $C_{13} = C_{31}$, $C_{23} = C_{32}$ nel caso di materiali dielettrici) e C_{11} , C_{22} , C_{33} sono le self-capacitances. Nel seguito le relazioni che le legano tutte, in un caso generale a n conduttori con/senza dielettrici.

Per passare **dalla Maxwell matrix alla Spice matrix**:

$$\begin{aligned}
 C_{10} &= C_{11}^M + C_{12}^M + C_{13}^M; & C_{12} &= -C_{12}^M; \\
 C_{20} &= C_{21}^M + C_{22}^M + C_{23}^M; & C_{13} &= -C_{13}^M; \\
 C_{30} &= C_{31}^M + C_{32}^M + C_{33}^M; & C_{23} &= -C_{23}^M;
 \end{aligned}
 \tag{2.13}$$

in generale, C_{i0} sarà la somma della i -esima riga della Maxwell matrix, mentre C_{ij} (con $i \neq j$) sarà il valore negativo della C_{ij}^M della Maxwell Matrix. Si ha inoltre che $C_{ij} = C_{ji}$ nel caso di materiali dielettrici.

Per passare **dalla Spice matrix alla Maxwell matrix**:

$$\begin{aligned}
 C_{11}^M &= C_{10} + C_{12} + C_{13}; & C_{12}^M &= C_{21}^M = -C_{12}; \\
 C_{22}^M &= C_{20} + C_{12} + C_{23}; & C_{13}^M &= C_{31}^M = -C_{13}; \\
 C_{33}^M &= C_{30} + C_{13} + C_{23}; & C_{23}^M &= C_{32}^M = -C_{23};
 \end{aligned}
 \tag{2.14}$$

in generale, C_{ii}^M sarà la somma della i -esima riga della Spice matrix, mentre C_{ij}^M (con $i \neq j$) sarà il valore negativo della C_{ij} della Spice Matrix. Si ha inoltre che $C_{ij}^M = C_{ji}^M$ nel caso di materiali dielettrici.

Quanto appena descritto per le capacità vale anche per le conduttanze G che sono proprio poste in parallelo a ciascuna capacità.

I series subcircuit invece, come già illustrato, sono costruiti utilizzando self e mutual inductances, self e mutual resistances. Per le induttanze (sia self che mutual), SPICE ha in libreria i componenti diretti che le modellizzano. Stesso discorso per le self-resistances. **Invece per le resistenze mutue (R_{ij} con $i \neq j$), SPICE non ha nessun componente diretto in libreria in grado di modellarle. Per ovviare a questo fatto, SPICE si avvale di utilizzare dei generatori pilotati.** Prendiamo in esempio il caso di Figura 2.6 descritto precedentemente, di un singolo conduttore avente 2 sources e un sink e analizziamolo dal punto di vista di tensioni e correnti, come mostrato in quest'altra figura:

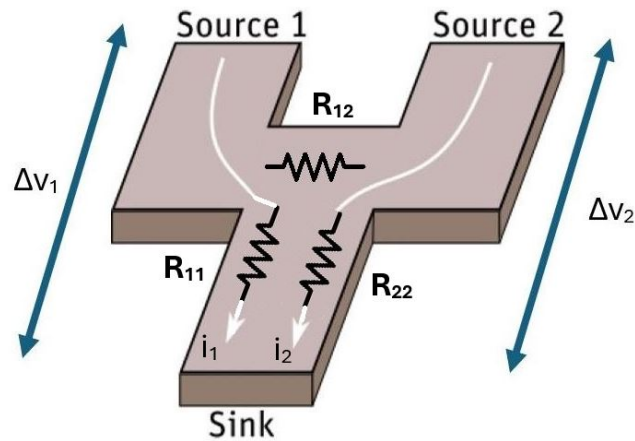


Figura 2.12: Esempio di singolo conduttore avente 2 sources e un sink con focus sulle resistenze.

In questo esempio, focalizziamoci sulla matrice delle resistenze che modella la relazione che vi è tra tensioni e correnti. Le cadute di tensione saranno rispettivamente:

$$\begin{aligned}\Delta v_1 &= R_{11}i_1 + R_{12}i_2; \\ \Delta v_2 &= R_{21}i_1 + R_{22}i_2;\end{aligned}\tag{2.15}$$

riscriviamo Δv_1 e Δv_2 raccogliendo R_{11} e R_{22} a fattore comune come:

$$\begin{aligned}\Delta v_1 &= R_{11}\left(i_1 + \frac{R_{12}}{R_{11}}i_2\right); \\ \Delta v_2 &= R_{22}\left(\frac{R_{21}}{R_{22}}i_1 + i_2\right);\end{aligned}\tag{2.16}$$

dove $R_{21} = R_{12}$.

Quanto appena scritto è modellizzato nella figura sottostante dove sono comprese anche le induttanze:

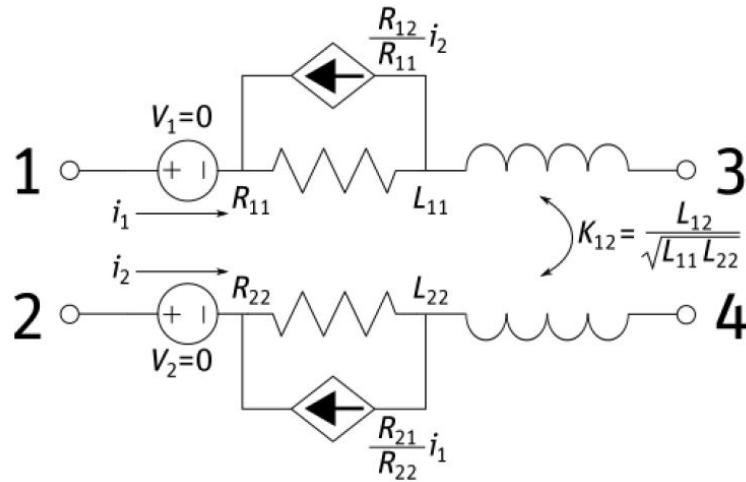


Figura 2.13: Series subcircuit del T-model equivalente ($R_{21} = R_{12}$).

Δv_1 e Δv_2 rappresentano rispettivamente le cadute di tensione sulle resistenza R_{11} e R_{22} nell'immagine di sopra.

La figura di sopra rappresenta un modello più dettagliato delle topologie circuitali di Figura 2.7 e Figura 2.10, perchè modella, utilizzando un generatore di corrente pilotato, il comportamento e la presenza della resistenza mutua R_{12} del singolo branch di impedenza $Z_{11}/2$. Il guadagno di questo generatore di corrente pilotato in corrente è dato proprio dal rapporto tra la resistenza mutua R_{12} e la self-resistance R_{11} o R_{22} a seconda del source che si considera. Quanto appena illustrato vale anche per casi in cui si hanno più di due sources e per più di un conduttore: in generale Q3D aggiungerà altri generatori di corrente pilotati per modellizzare la presenza delle resistenze mutue R_{ij} (con $i \neq j$).

Occorre precisare che il modello in Figura 2.13, per semplicità, è ottenuto trascurando le capacità C e le conduttanze G. Per questo motivo i rami circuitali comprendenti R e L non sono spezzati a metà con $R/2$ e $L/2$ e si ha quindi un unico ramo. A fine di illustrare come le resistenze mutue vengono modellizzate su SPICE. Ovviamente, e anche nel caso in cui si tenessero in considerazione le C e G, la presenza delle R mutue sarebbe modellizzata ugualmente con dei generatori pilotati.

Q3D permette anche di ottenere circuiti equivalenti RLCG ad una determinata frequenza oltre a quelli basati sulla topologia T-model (avente un sola cella). Infatti se nelle impostazioni si opta per un numero di celle $N > 1$ la topologia ottenuta avrà come elemento base il T-model, e prende il nome di **Ladder-model**:

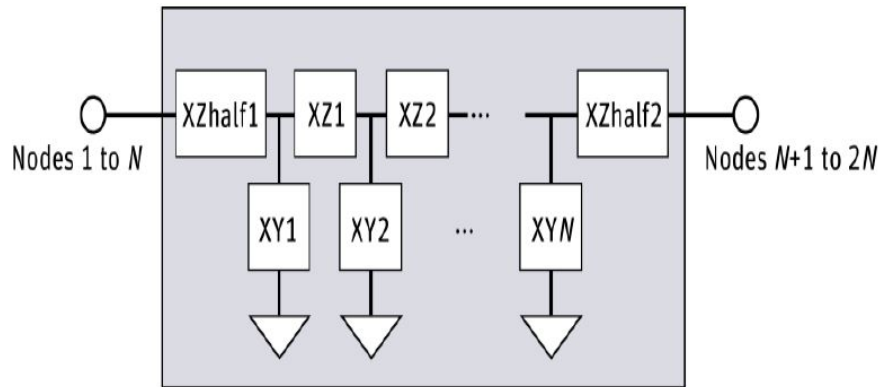


Figura 2.14: Generica topologia circuitale detta Ladder-model, avente N celle.

dove **XZhalf1** e **XZhalf2** sono i rami aventi resistenze e induttanze pari a $1/(2N)$ volte le matrici di resistenze e induttanze originali. Invece **XZ1**, **XZ2**, ..., **XZN-1** rappresentano gli elementi serie aventi resistenze e induttanze pari a $1/N$ volte le matrici di resistenze e induttanze originali. Ovviamente la resistenza totale degli elementi **XZhalf1**, **XZ1**, **XZ2**, ..., **XZN-1**, **XZhalf2** in serie è uguale alla resistenza nella matrice originale. Stesso discorso per l'induttanza. Invece per quanto riguarda i shunt elements **XY1**, **XY2**, ..., **XYN**, essi rappresentano $1/N$ volte le matrici delle capacità originali. La capacità totale della struttura è uguale a quella della matrice originale. Lo stesso principio è valido anche per le conduttanze G .

Come fatto in precedenza, va inoltre precisato che le resistenze e le induttanze che modellano un dato modello sono dipendenti dalla frequenza per via del fatto che all'aumentare della frequenza si presenta il fenomeno dello skin-effect che sarà argomento nei prossimi capitoli. Ora sono state definite le topologie circuitali con la quale il circuito RLCG ad una determinata frequenza è costruito. Nel prossimo capitolo verrà illustrato come sono calcolate rispettivamente le matrici R , L , C , G al variare della frequenza.

Su Q3D, dopo aver effettuato la simulazione del circuito, per ottenere il circuito equivalente RLCG ad una certa frequenza è necessario scegliere tra le impostazioni la frequenza di interesse, il numero di celle e dove salvare la netlist generata avente estensione .cir.

Tutta la teoria descritta fino ad ora può essere generalizzata per strutture circuitali molto più grandi e complesse rispetto a quelle descritte in questo paragrafo.

2.1.3 Calcolo della broadband netlist con Q3D [10]

La broadband netlist modella il circuito RLCG valido però a tutte le frequenze di segnali di ingresso sinusoidali e onde quadre. Fino ad ora si è visto come Q3D costruisce modelli equivalenti RLCG validi solamente ad una certa frequenza (valida per un segnale sinusoidale in ingresso). La broadband netlist è una rete di resistenze, capacità e generatori pilotati che riproduce il comportamento del circuito a tutte le frequenze di interesse.

Q3D è in grado di generare questa broadband netlist grazie ad un suo tool interno denominato **broadband export tool**. Questo tool riesce a costruire la broadband netlist cercando di ricalcare con una certa accuratezza i parametri S di un dato circuito utilizzando l'algoritmo di **vector fitting** che è illustrato nella sezione 4.5.

2.2 ADS

ADS è un altro software utile per effettuare simulazioni elettromagnetiche di layout e, per gli scopi di questo studio, si rivela utile anche per la generazione delle broadband netlist valide su tutto il range di frequenza che si prende in analisi.

Una volta definito/creato/importato il layout di un dato circuito (ad esempio l'induttore in tecnologia PCB) su ADS i principali step che si seguono per effettuare la simulazione elettromagnetica sono i seguenti:

1. **Scelta del simulatore elettromagnetico tra Momentum Microwave, Momentum RF e FEM** a seconda della tipologia di circuito che si intende analizzare;
2. **Definizione del substrato:** spessori e numero dei differenti strati che lo compongono, posizione, spessore e materiali dei via, dei conduttori;
3. **Assegnazione delle porte** e relative impostazioni ad esse riguardanti;
4. **Definizione del frequency plan:** start e stop frequencies e tipo di sweep (lineare, logaritmico, adaptive, ecc...). Per gli scopi di questa trattazione lo sweep lineare e logaritmico si sono rivelati adatti;
5. **Se necessario, setting di altre opzioni di simulazione:** impostazioni relative alle **mesh**, alla **temperatura**, ecc...;
6. **Start della simulazione elettromagnetica** del circuito [13].

ADS utilizza tre diversi metodi di simulazione elettromagnetica: Momentum RF, Momentum Microwave e FEM. Nel seguito sono descritti nel dettaglio.

2.2.1 Momentum RF e Momentum Microwave

Momentum (sia RF che Microwave) è utilizzato per calcolare i parametri S di Microwave/RF PCBs, IC (circuiti integrati), circuiti planari, in microstriscia, moduli multichip, guide d'onda coplanari e altre tipologie [14].

Momentum RF e Momentum Microwave sono simulatori planar 3D. Significa che le interazioni elettromagnetiche sono considerate su tutte e 3 le dimensioni spaziali e che i circuiti presi in esame sono stratificati su dei layer dielettrici, come ad esempio i PCB [1]. Questi due simulatori sono basati sul Metodo dei momenti (MoM).

Il **metodo dei momenti** consiste nel suddividere la superficie conduttiva in celle rettangolari, triangolari, o in generale, poligonali, che formano la mesh. Ogni bordo della cella è attraversato da una certa corrente. Le teoria ci porta ad ottenere per ogni cella della mesh, un circuito LC equivalente che poi viene utilizzato come mattoncino fondamentale per modellizzare tutta la superficie conduttiva, come mostrato per il circuito in figura [15]:

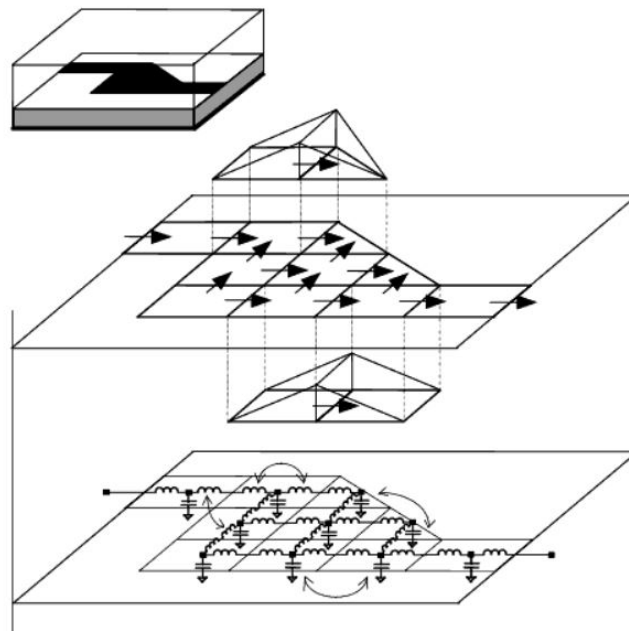


Figura 2.15: Rappresentazione del meshing sulla superficie conduttiva.

Nel seguito sono mostrati tutti i passaggi matematici che caratterizzano il metodo dei momenti e che illustrano come il simulatore calcola i parametri L e C [15] del circuito mostrato qui sopra:

1. **Calcolo delle funzioni di Green relative al substrato:** le funzioni di Green caratterizzano il substrato e rappresentano le risposte spaziali del

substrato alle eccitazioni impulsive di tipo delta di Dirac; queste sono calcolate per ogni coppia di strati di segnali presenti nel substrato. Utilizzando queste funzioni di Green, con il metodo dei momenti si definisce la MPIE (Mixed Potential Integral Equation) come segue:

$$\iint dS \vec{G}(r, r') \cdot J(r) = E(r) \quad (2.17)$$

dove $J(r)$ rappresenta le correnti superficiali incognite e $E(r)$ il campo elettrico che eccita la struttura circuitale.

2. **Meshing dei modelli di percorsi dei segnali:** tutte le metalizzazioni planari, i via e le aperture definite nei layer di segnale vengono suddivise in rettangoli o triangoli (a seconda della complessità definita nell'equazione (2.31)), come mostrato nella figura seguente:

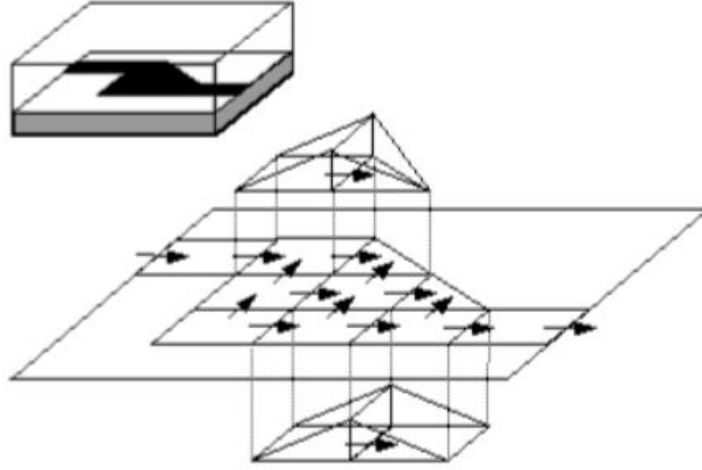


Figura 2.16: Rappresentazione delle correnti che fluiscono attraverso i bordi delle celle nella mesh. fonte guida di ADS

Il risultato è che la corrente di superficie $J(r)$ viene discretizzata a seguito del meshing utilizzando delle basis functions B_j rooftop (tetto): ogni B_j è associato ad un bordo della mesh e rappresenta una corrente con densità costante che fluisce attraverso un bordo, come mostrato sempre in Figura 2.16. L'ampiezza I_j incognita di queste B_j rappresenta la corrente che scorre attraverso un certo bordo della mesh. Si ha quindi che:

$$J(r) = \sum_{j=1}^N I_j B_j(r) \quad (2.18)$$

Il passo successivo è quello di discretizzare la MPIE sostituendo nell'equazione (2.17) la $J(r)$ discretizzata dell'equazione (2.18) e si ha che:

$$\sum_{j=1}^N \iint dS \vec{G}(r, r') \cdot I_j B_j(r) = E(r) \quad (2.19)$$

ora si applica la Galerkin testing procedure (??) moltiplicando primo e secondo membro per la funzione test evidenziata in blu identica alla basis function:

$$\sum_{j=1}^N \iint_S dS B_i(r) \cdot \iint_{S'} dS' \vec{G}(r, r') \cdot I_j B_j(r) = \iint_S dS B_i(r) \cdot E(r) \quad (2.20)$$

che scritto in maniera più compatta risulta essere:

$$\sum_{i=1}^N Z_{ij} I_j = V_i \quad \Rightarrow \quad [Z] \cdot [I] = [V] \quad (2.21)$$

con

$$Z_{ij} = \iint_S dS B_i(r) \cdot \iint_{S'} dS' \vec{G}(r, r') B_j(r) \quad (2.22)$$

$$V_i = \iint_S dS B_i(r) \cdot E(r) \quad (2.23)$$

dove $[Z]$ rappresenta la **interaction matrix**. L'elemento Z_{ij} di $[Z]$ definisce l'interazione elettromagnetica tra due rooftop basis functions confinanti aventi superfici S e S' rispettivamente. $[V]$ invece rappresenta il contributo discretizzato del campo elettrico che è applicato alle porte del circuito.

In MPIE la funzione di Green $\vec{G}(r, r')$ è composta da due parti: $G^A(r, r')$ che riguarda il vettore potenziale $A(r)$ e $G^V(r, r')$ che è riferita al potenziale scalare $V(r)$:

$$\vec{G}(r, r') = j\omega G^A(r, r') \vec{I} - \frac{1}{j\omega} \nabla[G^V(r, r') \nabla'] \quad (2.24)$$

3. **Sostituendo $\vec{G}(r, r')$ nella definizione di Z_{ij} nell'equazione (2.22)**, il simulatore costruisce e risolve l'equazione della matrice di interazione $[Z]$ del Metodo dei momenti, ricavando infine i componenti delle matrici Z , L e C come:

$$Z_{ij} = j\omega L_{ij} + \frac{1}{j\omega C_{ij}} \quad (2.25)$$

$$L_{ij} = \iint_S dS B_i(r) \cdot \iint_{S'} dS' G^A(r, r') B_j(r') \quad (2.26)$$

$$\frac{1}{C_{ij}} = \iint_S dS \nabla \cdot B_i(r) \iint_{S'} dS' G^V(r, r') \nabla \cdot B_j(r') \quad (2.27)$$

Una generica cella della mesh, quindi, sarà caratterizzata da un certo L_{ij} e C_{ij} . Ogni cella è sostituita con una capacità C_{ij} collegata direttamente a ground e un induttore L_{ij} collegato al bordo con l'altra cella confinante, come illustrato nella figura seguente:

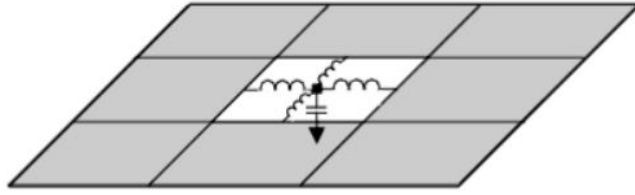


Figura 2.17: Modello circuitale LC di una singola cella della mesh.

Nella Figura 2.15 è mostrata tutta la rete LC equivalente riguardante tutta la struttura metallica che viene discretizzata dal metodo dei momenti.

4. **Calibrazione e scorporamento dei parametri S:** la calibrazione è effettuata eliminando gli effetti dovuti alle sorgenti connesse alle linee di trasmissione delle porte. In questo modo vengono rimossi i parassiti dovuti alle sorgenti che eccitano le porte. Lo scorporamento, detto anche de-embedding, invece consiste nello scalare la fase dei parametri S per eliminare gli effetti dovuti alle linee di trasmissione: il simulatore infatti per fare ciò necessita e si calcola la costante di propagazione e l'impedenza caratteristica di queste linee di trasmissione. Ad alte frequenze, queste linee di trasmissione hanno una lunghezza pari a $\lambda/2$ (metà della lunghezza d'onda) mentre a bassa frequenza hanno una lunghezza minore [15].

Per Momentum sono disponibili due modalità di simulazione: **RF (detto anche quasi-static mode)** e **Microwave (anche detto full-wave mode)**. Nel seguito sono descritte le differenze tra le due modalità di calcolo: entrambi i simulatori sono basati sulla metodologia appena descritta del Metodo dei Momenti.

Momentum RF usa una formulazione di tipo quasi-static: effettua delle approssimazioni rendendo le funzioni di Green (che teoricamente sono dipendenti dalla

frequenza) indipendenti dalla frequenza. Questo rende i calcoli più veloci, perchè le soluzioni (matrici L e C, descritte in precedenza) vengono calcolate a bassa frequenza, più precisamente ad una sola frequenza (la prima del range di interesse). **Momentum Microwave** invece, non effettua nessuna approssimazione sulle funzioni di Green che rimarranno dunque dipendenti dalla frequenza. Per via di questo, Momentum RF risulta più veloce perchè le matrici L e C sono calcolate ad una sola frequenza (la prima). **Momentum RF** può essere usato quando si ha a che fare con circuiti di radio frequenza o anche microwave, e più nello specifico risulta essere efficiente e conveniente utilizzarlo quando si ha a che fare con **circuiti elettricamente piccoli, geometricamente complessi e che non irradiano**[14] [15] [16]. Nel seguito sono spiegati i rispettivi significati di questi tre requisiti appena elencati:

Per **circuiti elettricamente piccoli** si intende quei circuiti aventi lunghezza fisica massima minore di metà della lunghezza d'onda riferita alla frequenza. Infatti Momentum Microwave risulta essere sempre meno preciso al crescere della lunghezza elettrica. Per capire quale è la massima lunghezza bisogna prima capire quale è la frequenza massima di funzionamento.

Per esempio se si vuole valutare se un circuito che irradia onde nello spazio circostante si può considerare elettricamente piccolo ci si fa ad una delle seguenti due condizioni, che vanno utilizzate come linee guida solo per sapere quale è la massima dimensione impostabile ad una determinata frequenza:

$$D < \frac{\lambda}{2} = \frac{c}{2f} = \frac{3 \cdot 10^8 m/s}{2f} = \frac{150 \cdot 10^6}{f} [m] = \frac{150}{f [GHz]} [mm] \quad (2.28)$$

$$f_{MAX} < \frac{150}{D} \quad (2.29)$$

Dove f_{MAX} rappresenta la massima frequenza di funzionamento in GHz e D è la massima lunghezza in mm diagonalmente attraverso il circuito. Se si conosce già la massima lunghezza D del circuito, per stimare la massima frequenza per la quale si può considerare che un circuito è elettricamente piccolo, si può utilizzare l'equazione (2.29). Viceversa, se f_{MAX} è già noto, si può utilizzare l'equazione (2.28) per trovare la massima lunghezza D. Il simulatore Momentum RF calcola f_{MAX} dato che D solitamente è già fissato dal progettista. In questo modo la frequenza di simulazione viene calibrata di conseguenza.

Invece nel caso di dispositivi dove l'irradiazione è a livello superficiale, come quello nella figura sottostante, la condizione di circuito elettricamente piccolo varia, come riportato nell'equazione (2.30) che regola la relazione tra frequenza f e spessore T del substrato.

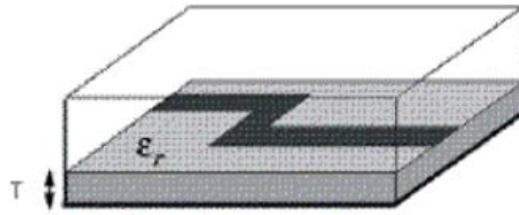


Figura 2.18: Esempio di dispositivo che irradia solamente a livello superficiale.

$$f[\text{GHz}] < \frac{300}{T[\text{mm}] \cdot \sqrt{\varepsilon_r - 1}} \quad \alpha = \begin{cases} 20 & \text{con groundplane} \\ 10 & \text{senza groundplane} \end{cases} \quad (2.30)$$

Circuiti geometricamente complessi: per valutare questa condizione bisogna valutare le mesh generate in fase di simulazione: in un circuito geometricamente complesso la mesh genera tanti triangoli. All'aumentare del numero di triangoli aumenta il livello di complessità della struttura. Il livello di complessità invece è considerato basso quando il meshing genera tanti rettangoli (e quindi pochi triangoli). In generale, il grado di complessità è dato dal rapporto tra il numero di celle triangolari e il numero di celle rettangolari generate dal meshing, come scritto nell'equazione sottostante:

$$\text{Complessità} = \frac{N^\circ \text{ triangoli}}{N^\circ \text{ rettangoli}} \quad (2.31)$$

Irradiazione: in radio frequenza (a frequenze elevate, quindi) questo fenomeno cresce sempre più e per questo motivo Momentum RF darà risultati sempre meno accurati [15]. Per questo motivo Momentum RF si è consigliabile utilizzarlo su circuiti che non irradiano.

Momentum Microwave è da utilizzarsi quando si ha a che fare con circuiti dove sono inclusi effetti di radiazione microwave. Momentum RF nei suoi calcoli non tiene conto dei fenomeni di radiazione. Ove necessario, Momentum RF lo si può utilizzare per sistemi dove è possibile ignorare gli effetti della radiazione, qual'ora presenti. In questo modo si riesce ad andare più veloci con la simulazione [16] e si risparmia memoria.

2.2.2 FEM

FEM è adatto per la simulazione di RF e Microwave circuits [17]. Il simulatore FEM utilizza il Finite element method: in FEM il layout del circuito è diviso in tanti piccoli tetraedri. Ogni tetraedro è formato da 4 triangoli equilateri. FEM risulta utile per circuiti in microstriscia, strutture multistrato, filtri, couplers, antenne, power splitters/combiners e altri.

Invece FEM è utilizzato per applicazioni inerenti a circuiti in RF, PC boards e per applicazioni riguardanti la signal integrity. Risulta valido anche per circuiti high-speed e high-frequency [17].

Il simulatore FEM è basato sulla tecnica del finite element method. Essendo un simulatore 3D, permette di calcolare il campo elettromagnetico su tutto lo spazio che circonda una data struttura circuitale.

Con il finite element method la geometria del circuito è divisa in tanti tetraedri. Ogni singolo tetraedro è formato da 4 triangoli equilateri. **I nodi sono definiti in maniera analoga a quanto descritto per il simulatore Q3D, nel paragrafo precedente.** Tale concetto è raffigurato nell'immagine 2.1 [18].

L'accuratezza della soluzione dipende dal livello di meshing e dalla grandezza della struttura: in circuiti grandi un basso livello di meshing produrrà soluzioni poco accurate, ma se si esagera ed il numero di tetraedri è sempre più alto ed essi sono sempre più piccoli si guadagnerà in accuratezza ma si il calcolo sarà lento perchè richiederà sempre più memoria, sempre più risorse. Occorre, in generale trovare il giusto trade-off.

Il simulatore FEM, per raggiungere il giusto compromesso tra accuratezza e livello di meshing, in modo automatico utilizza un processo iterativo che rifinisce le mesh nei punti più critici della geometria, dove serve. Inizialmente genera una soluzione iniziale poco precisa basata su un meshing limitato e poi rifinisce man mano le mesh in modo da non superare una certa soglia di errore. Nel fare ciò calcola i parametri S, e quando questi convergono verso un certo valore rispettando un certa soglia di errore, fa terminare l'iterazione.

Diventa importante sottolineare che le componenti riguardanti il campo magnetico H risultano essere meno accurate del campo elettrico E in virtù del fatto che H è calcolato a partire da E secondo la relazione che segue:

$$H = \frac{\nabla \times E}{-j\omega\mu} \quad (2.32)$$

Nell'eseguire la simulazione elettromagnetica, per il calcolo dei parametri S di una certa struttura, il simulatore FEM segue gli step seguenti:

1. **Meshing**, cioè la struttura 3D è divisa in tanti tetraedri;

2. **Calcolo delle onde di potenza** in ogni porta del circuito;

3. **Calcolo del campo elettromagnetico interno alla struttura** nel caso in cui ogni porta sia eccitata da una delle onde di potenza;

4. **Calcolo dei parametri S** in base alle riflessioni e trasmissioni che si generano su ogni porta del circuito.

Per prima cosa è necessario definire il campo che eccita ciascuna porta, e per fare ciò si ragiona in 2D.

Il circuito viene stimolato nel modo seguente: ciascuna porta è connessa a una guida d'onda avente la stessa sezione trasversale (meglio conosciuta come cross-section) della porta. Per le porte, il meshing è compiuto in 2D: le facce dei tetraedri che giacciono sull'area della porta saranno proprio i triangolini che formano la mesh in 2D che quindi è incorporata nella mesh in 3D. Ciò rappresenta una delle condizioni al contorno del problema generale in 3D.

All'ingresso della guida d'onda viene applicato un campo elettrico avente la seguente forma:

$$E(x, y, z, t) = \text{Re}[E(x, y)e^{j\omega t - \gamma z}] \quad (2.33)$$

Dove $\gamma = \alpha + j\beta$ è la costante di propagazione complessa del campo. Andando più nello specifico, α rappresenta la costante di attenuazione associata al campo, mentre β è la costante di propagazione che determina come la fase varia lungo la direzione z .

Avendo assunto che la porta abbia la stessa sezione trasversale della guida d'onda alla quale è connessa, nell'equazione sopra si è posto che l'interfaccia della porta giaccia sul piano avente coordinata $z = 0$. Cioè il sistema di coordinate è definito nell'immagine sottostante:

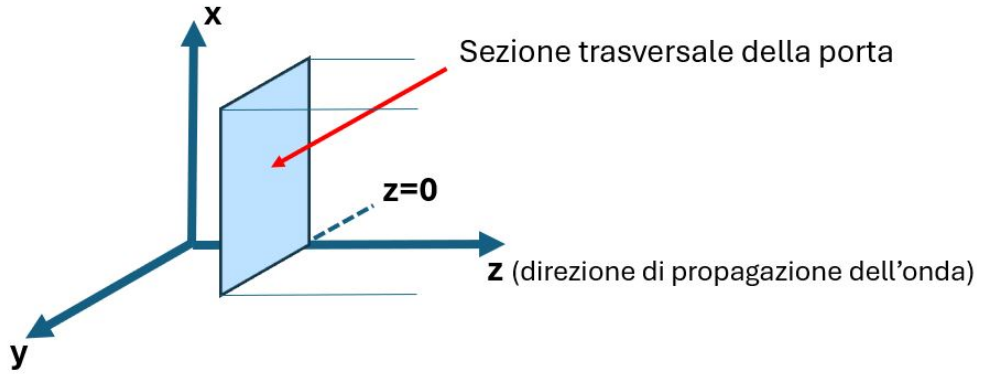


Figura 2.19: Sistema di coordinate utilizzato.

Il simulatore poi, per calcolare l'onda che attraversa la guida d'onda risolve le equazioni di Maxwell da cui genera la seguente equazione in 2D:

$$\nabla \times \left(\frac{1}{\mu_r} \nabla \times E(x, y) \right) - k_0^2 \varepsilon_r E(x, y) = 0 \quad (2.34)$$

dove il fasore $E(x, y)$ è l'incognita. Una volta risolta l'equazione d'onda scritta sopra, il simulatore controlla che la soluzione rispetti la soglia di errore imposta e se non la rispetta viene fatto un refinement della mesh della porta ed eseguita una nuova iterazione per ottenere una soluzione più precisa, che rispetti le tolleranze. Per refinement della mesh si intende che il simulatore divide la superficie in triangolini ancora più piccoli in modo da aumentare l'accuratezza.

Fino a qui il simulatore FEM esegue i calcoli in 2D. E questo rappresenta una delle condizioni al contorno per il problema in 3D.

Arrivati fin qui occorre prestare attenzione al fatto che i modi interni alla guida d'onda che soddisfano le equazioni di Maxwell fino a questo punto potrebbero essere più di uno e i parametri S dovranno tenerne conto, se si vuole mantenere un alto livello di accuratezza. Se si vuole evitare questo, bisognerà incrementare la lunghezza della guida d'onda il tanto da far decadere i modi di ordine più alto.

Si passa ora al caso 3D, dove l'equazione d'onda diventa la seguente:

$$\nabla \times \left(\frac{1}{\mu_r} \nabla \times E(x, y, z) \right) - k_0^2 \varepsilon_r E(x, y, z) = 0 \quad (2.35)$$

ed il campo elettrico (interno alla struttura circuitale) è definito come:

$$E(x, y, z, t) = \text{Re}[E(x, y, z)e^{j\omega t}] \quad (2.36)$$

Dopodichè, a partire dal campo elettrico incidente, trasmesso e riflesso in una certa porta, vengono calcolati i parametri S , che vengono definiti come descritto nelle equazioni (2.8) ed (1.4) nel Capitolo 1, Sezione 1.3.

A questo punto, la matrice S trovata non rappresenta ancora la soluzione definitiva perchè deve essere **normalizzata** rispetto all'impedenza che di solito è 50Ω .

Per fare ciò, il simulatore calcola la matrice delle impedenze Z riguardante tutto il circuito:

$$Z = \sqrt{Z_0}(I - S)^{-1}(I + S)\sqrt{Z_0} \quad (2.37)$$

dove Z_0 sarà la matrice avente nella sua diagonale i valori originali delle impedenze caratteristiche di ciascuna porta. I invece rappresenta la matrice identità.

La matrice S rinormalizzata secondo l'impedenza opportuna è calcolata sostituendo Z nella seguente equazione matriciale:

$$S_\Omega = \sqrt{Y_\Omega}(Z - Z_\Omega)(Z + Z_\Omega)^{-1}\sqrt{Y_\Omega} \quad (2.38)$$

dove Z_Ω e Y_Ω sono le matrici diagonali, aventi come valori della diagonale, rispettivamente l'impedenza e ammettenza rispetto il quale si intende rinormalizzare i parametri S . Se per esempio si vuole rinormalizzare la matrice S rispetto a 50Ω , i valori della diagonale di Z_Ω varranno 50Ω .

La matrice S appena normalizzata rispetto a 50Ω , ottenuta dall'equazione (2.38), risulta ancora influenzata dal fatto che su ciascuna porta sia stata connessa una guida d'onda. Ed è proprio la lunghezza fisica di questa guida d'onda che influisce sulla fase di un certo S_{ij} . Quanto appena descritto è modellizzato dalla relazione seguente che permette di trovare la matrice S' finale eseguendo cioè un **de-embedding dei parametri S**:

$$S' = e^{\gamma l} S e^{\gamma l} \quad (2.39)$$

dove, per esempio, nel caso di circuito a 3 porte:

$$e^{\gamma l} = \begin{pmatrix} e^{\gamma_1 l_1} & 0 & 0 \\ 0 & e^{\gamma_2 l_2} & 0 \\ 0 & 0 & e^{\gamma_3 l_3} \end{pmatrix} \quad (2.40)$$

nel quale γ_1 e l_1 rappresentano rispettivamente la costante di propagazione e la lunghezza della guida d'onda (aggiunta o rimossa) nella porta 1. Stesso significato per le altre porte 2 e 3. Se $l_1 < 0$ significa che la guida d'onda di lunghezza l_1 è stata aggiunta al circuito. Se invece $l_1 > 0$ significa che la linea di trasmissione di lunghezza l_1 era stata rimossa.

2.2.3 Calcolo della broadband netlist con ADS

Anche su ADS è presente un tool che è in grado di generare una broadband netlist equivalente ai circuiti presi in esame. Esso prende il nome di **Broadband SPICE Model generator**. Questo tool prende in ingresso i parametri S di un certo circuito e genera la netlist che sarà caratterizzata da parametri S (e quindi matrice delle impedenze Z) che sono il risultato del fitting dei parametri S originali. Come per Q3D, anche su ADS il metodo con la quale si approssimano i parametri S di un certo circuito è quello del **vector fitting**, che è illustrato nella sezione 4.5.

2.2.4 Calcolo della broadband netlist con Matlab

Anche su Matlab è possibile calcolare una broadband netlist grazie alla funzione **rational()** che esegue il **vector fitting** dei dati che riceve in ingresso. Quest' ultima necessita di avere il file touchstone dei parametri S e l'impedenza di riferimento alle porte del circuito.

2.3 EMPro

EMPro è un altro tool specializzato nell'eseguire simulazioni elettromagnetiche. Il simulatore può, a seconda della tipologia del circuito, utilizzare due metodi alternativi per eseguire i calcoli: il primo è **FEM** e il secondo è **FDTD**. Per quanto riguarda il simulatore FEM, quanto spiegato nella Sezione riguardante il tool ADS nel presente capitolo, vale anche per EMPro. Il principio di funzionamento e le equazioni sono identiche.

Il metodo **FDTD (Finite-difference Time-Domain)** risolve le equazioni di Maxwell nel dominio del tempo. (fonte EMPro documentation della Keysight) Con questo metodo sia il tempo che lo spazio è diviso in segmenti discreti. Lo spazio è suddiviso in tante piccole celle cubiche, e le componenti lungo x, y, z dei campi elettrico e magnetico sono disposte in ciascun cubo come mostrato nella figura seguente.

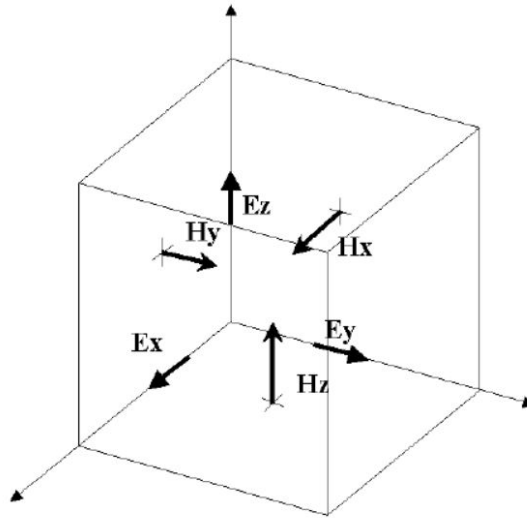


Figura 2.20: Rappresentazione di una singola cella cubica.

Le componenti E_x , E_y e E_z del campo elettrico sono poste lungo i bordi del cubo, mentre le componenti H_x , H_y e H_z sono perpendicolari alle facce del cubo. Invece il tempo è suddiviso in step che rappresentano il tempo impiegato dal campo per passare da una cella all'altra. Ad ogni step temporale i campi elettrico e magnetico vengono calcolati.

Il griglia è formata da tante celle cubiche combinate tra loro che formano il volume 3D del dispositivo, come si può evincere dall'immagine di esempio sottostante.

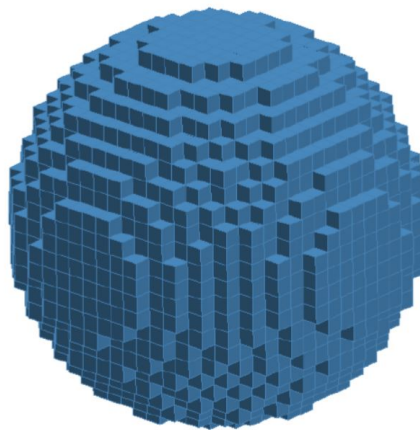


Figura 2.21: Esempio di griglia in cui è suddivisa la sfera blu nell'immagine.

dove ogni piccolo cubetto rappresenta una cella FDTD della griglia. Per griglia

si intende la geometria 3D dell'immagine sopra senza la definizione dei materiali, mentre per mesh si intende la stessa geometria della griglia ma con i materiali definiti. EMPro offre una vasta gamma di materiali sulla quale scegliere.

Le dimensioni della cella determinano lo step temporale e anche la massima frequenza, come regola generale si può imporre una risoluzione minima di 10 celle per lunghezza d'onda. Quindi ogni cella avrà una dimensione massima di un decimo della lunghezza d'onda. $\lambda = c/f$ cioè $l < c/(10f)$

EMPro considera l'intero volume della Figura 2.21 come near-zone region: focalizzando su un punto della mesh, il campo è calcolato in funzione del tempo. Oltre al campo, anche altre quantità come impedenze e parametri S sono considerati dal simulatore come valori di near-zone.

In caso si volessero avere risultati in far-field, una prima idea potrebbe essere quella di estendere il meshing aggiungendo nuove celle in modo da raggiungere la zona considerata di far-field: questo diventerebbe estremamente costoso in termini di tempi di calcolo e di memoria. EMPro infatti segue una strada più conveniente: trasforma i valori di un certo parametro calcolati in near-field in valori calcolati in far-field lontani dalla mesh. Il simulatore racchiude l'intera geometria in un unico grande cubo e salva i campi nelle sei facce del cubo. Le facce di questo cubo sono posizionate a 5 celle da ciascun bordo esterno della griglia. Il sistema di coordinate utilizzato per le zone di far-field è il seguente:

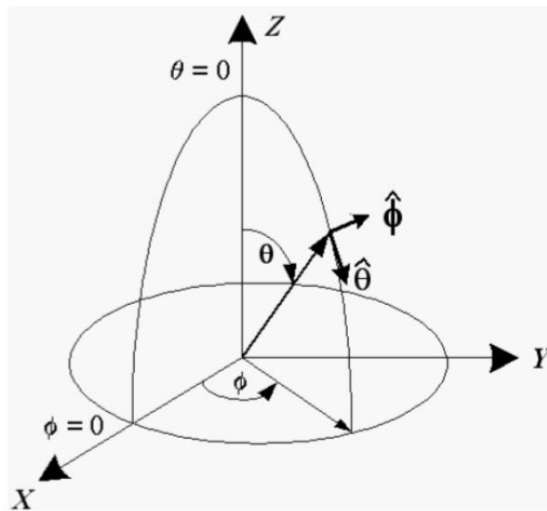


Figura 2.22: Sistema di riferimento in far-field.

Il timestep massimo consentito dipende dalle dimensioni delle celle che compongono la geometria, ed è determinato come:

$$\Delta t_{MAX} = \frac{1}{c} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2} \right)^{-1/2} \quad (2.41)$$

dove $c = 3 \cdot 10^8 m/s$ è la velocità della luce nel vuoto e Δx , Δy e Δz sono le lunghezze dei lati delle celle.

Bisogna infine precisare che il timestep Δt utilizzato non deve essere maggiore del Δt minimo tra le celle di tutta la geometria.

Sia che si utilizzi il simulatore FEM che FDTD, le impostazioni da scegliere per effettuare la simulazione elettromagnetica sono:

1. Creazione/import su EMPro della geometria e definizione dei materiali;
2. Creazione della mesh;
3. Definizione parametri di simulazione come start frequency e stop frequency, scelta delle porte e loro segnale in ingresso;
4. Scelta dei risultati da visualizzare;
5. Starting della simulazione;
6. Visualizzazione dei risultati ottenuti [19].

2.4 SPICE

SPICE, in questo studio, è stato utilizzato per eseguire le simulazioni in tran e AC dei diversi circuiti analizzati. In questa sezione verrà spiegato come il simulatore SPICE calcola la soluzione dei circuiti quando viene effettuata **l'analisi tran, nel dominio del tempo**. Come simulatore SPICE è stato utilizzato **ELDO**.

La maggior parte delle volte, il sistema di equazioni che modella un certo circuito non può essere risolto analiticamente: per questo motivo il simulatore utilizza degli algoritmi numerici. La soluzione che ELDO deve trovare è un set di tensioni nel dominio del tempo:

$$v(t) = \begin{pmatrix} v_1(t) \\ v_2(t) \\ \vdots \\ v_N(t) \end{pmatrix} \quad (2.42)$$

dove N rappresenta il numero di nodi del circuito. Queste tensioni rappresentano la soluzione del seguente sistema differenziale di equazioni non lineari:

$$f(v(t), q'(v(t)), t) = 0 \tag{2.43}$$

che rappresenta la relazione di Kirchoff per le correnti: cioè che la somma di tutte le correnti entranti in un nodo vale zero, per ogni istante di tempo. La funzione $q()$ rappresenta le cariche del circuito dipendenti dal BIAS.

Il sistema di equazioni appena menzionato è detto DAE system, vale a dire **Differential Algebraic Equations system**. $f()$ e $q()$ sono funzioni non lineari. Per risolvere il sistema numericamente, il tempo è discretizzato e le equazioni vengono risolte per istanti di tempo discreti. Le derivate in funzione del tempo sono approssimate utilizzando i valori della soluzione $v(t)$ ad un certo istante t_x e agli istanti precedenti ad esso: vale a dire $v(t_x), v(t_{x-1}), \dots, v(t_{x-m})$, dove m rappresenta il numero di istanti di tempo analizzati. Arrivati a questo punto il sistema non è più differenziale ed è rimasto solamente non lineare: per risolverlo vengono utilizzati dei metodi iterativi. Uno dei più comuni è il metodo Newton-Raphson. Questo è un metodo iterativo nella quale la soluzione x all'iterazione $n + 1$ dipende dalla soluzione calcolata all'iterazione precedente. Il principio è il seguente:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2.44}$$

f è la funzione per la quale si intende trovare la soluzione.

La risoluzione del sistema appena descritto introduce principalmente due tipi di errore: uno è quello dovuto al numerical integration process e il secondo è riguardante la risoluzione numerica di equazioni non lineari.

Il tempo viene discretizzato dal simulatore definendo un time step che viene utilizzato per sostituire le derivate rispetto al tempo con differenze finite.

Di default, Eldo utilizza il metodo dell' **LTE (Local Truncation Error)**.

Con l'LTE, Eldo prende il valore della soluzione calcolata con il metodo di Newton-Raphson all' i -esima iterazione al tempo t e lo utilizza per calcolare la soluzione all'iterazione numero $i+1$ al tempo $t+h$. Il time step h è calcolato dal simulatore e rappresenta il valore di incremento di tempo che permette di avere un Local Truncation error entro un limite stabilito. Una volta accertato ciò, viene calcolata la soluzione all'istante $t+h$: se questa soluzione converge, viene calcolato l'LTE. Se quest'ultimo non supera la soglia imposta la soluzione al tempo $t+h$ è accettata e viene eseguita e viene iniziato un nuovo ciclo. Se invece la soluzione non converge, o l'LTE viene superato, il time step h viene ridotto a $h' < h$ fino a che la soluzione a $t+h'$ converge e NON viene superato l'LTE. Quindi ad ogni iterazione vanno contemporaneamente rispettati due vincoli: la Newton convergence e l'LTE. Il

metodo dell’LTE fornisce i risultati più accurati solitamente, rispetto al agli altri disponibili.

Newton: iterazioni e accuratezza

Su Eldo, l’accuratezza delle iterazioni del metodo di Newton viene controllata tramite le tre seguenti relazioni.

Per le tensioni:

$$|V(i) - V(i - 1)| < RELTOL \times |max(|V(i)|, |V(i - 1)|)| + VNTOL \quad (2.45)$$

Per le correnti:

$$|I(i) - I(i - 1)| < RELTOL \times |max(|I(i)|, |I(i - 1)|)| + ABSTOL \quad (2.46)$$

Per le cariche, invece:

$$|Q(i) - Q(i - 1)| < RELTOL \times |max(|Q(i)|, |Q(i - 1)|)| + CHGTOL \quad (2.47)$$

dove RELTOL, VNTOL, ABSTOL e CHGTOL sono delle tolleranze settate in modo da regolare l’accuratezza della soluzione, a seconda del circuito che si analizza. V(i) e V(i-1) rappresentano il valore della tensione alle iterazioni i e i-1 rispettivamente. Stesso discorso per la corrente I(i) e I(i-1) e la carica Q(i) e Q(i-1). Il numero di iterazioni sale al decrescere RELTOL per ottenere maggiore precisione [20].

Capitolo 3

Analisi in frequenza

Come descritto nel Capitolo 2, Sezione 2.1, il tool Q3D di ANSYS Electronics Desktop è in grado di generare il **circuito equivalente RLCG valido ad una sola frequenza** e la **broadband netlist valida su tutto il range di frequenze** per un dato layout di una struttura complessa a n porte. Questo circuito equivalente RLCG modella i parametri parassiti ad una certa frequenza. Il layout quindi, sarà modellizzato da una certa topologia circuitale equivalente RLCG più o meno complessa che si basa sul modello a T, come è stato illustrato nel capitolo precedente. Al termine del precedente capitolo si era scritto che i parametri R ed L sono dipendenti dalla frequenza, e in questo nuovo capitolo si spiegherà in che modo R ed L variano. La fonte delle nozioni teoriche illustrate in questo capitolo (Sezione 3.1) è riportata in [10]. Inoltre è riportato il risultato di varie simulazioni che indicano come i parametri S (e quindi l'equivalente RLCG) di un certo circuito variano se si inietta in ingresso un'onda quadra, invece che una sinusoidale.

3.1 Variazione dei parassiti RLCG al variare della frequenza

Vi sono 3 diversi range di frequenza nella quale resistenza e induttanza parassite hanno un andamento diverso: **DC region**, **transition region** e **AC region**. In DC region sia la R che la L hanno un andamento costante mentre in transition region, col crescere della frequenza, la R inizia a crescere mentre l'induttanza L inizia a decrementare il suo valore. A frequenze sempre più alte, cioè entrando in AC region, l'induttanza si stabilizza ad un valore costante (minore di quello che aveva in DC region) e la resistenza R aumenta proporzionalmente con la radice della frequenza. Quanto appena scritto è riassunto nella figura seguente:

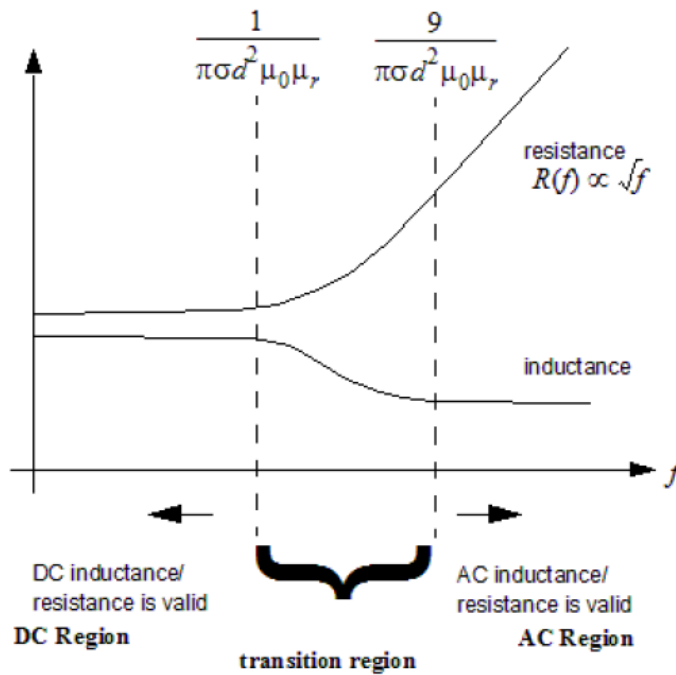


Figura 3.1: Grafici che modellizzano l'andamento in frequenza di R ed L parassite.

Questa variazione di R ed L al variare della frequenza è dovuta ad un fenomeno chiamato **Skin-effect (effetto pelle)**.

3.1.1 Skin-effect

Per spiegare il fenomeno è necessario focalizzarsi sulla densità di corrente J che scorre all'interno di un singolo conduttore: in DC region la densità di corrente risulta essere uniformemente distribuita su tutta la sezione del conduttore come mostrato in Figura 3.2. Con l'aumentare della frequenza, e quindi avvicinandosi alla transition region, e poi man mano, all'AC region, la densità di corrente si distribuisce in modo diverso: le cariche tendono ad accumularsi tutte verso la superficie del conduttore, come illustrato sempre in Figura 3.2. Sempre quest'ultima figura mostra come la distribuzione di carica J nella sezione del conduttore varia tra DC region e AC region. In transition region si avrà una via di mezzo. Sia in DC region che in AC region l'area in azzurro rappresenta l'area occupata dalle cariche all'interno della sezione del conduttore.

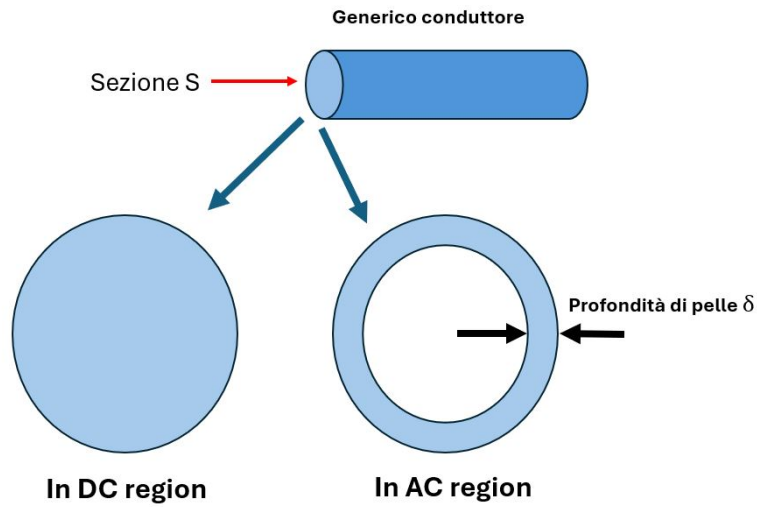


Figura 3.2: Distribuzione di carica all'interno della sezione S di un generico conduttore in DC region e in AC region.

In AC region e in transition region la densità di corrente J non è più uniforme e diminuisce man mano che ci si allontana dalla superficie del conduttore (avvicinandosi quindi al centro della sezione). Nell'immagine sopra, la **profondità di pelle (detta skin-depth) δ** rappresenta lo spessore di volume rispetto alla superficie del conduttore ancora occupato dalle cariche. La skin-depth è definita come:

$$\delta = \sqrt{\frac{2}{\omega\sigma\mu_0\mu_r}} \quad (3.1)$$

dove:

- ω in rad/s è la pulsazione angolare associata alla frequenza f in Hz, cioè $\omega = 2\pi f$;
- σ è la conducibilità del conduttore in S/m ;
- $\mu_0 = 4\pi \cdot 10^{-7} H/m$ è la permeabilità del vuoto;
- μ_r è la permeabilità relativa del conduttore, adimensionale.

al crescere della frequenza, la resistenza in questo modo crescerà perchè l'area, ovvero la sezione S (in azzurro, in Figura 3.2) entro la quale la corrente può scorrere è diminuita. Infatti:

$$R = \frac{1}{\sigma} \frac{l}{S} \quad (3.2)$$

L'induttanza in DC region rimane costante per via del fatto che il campo magnetico generato dalla corrente che scorre nel conduttore è pressochè statico. All'aumentare della frequenza, questo campo magnetico inizia ad oscillare inducendo altre correnti. Ad alte frequenze c'è meno interazione del campo magnetico con l'interno del conduttore e per questo motivo l'induttanza diminuisce. Il campo magnetico infatti continua ad interagire in gran parte con la superficie del conduttore.

Vengono definite 2 frequenze che delimitano rispettivamente la DC region, la transition region (f_{DC}) e la AC region (f_{AC}). Definito con d lo spessore del conduttore, f_{DC} rappresenta la minima frequenza che permette di entrare in transition region, cioè dove si inizia ad avere skin-effect e tale che $\delta < d$. In assenza di skin-effect, ovvero in DC region, $\delta = d$. Per convenzione f_{AC} rappresenta la più piccola frequenza che permette di entrare in AC region e tale che $\delta < d/3$. Matematicamente f_{DC} e f_{AC} sono ricavate come segue:

$$\delta = \sqrt{\frac{2}{2\pi f \sigma \mu_0 \mu_r}} < d \quad \Rightarrow \quad f > \frac{1}{\pi \sigma d^2 \mu_0 \mu_r} = f_{DC} \quad (3.3)$$

$$\delta = \sqrt{\frac{2}{2\pi f \sigma \mu_0 \mu_r}} < \frac{d}{3} \quad \Rightarrow \quad f > \frac{9}{\pi \sigma d^2 \mu_0 \mu_r} = f_{AC} \quad (3.4)$$

$$f_{DC} = \frac{1}{\pi \sigma d^2 \mu_0 \mu_r}; \quad f_{AC} = 9f_{DC} \quad (3.5)$$

Quindi, come si può evincere dalla Figura 3.1, per $f < f_{DC}$ ci si trova in DC region, per $f_{DC} < f < f_{AC}$ si è in transition region e per $f > f_{AC}$ si è in AC region.

Detto che in DC region R ed L sono costanti, che in AC region R cresce con la radice di f ed L risulta costante, c'è ancora da chiarire come Q3D calcola i due parametri in transition region. Per fare ciò si serve delle soluzioni trovate in DC e AC region. Q3D stima R ed L utilizzando il modello circuitale mostrato nella figura seguente interpolando le due soluzioni in DC e AC region.

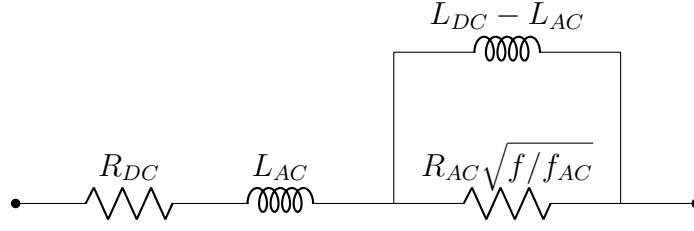


Figura 3.3: Circuito equivalente per stimare R ed L parassite in transition region. Questo modello circuitale ha un correct limiting behavior.

Quello nella figura di sopra, è un modello interpolatorio. All'interno della transition region l'impedenza vale:

$$Z(\omega) = R(\omega) + j\omega L(\omega); \quad \omega = 2\pi f \quad (3.6)$$

dalla quale Q3D estrae poi il valore di $R(\omega)$ ed $L(\omega)$. Di seguito sono riportati i calcoli matematici che ne stimano il valore teorico.

Per prima cosa, per alleggerire la notazione, poniamo:

$$\begin{aligned} R_x &= R_{AC} \sqrt{f/f_{AC}} \\ L_x &= L_{DC} - L_{AC} \end{aligned} \quad (3.7)$$

quindi avremo:

$$Z(\omega) = R_{DC} + j\omega L_{AC} + \frac{R_x \cdot j\omega L_x}{R_x + j\omega L_x} \quad (3.8)$$

moltiplichiamo il terzo addendo sopra e sotto per $R_x - j\omega L_x$ per eliminare la j a denominatore:

$$\begin{aligned} Z(\omega) &= R_{DC} + j\omega L_{AC} + \frac{R_x \cdot j\omega L_x}{R_x + j\omega L_x} \cdot \frac{(R_x - j\omega L_x)}{(R_x - j\omega L_x)} = \\ &= R_{DC} + j\omega L_{AC} + \frac{j\omega L_x R_x^2 + \omega^2 L_x^2 R_x}{R_x^2 + \omega^2 L_x^2} \end{aligned} \quad (3.9)$$

mettiamo ora assieme parte reale e immaginaria:

$$Z(\omega) = R_{DC} + \frac{\omega^2 L_x^2 R_x}{R_x^2 + \omega^2 L_x^2} + j\omega \left(L_{AC} + \frac{L_x R_x^2}{R_x^2 + \omega^2 L_x^2} \right) \quad (3.10)$$

Nell'equazione (4.2) sostituiamo:

$$\begin{aligned} R(\omega) &= R_{DC} + \frac{\omega^2 L_x^2 R_x}{R_x^2 + \omega^2 L_x^2} \\ L(\omega) &= L_{AC} + \frac{L_x R_x^2}{R_x^2 + \omega^2 L_x^2} \end{aligned} \quad (3.11)$$

sostituendo R_x e L_x definiti nella coppia di equazioni (3.7) nel sistema (3.11) si ottiene che in transition region si ha:

$$\begin{aligned} R(\omega) &= R_{DC} + \frac{\omega^2 (L_{DC} - L_{AC})^2 R_{AC} \sqrt{f/f_{AC}}}{R_{AC}^2 (f/f_{AC}) + \omega^2 (L_{DC} - L_{AC})^2} \\ L(\omega) &= L_{AC} + \frac{(L_{DC} - L_{AC}) R_{AC}^2 (f/f_{AC})}{R_{AC}^2 (f/f_{AC}) + \omega^2 (L_{DC} - L_{AC})^2} \end{aligned} \quad (3.12)$$

dove, è bene ricordare che $\omega = 2\pi f$. Per questo motivo, per far comparire la frequenza f , si possono fare delle semplificazioni finali a numeratore e denominatore di $R(\omega)$ e $L(\omega)$ rispettivamente:

$$\begin{aligned} R(\omega) &= R_{DC} + \frac{(2\pi)^2 f (L_{DC} - L_{AC})^2 R_{AC} \sqrt{f/f_{AC}}}{R_{AC}^2 (1/f_{AC}) + (2\pi)^2 f (L_{DC} - L_{AC})^2} \\ L(\omega) &= L_{AC} + \frac{(L_{DC} - L_{AC}) R_{AC}^2 (1/f_{AC})}{R_{AC}^2 (1/f_{AC}) + (2\pi)^2 f (L_{DC} - L_{AC})^2} \end{aligned} \quad (3.13)$$

3.1.2 Calcolo di R_{DC} e L_{DC} con Q3D

Q3D calcola le matrici delle resistenze R_{DC} in DC-region seguendo questi step:

1. Il volume del conduttore è diviso in mesh (tetraedri) utilizzando il metodo FEM;
2. Applicazione di una corrente pari a 1A sul terminale di source interessato, mentre gli altri terminali di source sono posti in circuito aperto;
3. Calcolo della densità di corrente volumica \vec{J} come:

$$\vec{J} = \sigma \vec{E} = -\sigma \nabla \Phi \quad (3.14)$$

dove Φ è potenziale elettrostatico associato al campo elettrico \vec{E} . In condizioni di DC si ha che la quantità di carica che lascia una quantità infinitesimale di spazio all'interno del materiale conduttore, deve essere uguale alla quantità di carica che scorre in quella stessa quantità di spazio. Cioè la densità volumica di carica $\rho(x, y, z)$ in ogni regione del conduttore rimane costante nel tempo. Ciò è modellizzato dalla seguente relazione:

$$\frac{\partial \rho(x, y, z)}{\partial t} = \nabla \cdot \vec{J} = 0 \quad (3.15)$$

Sostituendo \vec{J} definito nell'equazione (3.14) all'interno dell'equazione (3.15) si avrà che:

$$\nabla \cdot \sigma \nabla \Phi = 0 \quad (3.16)$$

Una volta trovato Φ , e quindi $\vec{J} = -\sigma \nabla \Phi$, si troverà la potenza persa in DC (Ohmic power loss in DC) nel conduttore come:

$$P = \int_V \frac{1}{\sigma} \left(\vec{J}_n(\vec{x}') \cdot \vec{J}_n(\vec{x}') \right) dV = R_{nn} I_n^2 \quad (3.17)$$

ma, come detto prima, si è posta la corrente $I_n = 1A$ che eccita l'n-esimo terminale di source che produce una densità di corrente volumica \vec{J}_n . Quindi si ha, in generale, che le **self-resistances** R_{nn} saranno definite come :

$$R_{nn} = \int_V \frac{1}{\sigma} \left(\vec{J}_n(\vec{x}') \cdot \vec{J}_n(\vec{x}') \right) dV \quad (3.18)$$

mentre le **resistenze mutue** R_{mn} saranno:

$$R_{mn} = \int_V \frac{1}{\sigma} \left(\vec{J}_m(\vec{x}') \cdot \vec{J}_n(\vec{x}') \right) dV \quad (3.19)$$

Invece per la matrice delle induttanze L_{DC} , supponendo sempre che l'n-esimo source sia eccitato con una corrente di 1A, Q3D calcola il vettore potenziale magnetico \vec{A}_n come:

$$\vec{A}_n(\vec{x}) = \int_V \frac{\mu_0 \vec{J}_n(\vec{x}')}{4\pi \|\vec{x} - \vec{x}'\|} dV \quad (3.20)$$

dove \vec{x}' rappresenta un certo source mentre \vec{x} rappresenta un certo punto di osservazione all'interno della struttura.

Poi Q3D calcola l'energia associata come segue:

$$W_M = \frac{1}{2} \int_V \vec{A}_n(\vec{x}) \cdot \vec{J}_n(\vec{x}) dV = \frac{1}{2} L_{nn} I_n^2 \quad (3.21)$$

ma si era posto $I_n = 1A$, perciò la generica **self-L** del source n è ricavata come:

$$L_{nn} = 2W_M = \int_V \vec{A}_n(\vec{x}) \cdot \vec{J}_n(\vec{x}) dV \quad (3.22)$$

per **L mutue** tra due differenti sources m ed n invece, in maniera analoga si ha:

$$L_{mn} = \int_V \vec{A}_m(\vec{x}) \cdot \vec{J}_n(\vec{x}) dV \quad (3.23)$$

Sia per le R che per le L in DC-region, quanto appena illustrato viene ripetuto dal simulatore per ogni terminale di source, ponendo a turno per ogni terminale di source una corrente di test pari ad 1A e mantenendo in circuito aperto gli altri sources, fino a quando la soluzione è calcolata per tutti i sources.

3.1.3 Calcolo di R_{AC} e L_{AC} con Q3D

Per il calcolo delle resistenze in AC-region (dove è presente il fenomeno dello skin-effect) il conduttore è diviso in mesh a forma di tetraedro che caratterizzano il FEM.

Viene poi applicata una tensione pari a 1V sul source interessato e vengono posti in corto circuito tutti gli altri terminali.

Le **self-R** sono calcolate come:

$$R_{nn} = Re \left[\int_S Z_s(\omega) \left(\vec{K}_n(\vec{x}')^* \cdot \vec{K}_n(\vec{x}') \right) dS \right] \quad (3.24)$$

mentre per le **R mutue** si ha:

$$R_{mn} = Re \left[\int_S Z_s(\omega) \left(\vec{K}_m(\vec{x}')^* \cdot \vec{K}_n(\vec{x}') \right) dS \right] \quad (3.25)$$

dove \vec{K}_n è la distribuzione di corrente superficiale e $Z_s(\omega)$ è l'impedenza superficiale del conduttore riguardante l'effetto pelle:

$$Z_s(\omega) = \sqrt{\frac{j\omega\mu_0}{\sigma}} \quad (3.26)$$

Le **self-L** e le **L mutue** sono rispettivamente calcolate come integrali di superficie:

$$L_{nn} = \int_S \vec{A}_n(\vec{x}) \cdot \vec{K}_n(\vec{x}) dS \quad (3.27)$$

$$L_{mn} = \int_S \vec{A}_m(\vec{x}) \cdot \vec{K}_n(\vec{x}) dS \quad (3.28)$$

3.1.4 Calcolo di C e G con Q3D

Per calcolare le matrici delle C e G Q3D divide la superficie dei conduttori e dei dielettrici in mesh triangolari.

Q3D poi applica una tensione di 1V ad una delle net e 0V alle altre.

Viene poi calcolata la distribuzione di carica nelle superfici del conduttore e le cariche confinate nelle superfici del materiale dielettrico. Queste cariche hanno delle condizioni al contorno ben precise.

Tutte le cariche nella superficie del conduttore sono sommate per costruire una colonna della matrice delle capacità e delle conduttanze. La parte reale delle carica equivale alla capacità e la parte immaginaria rappresenta la conduttanza.

L'intero processo è ripetuto, ponendo ogni net a 1V a turno mentre le altre net sono mantenute a 0V, fino a che la soluzione non è ottenuta per ogni terminale.

Nella superficie S , il potenziale elettrostatico Φ relativo alla distribuzione superficiale di carica ρ è dato da:

$$\Phi(\vec{x}) = \int_S \frac{\rho(\vec{x}')}{4\pi\epsilon\|\vec{x} - \vec{x}'\|} dS \quad (3.29)$$

da cui si ricava poi:

$$I = (G + j\omega C)V = j\omega Q \quad \Rightarrow \quad \frac{Q}{V} = C - j\frac{G}{\omega} \quad (3.30)$$

dove le capacità e le conduttanze sono rispettivamente:

$$C_{mn} = Re \left[\frac{Q_{mn}}{V_n} \right]; \quad G_{mn} = -\omega \cdot Im \left[\frac{Q_{mn}}{V_n} \right] \quad (3.31)$$

3.2 Simulazioni onda quadra in ingresso

Si è visto fin qui il comportamento degli elementi parassiti che costituiscono il circuito RLCG per ogni singola componente in frequenza. Eccitando quindi il

circuito con un segnale sinusoidale ad una singola frequenza si è in grado di ottenere il circuito RLCG corrispondente.

Cosa accade se invece si dà come ingresso un onda quadra, in cui si hanno più componenti in frequenza?

Ci si è chiesto questo perchè i circuiti analizzati in questo studio saranno stimolati anche con segnali ad onda quadra. Considerando un'onda quadra, che è una funzione dispari, tra -1V e 1V con valor medio nullo, si ha che ne dominio del tempo può essere scomposta in Serie di Fourier come:

$$f(t) = \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{\sin([2k - 1]2\pi ft)}{2k - 1} \quad (3.32)$$

con $2k - 1 = 1, 3, 5, \dots$ in modo tale da avere solo le armoniche dispari. Essendo l'onda quadra una funzione dispari, le sue armoniche pari sono nulle. In questo test ci si è fermati alla nona armonica, tale che $2k - 1 = 9$ (per $k = 4$). Invece f rappresenta la frequenza dell'onda quadra (e dell'armonica fondamentale). Le frequenze delle armoniche analizzate sono quindi 2MHz, 6MHz, 10MHz, 14MHz, 18MHz.

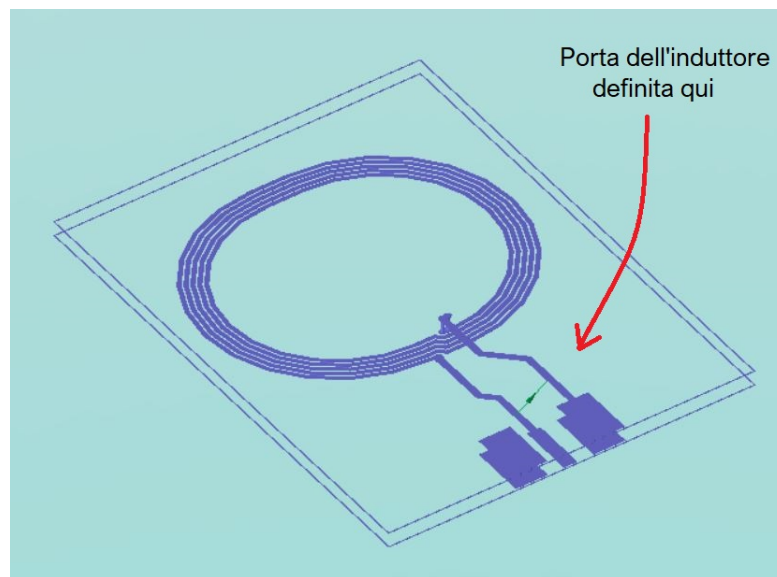


Figura 3.4: Layout dell'induttore simulato su EMPro.

Le impostazioni settate per effettuare la simulazione elettromagnetica dell'induttore sono state:

1. Scelta delle impostazioni relative alle mesh;
2. Assegnazione dei materiali;

3. Definizione delle porte;
4. Scelta del segnale in ingresso da iniettare su ciascuna porta (prima sinusoidale e poi onda quadra);
5. Scelta del frequency plan (start e stop frequency);
6. Scelta dei parametri S;
7. Scelta del simulatore: FDTD o FEM.

Per eseguire la simulazione elettromagnetica su EMPro dell'induttore è stato utilizzato il simulatore **FDTD**, la cui teoria è descritta nella Sezione 2.3.

Inizialmente è stata iniettata alla porta dell'induttore una sinusoidale avente una frequenza corrispondente alle componenti in frequenza dell'onda quadra considerata. Quindi prima si è iniettata una sinusoidale a 2MHz, poi una a 6MHz, poi una a 10MHz e così via, fino a 14MHz.

Successivamente si è iniettata un'onda quadra compresa tra -1V e 1V a 2MHz. Si sono poi confrontati i risultati in termini di parametri S tra la sinusoidale ad una frequenza e l'armonica dell'onda quadra a quella stessa frequenza. Cioè confrontando i parametri S della sinusoidale a 2MHz con l'armonica a 2MHz, successivamente la sinusoidale a 6MHz con l'armonica a 6MHz e così via.

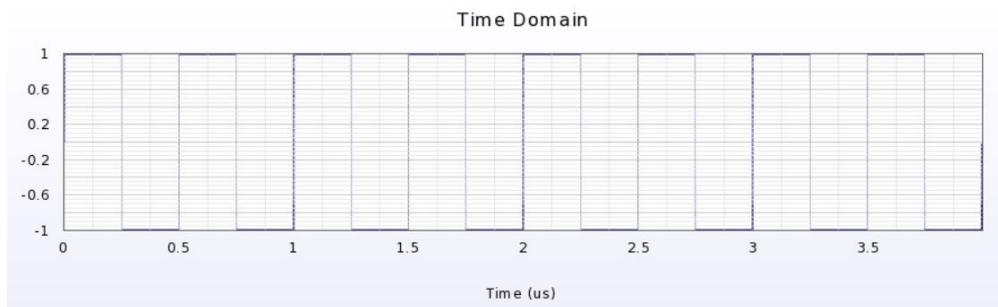


Figura 3.5: Onda quadra a 2MHz in ingresso alla porta dell'induttore.

Nel seguito sono riportati i risultati ottenuti nel caso di ingresso sinusoidale (a sinistra) e nel caso di onda quadra in ingresso all'induttore. Nelle immagini seguenti viene confrontato S_{11} ottenuto con una sinusoidale ad una certa frequenza f_x con l' S_{11} dell'armonica dell'onda quadra sempre alla frequenza f_x .

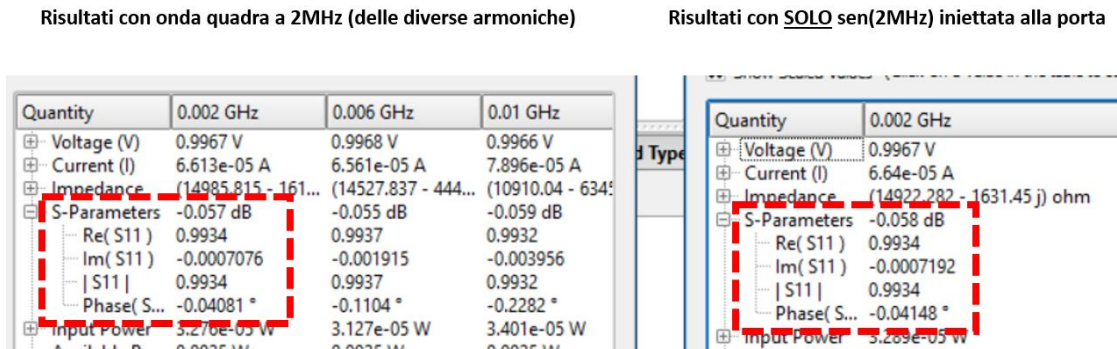


Figura 3.6: Confronto di S_{11} della portante a 2MHz con quello della singola sinusoidale a 2MHz da sola alla porta dell'induttore su EMPro.

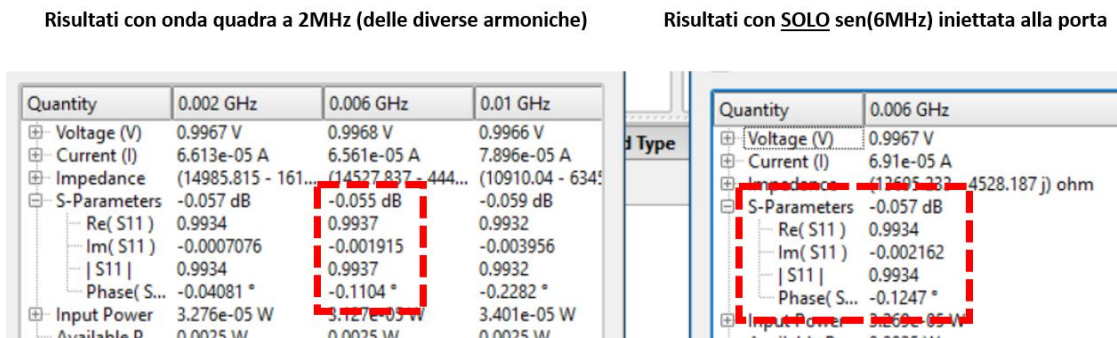


Figura 3.7: Confronto di S_{11} dell'armonica a 6MHz con quello della singola sinusoidale a 6MHz da sola alla porta dell'induttore su EMPro.

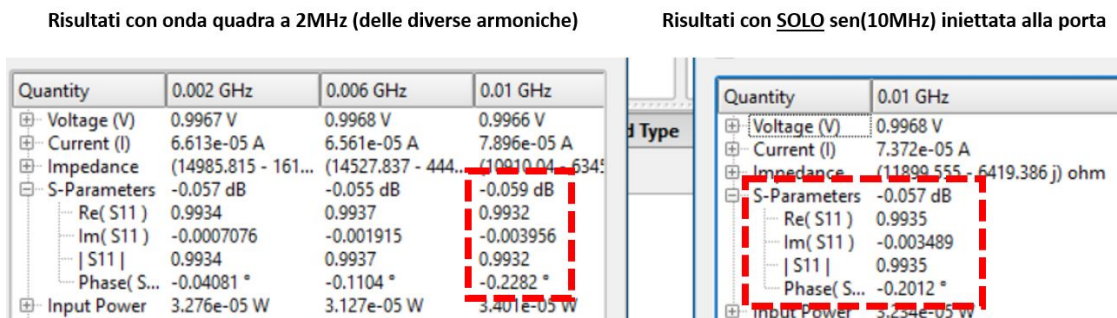


Figura 3.8: Confronto di S_{11} dell'armonica a 10MHz con quello della singola sinusoidale a 10MHz da sola alla porta dell'induttore su EMPro.

Risultati con onda quadra a 2MHz (delle diverse armoniche)

Risultati con SOLO sen(14MHz) iniettata alla porta

Quantity	0.002	0.006	0.01 G	0.014 GHz	0.018 GHz
Voltage (V)	0.99...	0.99...	0.99...	0.9968 V	0.9969 V
Current (I)	6.61...	6.56...	7.89...	7.713e-05 A	9.547e-05 A
Impedance	(149...	(145...	(109...	(10766.402 - 7...	(6693.748 - 8...
S-Parameters	-0.0...	-0.0...	-0.0...	-0.056 dB	-0.053 dB
Re(S11)	0.99...	0.99...	0.99...	0.9936	0.9939
Im(S11)	-0.0...	-0.0...	-0.0...	-0.004253	-0.007305
S11	0.99...	0.99...	0.99...	0.9936	0.9939
Phase(S...	-0.0...	-0.1...	-0.2...	-0.2452 °	-0.4211 °
Input Power	3.27...	3.12...	3.40...	5.205e-05 W	3.05e-05 W

Quantity	0.014 GHz
Voltage (V)	0.9968 V
Current (I)	8.166e-05 A
Impedance	(9710.383 - 7396...
S-Parameters	-0.057 dB
Re(S11)	0.9935
Im(S11)	-0.004932
S11	0.9935
Phase(S...	-0.2844 °
Input Power	5.257e-05 W

Figura 3.9: Confronto di S_{11} dell'armonica a 14MHz con quello della singola sinusoidale a 14MHz da sola alla porta dell'induttore su EMPro.

Le figure dalla 3.6 alla 3.9 mostrano come i parametri S_{11} , confrontando anche la parte reale e immaginaria, siano gli stessi tra il caso in cui si inietta in ingresso all'induttore una sinusoidale ad una determinata frequenza f_x e il caso in cui si inietta l'onda quadra: i valori di S_{11} alla frequenza f_x dell'armonica sono molto vicini a quelli ottenuti alla frequenza f_x della sinusoidale.

Si può quindi notare che come l'onda quadra è una sovrapposizione di segnali sinusoidali, anche per quanto riguarda i parametri S in frequenza, essi risultano essere una combinazione lineare dei valori alle singole frequenze. **Questo dimostra che se si inietta un segnale sinusoidale, per ogni parametro S, per ogni frequenza, esiste un circuito equivalente RLCG. Se si inietta un'onda quadra, il circuito RLCG totale è una sovrapposizione dei circuiti RLCG delle singole frequenze delle armoniche.**

Capitolo 4

Analisi nel tempo con parametri variabili in frequenza in Cadence

In questo capitolo, sono illustrati gli step principali seguiti per ottenere i modelli equivalenti di un dato circuito, utilizzati in questo studio di Tesi; ovvero i parametri S, il circuito equivalente RLCG ad una certa frequenza e la broadband netlist. Questi modelli, che descrivono il comportamento degli elementi parassiti ottenuti dalla simulazione elettromagnetica, sono utilizzati per effettuare le cosimulazioni in transient su SPICE.

4.1 SPICE: simulazione transient dei parametri S, della broadband netlist e del circuito equivalente RLCG ad una sola frequenza

Come scritto in precedenza, l'ambiente SPICE è stato utilizzato per effettuare le cosimulazioni nel dominio del tempo dei circuiti presi in esame. Sono state effettuati 3 diversi tipi di cosimulazione:

1. Cosimulazione su Eldo con i parametri S definiti su tutto il range di frequenza;
2. Cosimulazione su Eldo con la broadband netlist ottenuta a partire dai parametri S;
3. Cosimulazione su Eldo con il circuito RLCG equivalente definito ad una certa frequenza;

I risultati ottenuti con queste 3 tipologie di cosimulazione sono stati confrontati tra di loro.

Per un dato circuito, per leggere i parametri S (cioè il file touchstone che li contiene), su SPICE è disponibile il componente denominato **nport**: si tratta di un blocchettino che legge i file touchstone di un qualsiasi dispositivo a n porte. Questo componente è dotato di pin collegabili con qualsiasi altro componente circuitale, esso legge il file touchstone dei parametri S definito in un certo range di frequenza. Le impostazioni più importanti e significative di questo componente sono il **numero di porte**, il **file touchstone .snp**, l'**algorithm index**. In particolare, l'algorithm index presenta 2 diverse possibili scelte: 0 o 1, dove 0 stà per algoritmo CPF e 1 stà per algoritmo DSP. Ciascuna delle due opzioni è brevemente decritta:

1. **CPF (Complex pole fitting)**: in questo metodo i parametri S sono rappresentati secondo la tecnica del vector fitting descritta nella sezione 4.5. Si ha quindi una rappresentazione in termini di poli complessi e residui. L'algoritmo CPF si presta per **parametri S aventi step di campionamento logaritmico**.
2. **DSP (Digital signal processing)**: in questo approccio i dati definiti nel dominio della frequenza vengono trasformati nel dominio del tempo effettuando la FFT inversa.

In generale tutti e 2 gli algoritmi accettano qualsiasi tipo di passo di campionamento (lineare, logaritmico o altro) dei parametri S: va comunque precisato che il **DSP** richiede uno **step di campionamento lineare** con $K^2 + 1$ punti a partire dalla frequenza 0Hz. Se tale condizione non è soddisfatta, l'algoritmo DSP effettua una interpolazione dei dati per rispettarla e quindi ricordarsi allo step lineare appena menzionato [20]. Nella figura sottostante è riportato il simbolo grafico del component nel caso di un dispositivo a 4 porte.

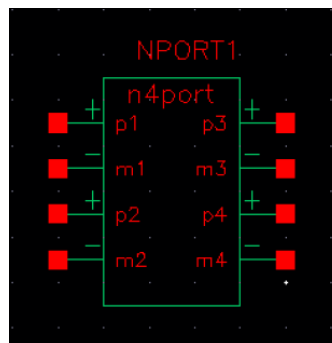


Figura 4.1: Blocchettino nport, esempio con 4 porte.

Per il secondo e terzo tipo di cosimulazione descritto, è necessario importare su SPICE rispettivamente la **broadband netlist (in formato hspice)** e la **netlist RLCG ad una frequenza (.cir)**, definendo un file di tipo **text.netlist**. Una volta salvato il file, bisognerà scegliere la forma grafica del component e come posizionare i terminali di ciascuna porta. Una volta fatto ciò sarà automaticamente creato il component che sarà disponibile in libreria e utilizzabile in qualsiasi schematico di Cadence.

La broadband netlist di un determinato circuito è ottenuta sempre a partire dai suoi parametri S. Per ottenere la broadband netlist (valida su tutto il range di frequenza scelto) si è visto che è possibile utilizzare tre differenti tool:

- Broadband SPICE Model Generator presente su ADS;
- Broadband export tool su Q3D (in ANSYS Electronics Desktop);
- Matlab, con la funzione `rational()`.

Nel seguito sono descritti nel dettaglio questi differenti modi per la generazione delle broadband netlist.

Inoltre sono presentati gli step necessari per ottenere i parametri S e le broadband netlist utilizzando ADS, Q3D e Matlab. Come scritto in precedenza, Q3D, a differenza di ADS, è stato utilizzato per generare anche le netlist RLCG ad una determinata frequenza.

Nel Capitolo 5 è mostrato come il componente dei parametri S e le broadband netlist importate sono utilizzate su SPICE per effettuare la cosimulazione transient nel caso specifico di un convertitore Buck.

4.2 ADS: step per ottenere il file touchstone dei parametri S e la broadband netlist

Prendendo come esempio il layout di un semplice induttore in tecnologia PCB, una volta ultimata la simulazione elettromagnetica, su ADS si è costruito un testbench apposito che permette di ottenere il file dei parametri S sul range di frequenza scelto, come si può notare nelle due figure sottostanti. Con il costrutto SPOutput nello schematico si possono impostare diversi parametri che contraddistinguono il file dei parametri S che si intende generare: i più importanti sono il formato (modulo/fase o parte reale/immaginaria), il tipo di file (touchstone o altro) e il range di frequenza. Si è scelto il formato touchstone perchè è quello più diffuso e comodo da utilizzare.

Il simulatore, nel costruire il file dei parametri S, legge i risultati della simulazione elettromagnetica.

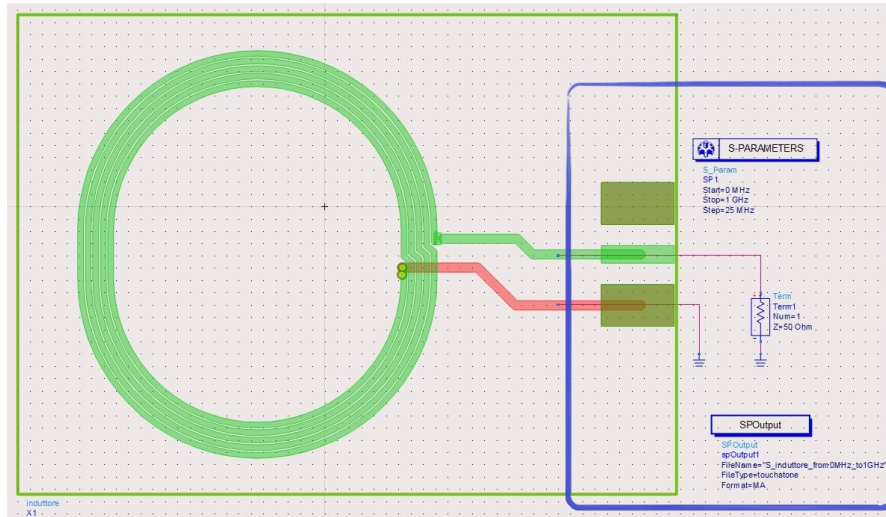


Figura 4.2: Testbench costruito su ADS per generare il file touchstone dei parametri S: Layout dell'induttore + costrutti dedicati.

Una volta ottenuto il file touchstone dei parametri S, quest'ultimo lo si utilizza pr ottenere la netlist equivalente valida anch'essa su tutto il range di frequenza di interesse (la broadband netlist). Il tool in ADS che permette di generare questa broadband netlist in formato SPICE è il **Broadband SPICE Model Generator**. Se gli si dà in ingresso il file touchstone dei parametri S, genera in uscita il broadband circuit in un formato a scelta (spice2, spice3, hspice, pispice, spectre, ads). Il file dei parametri S che si da in ingresso al tool deve avere impedenza di riferimento di 50Ω. Nelle due immagini seguenti sono mostrati gli screenshot riguardanti le due sezioni del tool contenenti le relative impostazioni:

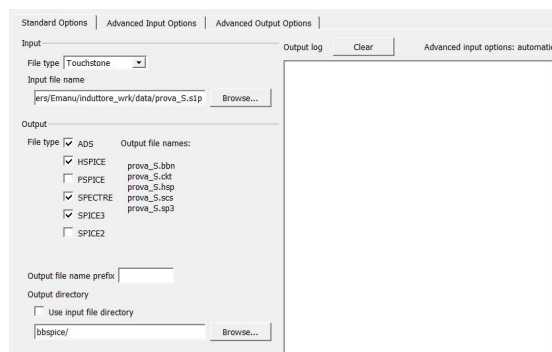


Figura 4.3: Opzioni standard del Broadband SPICE Model Generator su ADS.

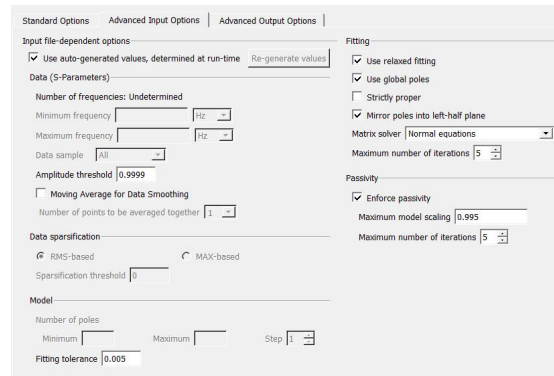


Figura 4.4: Opzioni di input avanzate del Broadband SPICE Model Generator su ADS.

Settando opportunamente le impostazioni del Broadband SPICE Model Generator si otterrà un broadband circuit avente i parametri S il più possibile fedeli a quelli di ingresso. Il Broadband SPICE model generator esegue un fitting dei parametri S ricevuti in ingresso. Il metodo utilizzato è quello del **vector fitting**, la cui teoria è descritta nella sezione 4.5. Per un certo parametro S_{ij} , il fitting model generato ha la seguente forma:

$$S'_{ij}(s) = \sum_{n=1}^N \frac{r_n}{s - p_n} = \frac{r_1}{s - p_1} + \frac{r_2}{s - p_2} + \dots + \frac{r_N}{s - p_N} \quad (4.1)$$

dove r_n sono i residui e p_n sono i poli.

$S'_{ij}(s)$ deve il più possibile ricalcare l' S_{ij} originario.

La broadband netlist è un circuito equivalente formato da generatori pilotati, resistenze e capacità che modelizza il circuito che poi si vuole ottenere: quello RLCG su tutto il range di frequenza. Diventa importante precisare che con Q3D si otteneva un circuito RLCG valido ad una sola frequenza e la broadband netlist.

Segue ora una spiegazione riguardante le impostazioni principali configurabili nel Broadband SPICE Model Generator tool mostrate in Figura 4.4 [21]:

- **Use auto-generated values determined at run-time:** quando questa impostazione viene settata, alcuni opzioni presenti vengono disabilitate, perchè durante la generazione della netlist, sono calcolate in base al contenuto del file dei parametri S ricevuto in ingresso. Le uniche opzioni che rimangono ancora modificabili sono quelle mostrate in Figura 4.4. Se l'impostazione invece viene de-selezionata, le altre impostazioni che prima non erano modificabili

diventano modificabili e vengono utilizzate durante la generazione della netlist. Il bottone **Re-generate values** se premuto, premette di riportare tutte le impostazioni dipendenti dal file dei parametri S in ingresso al loro valore di default.

- **Minimum frequency e Maximum frequency:** questi due parametri rappresentano rispettivamente la minima e la massima frequenza contenuta nel file dei parametri S in ingresso. Volendo, si possono modificare a vanno rispettati alcuni vincoli:
 - La frequenza minima scelta non deve essere minore della frequenza minima del file di ingresso dei parametri S;
 - La frequenza massima scelta non deve essere maggiore della frequenza massima del file di ingresso dei parametri S;
 - La frequenza minima scelta non deve essere maggiore della frequenza massima scelta.
- **Data sample:** permette di specificare il numero di campioni da prendere in considerazione durante la generazione del rational model. Le scelte possibili sono: **All**, **ReducedX2**, **ReducedX5**, **ReducedX10**, **ReducedX20** e **ReducedX50**. Il valore ottimale dipende anche dal tipo di file dei parametri S in ingresso: nel caso di touchstone il valore consigliato è **ReducedX20**. Ridurre il numero di punti è vantaggioso nel caso in cui il file dei parametri S contenga un numero alto di campioni e un numero limitato di poli. Questo quindi può aumentare la velocità di calcolo senza però diminuire l'accuratezza. Se dovesse capitare poi che il numero di campioni scartato sia esagerato, il Broadband SPICE Model Generator in automatico sceglierà un diverso numero di punti da ridurre. Avvisando con un warning.
- **Amplitude threshold:** rappresenta una soglia che permette di pre-imporre il fatto che la netlist generata debba essere un dispositivo passivo cioè $|S| < 1$. I valori consigliati, se si vuole imporre ciò sono 1 oppure qualcosa di vicino (ad esempio 0.9999). Se invece si vuole ignorare questo parametro, è consigliato porre l'amplitude threshold ad un valore grande, come 100 per esempio. Nel caso dell'analisi dell'induttore e del buck, limitare l'amplitude threshold a 1 non ha mai portato ad ottenere delle netlist precise, mentre porre amplitude threshold a 100 ha permesso di ottenere una netlist che ricalca molto accuratamente i parametri S in ingresso (cioè con errori percentuali su R e L bassissimi). Porre amplitude threshold a valori alti permette di avere meno vincoli e più gradi di libertà nella generazione della netlist stessa. In più per i casi di induttore e Buck si è riscontrato che i tempi di calcolo sono molto più ridotti rispetto al caso con amplitude threshold a 1.

- **Moving average for data smoothing:** si rivela un'impostazione utile nel caso i parametri S in ingresso siano frutto di misure. Si rivela quindi utile per file contenenti più rumore che simulation data. Questa impostazione, se attivata, permette di modificare i dati in ingresso in modo da appianarli per poter diminuire il rumore da modellizzare durante la creazione della netlist. Troppo rumore infatti provocherebbe la generazione di netlist aventi dimensioni più grandi. Attivando questa opzione il tool permette di scegliere di quanti punti calcolare la media insieme. **Number of points to be averaged together** deve essere un numero positivo dispari.
- **Data sparsification:** permette di scegliere quali parametri S utilizzare dipendentemente dal loro RMS o dal loro massimo. La **sparsification threshold** permette di imporre una soglia per scegliere quali parametri S prendere e quali scartare per generare la netlist. I parametri S scartati vengono posti a 0.
- **Number of poles:** il valore inizialmente viene calcolato in base al file dei parametri S dato in ingresso. Poi è possibile decidere altri valori per **minimum, maximum e step** nel caso in cui il processo di fitting non converga oppure se il passivity enforcement dovesse fallire.
- **Fitting tolerance:** rappresenta la radice quadrata media (RMS) permessa per il fitting error. Se il valore RMS del modello razionale fittato è minore del fitting tolerance il modello è considerato convergente. Il valore raccomandato è tra 0.001 e 0.02.
- **Use relaxed fitting:** se selezionata, questa impostazione permette di utilizzare un algoritmo di fitting leggermente diverso da quello standard che utilizza relaxation. Può essere utile selezionare o deselezionare questa impostazione nel caso in cui la procedura di fitting non converga. La teoria del **relaxed fitting** è descritta all'interno della sezione 4.5.
- **Use global poles:** se scelto, permette per tutti i parametri S, di condividere lo stesso set di poli. Selezionare questa impostazione è consigliato quando si ha a che fare con un circuito avente un numero limitato di porte (non più di 4). Se il numero di porte è sopra 4 la selezione di Use global poles permette di incrementare la velocità del fitting ma rallenta il processo di passivity enforcement.
- **Strictly proper:** attivando lo Strictly proper si impone che $N = D - 1$, dove N è il grado del polinomio al numeratore del rational model e D è il grado del polinomio al denominatore del rational model. Se invece lo Strictly proper è deselezionato si avrà che $D - 1 \leq N \leq D$. Risulta comunque consigliato attivare questa opzione se si vuole imporre la passivity a frequenze elevate.

- **Mirror poles into left-half plane:** questa impostazione, se scelta, impone che i poli non-fisici siano specchiati nel semipiano di sinistra complesso. Bisogna precisare che per ottenere un circuito passivo, i poli devono stare nel semipiano di sinistra. Se il processo di fitting fallisce, selezionare questo campo può portare alla convergenza. Questa opzione influisce anche sulla precisione della netlist nelle simulazioni SPICE in transient.
- **Matrix solver:** le scelte possibili sono **QR factorization**, **Singular value decomposition (SVD)**, **Normal equations**. Se si ha a che fare con un circuito avente molte porte e molti dati campioni è consigliato scegliere **Normal equations**. Se invece si vuole privilegiare l'accuracy a discapito della velocità di calcolo **Singular value decomposition (SVD)** risulta essere la scelta più conveniente. Se nessuna tra accuracy e velocità è importante e il numero di poli e di porte è limitato, **QR factorization** risulta essere la scelta più giusta. La teoria e il significato del **vector fitting standard (normal equations)**, della **QR factorization** e del **Singular value decomposition (SVD)** è descritta all'interno della sezione 4.5.
- **Maximum number of iterations (Fitting):** rappresenta il numero di step di ricollocamento dei poli. Il valore consigliato è tra 4 e 7. Al crescere di questo valore, aumenta il tempo di fitting e di generazione del modello.
- **Enforce passivity:** se selezionata, impone uno step ulteriore che modifica alcuni coefficienti del modello di fitting appena generato per assicurare che il modello sia passivo. Se invece si deselecta l'**Enforce passivity**, lo step che impone la passivity non viene eseguito sul modello di fitting che è stato appena generato.
- **Maximum model scaling:** questa impostazione è utile per eliminare le ultime tracce di passività del modello. Potrebbe accadere infatti che anche dopo un passivity enforcement $|S| = 1.0001$ ad una certa frequenza. Per questo motivo scalare tutto il modello per 0.9999 elimina le ultime violazioni riguardanti la passivity. Questo rappresenta l'ultimo passo riguardante il processo di passivity enforcement che il tool esegue. Se si vuole saltare questo ultimo step si può impostare un valore ≥ 1 di Maximum model scaling.
- **Maximum number of iterations (Passivity):** rappresenta il numero di cicli di passivity enforcement. Se si pone a 0, non viene forzata la passivity per il rational fitting model che viene generato.

Le impostazioni appena descritte, vanno scelte nel modo opportuno, in modo da ottenere una broadband netlist di uscita i cui parametri S ricalchino il più possibile i parametri S dati in ingresso. Bisogna trovare la combinazione delle impostazioni

che minimizzi le differenze tra modello dei parametri S in ingresso e netlist ottenuta in uscita.

In generale è bene tenere conto di alcune raccomandazioni che possono dare un aiuto nell'utilizzo del Broadband SPICE Model Generator, per ottenere in uscita un modello con la migliore precisione possibile.

Nel caso si voglia ottenere una broadband netlist di circuiti in **tecnologia PCB**, è bene seguire le seguenti raccomandazioni [22]:

1. de-selezionare l'opzione **Use auto-generated values determined at run-time** in modo da attivare le altre impostazioni più avanzate.
2. Nella sezione **Model** ⇒ **Number of poles** porre **Minimum** a 1 e lasciare **Maximum** al suo valore di default.
3. Se, sempre dentro la sezione **Model** si riduce la **Fitting tolerance** per avere una netlist accurata, è bene aumentare il massimo numero di poli **Maximum** ad un valore alto. Altrimenti il rischio è quello che la generazione della netlist fallisca per via del fatto che il broadband generator non riesce a convergere.
4. Selezionare l'opzione **Strictly proper**.
5. Nel caso il circuito in questione abbia un numero grande di porte, de-selezionare l'opzione **Mirror poles into left-half plane** per ottenere in uscita una netlist maggiormente accurata.
6. L'opzione **Enforce passivity** va ad influire sulla velocità di generazione della netlist nel caso di circuiti con un numero di porte maggiore di 50 circa. Questa opzione andrebbe abilitata solo quando la simulazione in transient (della netlist ottenuta con **Enforce passivity** disabilitato) non converge. Se si abilita l'**Enforce passivity** selezionare anche l'opzione **Use global poles**. Sempre per un numero di porta superiore a 50 circa, mantenere disabilitato **Use global poles** e attivare **Enforce passivity** la generazione della netlist potrebbe fallire.
7. Mantenere tutte le altre opzioni al loro valore di default.

Vi sono altre raccomandazioni (questa volta per circuiti più generici) che possono essere utili [23]:

1. Se diventa difficile ottenere un buon livello di accuratezza e di passività, in **Advanced Input Options** è consigliabile selezionare l'opzione **Use relaxed fitting**.

2. Almeno che non sia veramente necessario, è sconsigliato selezionare l'estensione .sp2 (SPICE2) (della broadband netlist che si vuole generare) perchè ha un livello di accuratezza inferiore all'estensione .sp3 (SPICE3). In più la generazione della netlist .sp2 richiede dei calcoli extra.
3. Per quanto riguarda l'opzione **Data sample**, sarebbe bene selezionare **ReducedX50** se si ha un numero di frequenze elevato. Il tool, nel caso, sceglie in autonomia un'opzione che permetta di avere una riduzione minore di sample nel caso in cui necessiti di un numero maggiore di campioni per effettuare i calcoli. Può essere utile precisare che l'opzione REDUCEDX generica non può essere scelta nel caso in cui il file di input fosse di tipo Momentum rational anzichè touchstone. Per i Momentum rational file è meglio scegliere AIIX.
4. Solitamente, il numero di poli nella sezione **Model** è stimato dal tool una volta che il file touchstone è stato caricato in esso. Di solito la stima effettuata dal tool è buona, ma in caso di necessità è possibile definire un numero diverso di poli, specialmente quando:
 - Il file touchstone dei parametri S contiene dati caratterizzati da rumore;
 - Minimum frequency e Maximum frequency sono state modificate rispetto a quello definito nel file touchstone. In questa situazione è meglio modificare il numero di poli in modo tale da avere tra il minimo e il massimo numero di poli una differenza larga.
5. in presenza di circuiti con un numero alto di porte e di poli i tempi di calcolo potrebbero diventare molto lunghi. Per tale ragione disattivare il **Passivity enforcement** può ridurre di molto i tempi di calcolo. Se non si è interessati ad ottenere una broadband netlist passiva è meglio deselezionare l'opzione **Use global poles**. Se si desidera vedere quanto è passiva la broadband netlist che si ottiene, è meglio attivare l'**Enforce passivity** ma porre a 0 l'opzione **Maximum number of iterations**. Se invece si ha la necessità di avere una netlist passiva (come nel caso di questo studio sui parassiti) è necessario attivare l'**Enforce passivity** e porre **Maximum number of iterations** a 5. **In questo studio di Tesi la necessità è quella di ottenere delle Broadband netlist PASSIVE.**
6. In generale, se si ha a che fare con circuiti molto grandi, e con un numero elevato di porte, è meglio scegliere le impostazioni seguendo le raccomandazioni appena illustrate o una combinazione di esse. Potrebbe essere infatti necessario effettuare diverse simulazioni del Broadband SPICE Model Generator e vedere quale sarà il setting delle impostazioni che permetterà di ottenere la netlist più accurata.

Una volta settate nel modo migliore tutte le impostazioni di input del broadband SPICE Model generator, è possibile anche scegliere le impostazioni di output,

Selezionando l'opzione **Open data display when simulation completes**, al termine della generazione del Broadband SPICE Model, ADS genera di default un template (c'è una vasta scelta di tipologie di template) dove mostra l'andamento dei parametri S (modulo e fase) in frequenza sia per il file touchstone dato in ingresso che per il Broadband SPICE Model generato. ADS fa proprio una sovrapposizione dei grafici per fare un confronto.

Questa sovrapposizione su tutto il range di frequenza dei parametri S, **non è sufficiente per capire se il modello generato ricalca abbastanza accuratamente i parametri S di ingresso**. Non ci si deve quindi fermare a guardare l'errore sui parametri S, perchè anche variazioni piccolissime possono generare differenze non trascurabili tra il modello dei parametri S in ingresso e la broadband netlist generata in uscita.

Per valutare di quanto i due modelli equivalenti si discostano tra loro, bisogna calcolare, a partire dai parametri S dei due modelli stessi, **il valore dei corrispettivi parassiti R ed L e valutare quanto è il massimo errore percentuale relativo di R e L del broadband circuit generato rispetto ai parametri S di ingresso**. Lo scopo è avere questi due errori relativi il più piccoli possibile su tutto il range di frequenza, tra tutti gli S_{ij} . Nel template su ADS, è possibile definire delle formule riguardanti i dati presenti nel dataset (in questo caso i parametri S). Per il circuito dell'induttore configurato come in Figura 4.2 a partire dai parametri S può essere calcolata l'impedenza Z_{ij} e gli errori relativi percentuali sono calcolati, partendo da un generico S_{ij} , come segue:

$$S_{ij}(\omega) \Rightarrow Z_{ij} = \text{stoz}(S_{ij}(\omega)) = R + j\omega L \quad (4.2)$$

$$R = R(\omega); \quad L = L(\omega) \quad (4.3)$$

$$\text{Max} \left(\frac{\Delta R}{R} \% \right) = \text{Max} \left(\left| \frac{R_{inmodel} - R_{bbsmodel}}{R_{inmodel}} \right| \cdot 100 \right) \quad (4.4)$$

$$\text{Max} \left(\frac{\Delta L}{L} \% \right) = \text{Max} \left(\left| \frac{L_{inmodel} - L_{bbsmodel}}{L_{inmodel}} \right| \cdot 100 \right) \quad (4.5)$$

Nell'equazione (4.2) sia per quanto riguarda il modello di parametri S in ingresso, che per il modello di broadband netlist che si genera in uscita si valuta l'impedenza Z_{ij} a partire da una funzione di ADS $\text{stoz}()$.

In (4.4) e (4.5) sono definiti rispettivamente il massimo errore relativo percentuale

sulle resistenze e sulle induttanze.

Sempre nelle equazioni (4.4) e (4.5) $R_{inmodel}$ e $L_{inmodel}$ rappresentano rispettivamente la resistenza e l'induttanza inerente al parametro S_{ij} del modello touchstone (inmodel) dato in ingresso al Broadband SPICE Model Generator. Invece $R_{bbsmodel}$ e $L_{bbsmodel}$ rappresentano la resistenza e induttanza relativa al parametro S_{ij} del modello di netlist che si ottiene in uscita che è il Broadband SPICE Model (bbsmodel). L'obiettivo è ottenere una broadband netlist il più possibile fedele ai parametri S in ingresso al Broadband SPICE Model Generator. Per ottenere ciò, bisogna minimizzare gli errori relativi percentuali massimi $Max\left(\frac{\Delta R}{R}\%\right)$ e $Max\left(\frac{\Delta L}{L}\%\right)$ valutati su tutto il range di frequenza di interesse. Per tenere questi errori più bassi possibile, e quindi aumentare l'accuratezza del modello prodotto, bisogna reiterare la procedura modificando, di volta in volta, le impostazioni di ingresso del Broadband SPICE Model Generator (mostrate in Figura 4.4) fino a che si ottengono degli errori relativi percentuali desiderati.

4.3 Q3D: step per ottenere il file touchstone dei parametri S, la broadband netlist e la netlist RLCG definita ad una frequenza

Anche su Q3D è possibile ottenere i parametri S di un circuito. Per ricavarli è necessario prima di tutto definire le porte del circuito attraverso due categorie: source e sink. Risulta necessario fare ciò affinché il simulatore decida dove iniettare i segnali che stimoleranno il circuito per effettuare la simulazione elettromagnetica e per il calcolo dei parametri S. Le porte di tipo source sono quelle porte alla quale si connette lo stimolo tensione/corrente utile per la simulazione elettromagnetica. Le porte di tipo sink sono le porte che definiscono la direzione di uscita del segnale iniettato. Se interessa solo trovare le capacità e le conduttanze non è necessario definire source e sink.

Una volta definite le source e sink ports è necessario definire la frequenza di simulazione che può essere singola o definita su un range. Il range può essere di tipo **lineare** (definendo a scelta lo step più opportuno o il n° di punti all'interno del range) o **logaritmico** (definendo il n° di punti/decade). Sia se si sceglie uno step lineare che uno logaritmico è necessario definire i due estremi del range: la frequenza di start e la frequenza di stop. Un'altra impostazione da scegliere è il tipo di sweep: **discrete o interpolating**. Nel caso di discrete sweep lo si sceglie quando ci si accontenta di definire la soluzione solo su frequenze specifiche all'interno di tutto il range. Può essere meglio usarlo quando sono necessarie solo

poche frequenze per rappresentare in modo preciso e accurato i risultati in quel determinato range di frequenza. Invece l'interpolating sweep permette di calcolare i risultati finali su tutto il range di frequenza. Conviene scegliere interpolating sweep quando si ha a che fare con range di frequenza larghi [10].

Una volta definite tutte queste impostazioni si è pronti per far partire la simulazione elettromagnetica. Una volta terminata la simulazione sarà possibile estrarre ed esportare il file dei parametri S definiti sul range di frequenza scelto. Questo file dei parametri S avrà una certa sintassi. Prima di generarlo occorrerà scegliere il nome del file, dove salvarlo, il formato dei parametri S (modulo/fase o parte reale/immaginaria), il tipo di file (Touchstone o altri), tipo di step (lineare o logaritmico) e infine l'impedenza di riferimento di ciascuna porta (nel 99% dei casi è 50Ω). Di solito il tipo di file scelto e che è utile anche in questo studio è il file touchstone. Questi file touchstone hanno estensione .snp dove n è il numero di porte del circuito preso in esame.

Oltre che i parametri S è possibile generare anche il **modello equivalente RLCG del circuito preso in esame e valido ad una sola frequenza**. Gli step per definire il setup della simulazione sono gli stessi seguiti per i parametri S ma c'è da definire a che frequenza il circuito RLCG deve essere generato/esportato: la soluzione verrà pescata dallo sweep nel range di frequenze definito in precedenza per i parametri S.

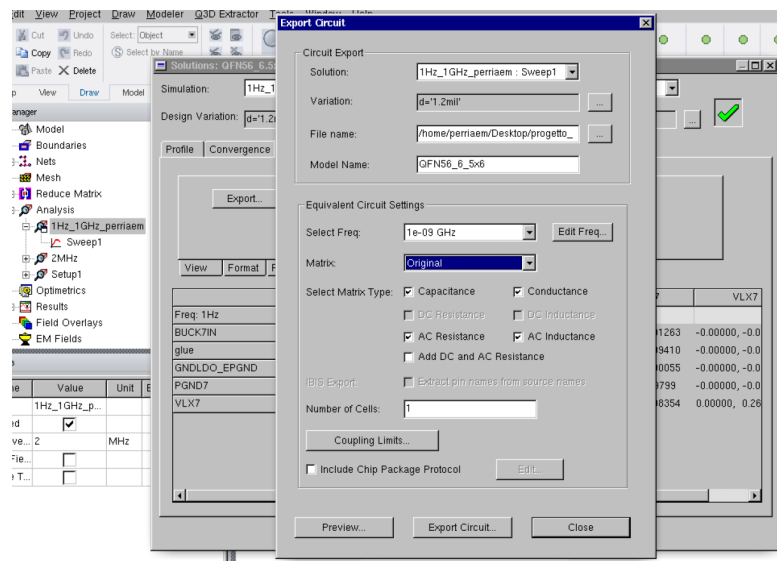


Figura 4.5: Impostazioni per la generazione del circuito RLCG ad una certa frequenza su Q3D.

La Figura 4.5 mostra come bisogna impostare la soluzione selezionando le voci **capacitance**, **conductance**, **AC resistance** e **AC inductance** per ottenere il

circuito desiderato.

Un terzo modello, equivalente a quello dei parametri S, che si può ottenere come risultato della simulazione elettromagnetica è quello del broadband circuit. Il broadband circuit è una rete definita su tutto il range di frequenza scelto ed è formata da resistenze, capacità e generatori pilotati che ricalcano il comportamento del **circuito equivalente RLCG valido a tutte le frequenze** (la cui generazione è uno degli obiettivi di questo studio di Tesi). Q3D lo crea utilizzando l'algoritmo che riguarda il metodo del **vector fitting** che ha l'obiettivo di ricalcare, fittare il modello originale dei parametri S ricavato dalla simulazione elettromagnetica. La teoria riguardante il vector fitting è riportata nella sezione 4.5. Per ottenere questa broadband netlist Q3D mette a disposizione un tool denominato **Broadband export tool**. Il principio di funzionamento risulta essere lo stesso del Broadband SPICE Model Generator di ADS. Utilizzando il broadband export tool di Q3D è possibile settare il fitting error desiderato, i tipi di algoritmi/opzioni di fitting, di passivity, il massimo numero di poli (cioè l'ordine massimo), il tipo di file (per il nostro studio si rivela comodo HSPICE).

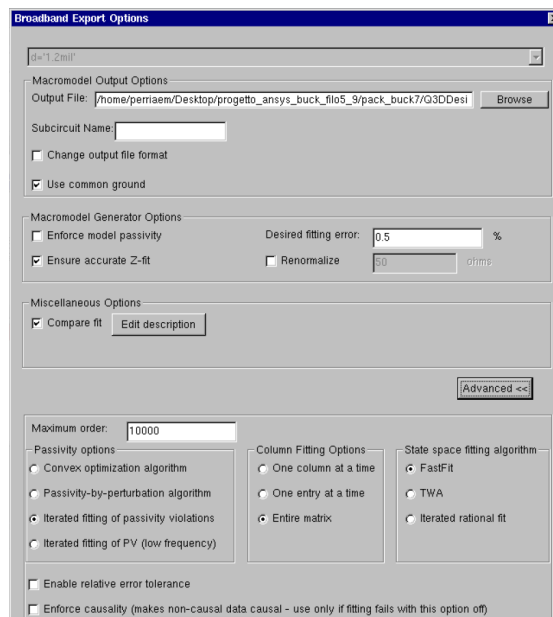


Figura 4.6: Opzioni avanzate del Broadband Generator su Q3D.

Come si può vedere dalla Figura 4.6, anche il broadband export tool permette di scegliere delle impostazioni [10]. Le più significative e quelle che è consigliabile

modificare sono:

1. **Renormalize**: di default l'impedenza di riferimento è 50Ω ma si può scegliere qualsiasi altro valore. Le R ed L sono molto sensibili anche a piccole variazioni dei parametri S. Per esempio le resistenze parassite R_{ij} (associate ad un certo S_{ij}) sono dell'ordine di grandezza dei $m\Omega$. Ponendo un'impedenza di riferimento di 50Ω si commetterebbe quindi un errore di 5000 volte. Per questo motivo sarebbe meglio rinormalizzare ponendo come impedenza di riferimento ad un valore più piccolo di 50 (per esempio 1Ω in generale va bene). In questo modo si riduce l'errore sul calcolo delle R e delle L.
2. **FastFit**: questo state space fitting algorithm è basato sempre sul metodo del **vector fitting**, in particolare sulla variante **fast vector fitting** descritta all'interno della sezione 4.5. Questa variante ha una buona accuratezza a in particolar modo a basse frequenze.
3. **TWA (Tsuk-White Algorithm)**: questo state space fitting algorithm è basato sulla tecnica del **singular value decomposition** (sezione 4.5).
4. **Iterated rational fit**: questo state space fitting algorithm, per ogni elemento della matrice dei parametri S, calcola poli e zeri in modo iterativo. Ad ogni iterazione l'ordine (e quindi il numero di poli) viene aumentato per avere una maggiore accuratezza di calcolo. Le iterazioni vengono eseguite fino a che non si ha un fitting dei dati accettabile.

Nell'esempio del convertitore Buck del Capitolo 5, le opzioni relative alla **causality**, alla **passivity** e al **column fitting** non hanno prodotto risultati migliori rispetto a quelli ottenuti scegliendo solamente le impostazioni come **renormalize**, **desired fitting error** e **state space fitting algorithm**.

Le impostazioni che **inizialmente** sono risultate più convenienti sono state:

1. Desired fitting error: 0.01%;
2. Renormalize: 1 Ohms;
3. State space fitting algorithm: Iterated rational fit.

Oltre a generare circuiti equivalenti a quello originale, Q3D permette anche di eseguire dei plot di quantità di interesse (come R, L, C, G parassite) nella sezione "results". Ovviamente Q3D presenta tante altre funzionalità fuori dai bisogni legati a questo studio. In questa sezione ci si è limitati a descrivere solo le funzionalità scoperte e utili per gli scopi di questo studio sui parassiti.

4.4 Matlab: generazione delle broadband netlist

Oltre che con Q3D e ADS, è possibile generare delle broadband netlist anche con l'ausilio della funzione `rational()` su matlab. Maggiori dettagli pratici sono illustrati nel Capitolo 5.

Le broadband netlist ottenute tramite ADS, Q3D e Matlab **devono rispondere ai requisiti di passivity, causality e stability per poter essere simulate nel dominio del tempo in Cadence. Se solo uno di questi 3 requisiti non venisse rispettato, la simulazione in transient della netlist non convergerebbe.** Le definizioni di passivity, causality e stability sono riportate nel paragrafo 1.4.

4.5 Vector fitting - teoria

I broadband generators tool fin qui analizzati, permettono di ottenere le netlist equivalenti di un dato circuito eseguendo un metodo iterativo di calcolo che prende il nome di **vector fitting**. Questo metodo, in teoria permette di ottenere delle netlist contemporaneamente **passive, causali e stabili**. Se queste tre condizioni non sono soddisfatte, ci saranno problemi di non convergenza durante la cosimulazione in transient delle netlist equivalenti dei circuiti esaminati. Una volta ottenuta la netlist, occorre quindi assicurarsi che queste tre condizioni siano soddisfatte. I concetti riguardanti la passivity, la causality e la stability sono già stati illustrati nel Capitolo 1. Segue ora una trattazione dettagliata del metodo del vector fitting [24] e di alcune sue varianti [25] [26] incontrate in questo studio.

4.5.1 Standard vector fitting

Nel seguito è mostrata la teoria riguardante il vector fitting: grazie al vector fitting i parametri S possono essere approssimati nel modo seguente: Una qualsiasi funzione $f(s)$ può essere ricostruita, approssimata da una funzione razionale, nel modo seguente:

$$f(s) \cong \sum_{n=1}^N \frac{c_n}{s - a_n} + d + sh \quad (4.6)$$

dove $s = j\omega$ rappresenta la frequenza, i c_n sono i residui e gli a_n rappresentano i poli. Invece d ed h sono numeri reali. I c_n e gli a_n sono numeri complessi, mentre d e h sono reali. In questo problema i c_n , gli a_n , d ed h sono le incognite. Quello appena menzionato è un problema non lineare perchè gli a_n compaiono a denominatore. Il vector fitting prevede che il problema sia risolto **in due soli step sequenziali e lineari**, utilizzando in entrambi gli step un set di poli predefinito, iniziale, ovvero gli

\tilde{a}_n . Nel primo step si calcolano i poli di $f(s)$, ovvero gli a_n (**da NON confondere con gli \tilde{a}_n**), mentre nel secondo step si trovano i residui di $f(s)$, cioè i c_n (**da NON confondere con i \tilde{c}_n** che compariranno in seguito nei calcoli).

Step 1: calcolo dei poli a_n di $f(s)$

Per prima cosa si definisce una funzione $\sigma(s)$ sconosciuta, la cui approssimazione razionale è la seguente:

$$\sigma(s) = \sum_{n=1}^N \frac{\tilde{c}_n}{s - \tilde{a}_n} + 1 \quad (4.7)$$

In $\sigma(s)$ è contenuto il set iniziale di poli \tilde{a}_n . Viene definito poi il prodotto tra $\sigma(s)$ e $f(s)$ come:

$$\sigma(s) \cdot f(s) \cong \sum_{n=1}^N \frac{c_n}{s - \tilde{a}_n} + d + sh \quad (4.8)$$

Da notare come $\sigma(s)$ e il prodotto $\sigma(s) \cdot f(s)$ abbiano gli stessi poli \tilde{a}_n . Sostituendo $\sigma(s)$ con la sua approssimazione razionale nell'equazione (4.8) si otterrà quindi:

$$\left(\sum_{n=1}^N \frac{\tilde{c}_n}{s - \tilde{a}_n} + 1 \right) \cdot f(s) \cong \sum_{n=1}^N \frac{c_n}{s - \tilde{a}_n} + d + sh \quad (4.9)$$

che, in termini di funzioni è equivalente a:

$$\sigma(s) \cdot f(s) \cong (\sigma \cdot f)(s) \quad (4.10)$$

L'equazione (4.9) può essere riscritta come:

$$\left(\sum_{n=1}^N \frac{c_n}{s - \tilde{a}_n} + d + sh \right) - \left(f(s) \sum_{n=1}^N \frac{\tilde{c}_n}{s - \tilde{a}_n} \right) = f(s) \quad (4.11)$$

che rappresenta un'equazione lineare del tipo $\mathbf{Ax} = \mathbf{b}$ dal punto di vista delle incognite c_n, d, h e dei \tilde{c}_n (con n che va da 1 a N).

Ad una certa frequenza $s_k = j\omega_k$ gli elementi dei vettori \mathbf{A}_k , \mathbf{x} e \mathbf{b} sono definiti come:

$$A_k = \begin{bmatrix} \frac{1}{s_k - \tilde{a}_1} & \dots & \frac{1}{s_k - \tilde{a}_N} & 1 & s_k & \frac{-f(s_k)}{s_k - \tilde{a}_1} & \dots & \frac{-f(s_k)}{s_k - \tilde{a}_N} \end{bmatrix} \quad (4.12)$$

$$x = \begin{bmatrix} c_1 & \dots & c_N & d & h & \tilde{c}_1 & \dots & \tilde{c}_N \end{bmatrix}^T \quad (4.13)$$

$$b_k = f(s_k) \quad (4.14)$$

dove \mathbf{A}_k è la k -esima riga corrispondente alla frequenza s_k della matrice \mathbf{A} . Ritornando all'equazione (4.10), le funzioni $(\sigma \cdot f)(s)$ e $\sigma(s) \cdot f(s)$ possono essere riscritte rispettivamente come prodotti di zeri e poli nel modo seguente:

$$(\sigma \cdot f)(s) = h \frac{\prod_{n=1}^{N+1} (s - z_n)}{\prod_{n=1}^N (s - \tilde{a}_n)} \quad (4.15)$$

$$\sigma(s) = \frac{\prod_{n=1}^N (s - \tilde{z}_n)}{\prod_{n=1}^N (s - \tilde{a}_n)} \quad (4.16)$$

Dalle due equazioni (4.15) e (4.16) possiamo quindi ricavare $f(s)$ come:

$$f(s) = \frac{(\sigma \cdot f)(s)}{\sigma(s)} = h \frac{\prod_{n=1}^{N+1} (s - z_n)}{\prod_{n=1}^N (s - \tilde{z}_n)} \quad (4.17)$$

Osservando queste ultime equazioni si nota che i poli di $f(s)$ coincidono con gli zeri di $\sigma(s)$. Come è possibile notare da quest'ultima equazione, il set di poli iniziale \tilde{a}_n scompare, si semplifica nella divisione perchè sia per $\sigma(s)$ che per $(\sigma \cdot f)(s)$ era stato definito il set di poli iniziale \tilde{a}_n .

il nuovo set di poli trovato \tilde{z}_n potrebbe avere dei poli instabili, cioè aventi parte reale positiva. In tal caso vengono resi stabili invertendo la loro parte reale.

Gli zeri \tilde{z}_n di $\sigma(s)$ vengono calcolati risolvendo un problema agli autovalori (eigenvalues), cioè:

$$\{\tilde{z}_n\} = \text{eig}(A - b \cdot \tilde{c}^T) \quad (4.18)$$

dove A è la matrice diagonale contenente in poli iniziali \tilde{a}_n , b è un vettore colonna avente tutti 1, mentre \tilde{c}^T è il vettore riga (trasposto) contenente i residui \tilde{c}_n di $\sigma(s)$

definiti nell'equazione (4.7).

Come è possibile capire se è stato scelto il giusto set di poli di partenza \tilde{a}_n ? Nell'equazione (4.16) di $\sigma(s)$, a numeratore compare il set di poli di $f(s)$ calcolati (gli \tilde{z}_n), mentre a denominatore si ha il set di poli di partenza di $f(s)$ (gli \tilde{a}_n). Numeratore e denominatore hanno lo stesso grado N . Come spiegato in precedenza, gli \tilde{a}_n sono gli stessi sia per $f(s)$ che per $\sigma(s)$. Se i \tilde{z}_n calcolati vengono uguali (o si discostano di pochissimo) rispetto agli \tilde{a}_n di partenza significa che questi ultimi sono stati scelti correttamente fin all'inizio. Ciò significa che:

$$\sigma(s) = \frac{\prod_{n=1}^N (s - \tilde{z}_n)}{\prod_{n=1}^N (s - \tilde{a}_n)} \cong 1 \quad (4.19)$$

Ma $\sigma(s)$ è definito anche come in (4.7). Quindi $\sigma(s) \cong 1$ implica che in (4.9) si abbiano i $\tilde{c}_n \cong 0$. Perciò il vettore delle incognite \mathbf{x} in (4.13) avrà gli elementi $\tilde{c}_1 \cong 0, \dots, \tilde{c}_N \cong 0$. Quanto appena illustrato indica che nel processo di calcolo dei poli di $f(s)$ vi è convergenza. **Questo step di calcolo dei poli è un processo iterativo. Vengono eseguiti iterativamente i calcoli definiti in (4.11) e in (4.31). All'iterazione successiva, i poli ottenuti in (4.31) vengono utilizzati come set di poli iniziali in (4.11) e così via finché non si ha convergenza.** Significa che ripetere più volte il calcolo dei poli utilizzando quelli calcolati nell'iterazione precedente permette di riposizionare i poli in posizioni migliori e quindi di avere una migliore accuratezza della $f(s)$ fittata che si andrà ad ottenere.

Nel caso si debbano fittare funzioni aventi dei picchi di risonanza, la scelta del set di poli iniziale \tilde{a}_n deve essere tale che:

$$\tilde{a}_n = -\alpha + j\beta; \quad \tilde{a}_{n+1} = -\alpha - j\beta; \quad \text{con} \quad \alpha = \frac{\beta}{100} \quad (4.20)$$

Invece per funzioni che presentano un andamento più smooth risulta sufficiente scegliere poli reali.

Step 2: calcolo dei residui c_n di $f(s)$

I residui di $f(s)$ sono calcolati sostituendo i poli a_n in (4.6) con gli zeri di $\sigma(s)$, cioè gli \tilde{z}_n dell'equazione (4.16). $f(s)$ sarà quindi definita come:

$$f(s) \cong \sum_{n=1}^N \frac{c_n}{s - \tilde{z}_n} + d + sh \quad (4.21)$$

in questo modo, si risolve il sistema $\mathbf{Ax}=\mathbf{b}$ che rispecchia proprio l'equazione (4.21). L'unica differenza rispetto allo **Step 1** è che ad una certa frequenza $s_k = j\omega_k$ si ha che \mathbf{A} , \mathbf{x} e \mathbf{b} sono definiti come:

$$A_k = \begin{bmatrix} 1 & \dots & 1 & 1 & s_k \\ s_k - \tilde{z}_1 & & s_k - \tilde{z}_N & & \end{bmatrix} \quad (4.22)$$

$$x = \begin{bmatrix} c_1 & \dots & c_N & d & h \end{bmatrix}^T \quad (4.23)$$

$$b_k = f(s_k) \quad (4.24)$$

I residui di $f(s)$ potrebbero anche essere calcolati utilizzando l'equazione (4.17) ma si ottengono risultati più accurati seguendo la modalità appena menzionata.

Per quanto riguarda i parametri S , essi vengono ricostruiti nel modo seguente:

$$S(s) = \sum_{m=1}^N \frac{R_m}{s - a_m} + D \quad (4.25)$$

4.5.2 Relaxed vector fitting

Rappresenta una variante con un grado di libertà in più rispetto allo standard vector fitting. A differenza di quest'ultimo, \tilde{d} è generico (non fissato). Quindi la funzione $\sigma(s)$ sarà definita come:

$$\sigma(s) = \sum_{n=1}^N \frac{\tilde{c}_n}{s - \tilde{a}_n} + \tilde{d} \quad (4.26)$$

Il passaggio successivo non cambia, definendo la relazione:

$$\sigma(s) \cdot f(s) \cong (\sigma \cdot f)(s) \quad (4.27)$$

e sostituendo si otterrà:

$$\left(\sum_{n=1}^N \frac{\tilde{c}_n}{s - \tilde{a}_n} + \tilde{d} \right) \cdot f(s) \cong \sum_{n=1}^N \frac{c_n}{s - \tilde{a}_n} + d + sh \quad (4.28)$$

A differenza dello standard vector fitting, si ha che \tilde{d} non è più fissato ad 1 ma può avere un valore più libero, sconosciuto. Predersi questo grado di libertà impone una nuova equazione che permetta di avere convergenza:

$$Re \left\{ \sum_{k=1}^{N_s} \left(\sum_{n=1}^N \frac{\tilde{c}_n}{s - \tilde{a}_n} + \tilde{d} \right) \right\} = N_s \quad (4.29)$$

dove N_s rappresenta una quantità finita.

Questa relazione permette di evitare di avere $\sigma(s) \cong 0$ alle alte frequenze.

Alle alte frequenze, $\sigma(s)$ tenderà comunque a 1 anche se \tilde{d} non è forzato a 1. In questo modo le condizioni di convergenza rimangono immutate rispetto allo standard vector fitting ($\tilde{c}_n \cong 0$, cioè $\sigma(s) \cong 1$). Viene definita inoltre, la seguente quantità peso come:

$$weight = \frac{\|w(s) \cdot f(s)\|_2}{N_s} \quad (4.30)$$

Gli zeri vengono così calcolati:

$$\{\tilde{z}_n\} = eig\left(A - b \cdot \tilde{d}^{-1} \cdot \tilde{c}^T\right) \quad (4.31)$$

Questa variante del vector fitting permette di avere una ricollocazione dei poli migliore e più precisa di quella che si avrebbe utilizzato lo standard vector fitting.

4.5.3 Fast vector fitting

In certi casi la risoluzione del sistema $Ax = b$ ricavata da (4.11) (standard vector fitting) e (4.28) (relaxed vector fitting) può essere time consuming e richiedere molta memoria. Esiste una versione del vector fitting, detta **fast vector fitting** che permette di effettuare meno calcoli in meno tempo. Questa metodologia di vector fitting si basa sulla tecnica del **QR-decomposition**. Partendo da un sistema lineare:

$$Ax = b \quad (4.32)$$

dove A è una matrice, x e b sono vettori colonna.

Poniamo poi:

$$A = QR \quad (4.33)$$

dove Q è una matrice unitaria ed R è una matrice upper triangular definita come:

$$R = \begin{pmatrix} R_{11} & R_{12} & \dots & R_{1N} \\ 0 & R_{22} & \dots & R_{2N} \\ \vdots & 0 & \dots & \vdots \\ \vdots & \dots & \dots & \vdots \\ 0 & \dots & 0 & R_{NN} \end{pmatrix} \quad (4.34)$$

nella quale gli elementi che stanno sotto la diagonale principale $R_{11}, R_{22}, \dots, R_{NN}$ sono tutti 0.

In (4.32), moltiplichiamo primo e secondo membro a sinistra per la matrice trasposta di Q , vale a dire Q^T :

$$Q^T Ax = Q^T b \quad (4.35)$$

Nella (4.35) sostituiamo la matrice A con la sua nuova definizione in termini di QR-decomposizione illustrata in (4.33). Si otterrà quindi che il vettore soluzione incognito x si trova come soluzione di:

$$Q^T QRx = Q^T b \quad \Rightarrow \quad Rx = Q^T b \quad (4.36)$$

4.5.4 Vector fitting - Singular value decomposition

La variante del singular value decomposition può essere utile usarla per migliorare l'accuratezza: quando si risolve il sistema $\mathbf{Ax}=\mathbf{b}$ proposto in (4.11), può essere utile ridefinire la matrice \mathbf{A} come:

$$A = U\Sigma V^T \quad (4.37)$$

dove U e V^T sono matrici quadrate di dimensione $M \times M$ ed $N \times N$ rispettivamente. Le matrici A e Σ invece sono rettangolari di dimensione $M \times N$. N rappresenta il numero di poli ed M il numero di punti in frequenza.

La matrice U è formata da vettori colonna definiti come $v_{L,i}$, con $1 \leq i \leq M$, dove i rappresenta la i -esima colonna. I $v_{L,i}$ sono detti **left singular vectors**.

$$U = \begin{pmatrix} v_{L,1} & v_{L,2} & \dots & v_{L,M} \end{pmatrix} \quad (4.38)$$

La matrice V^T è formata da vettori riga definiti come $v_{R,i}$, con $1 \leq i \leq N$, dove i rappresenta la i -esima riga. I $v_{R,i}$ sono detti **right singular vectors**.

$$V^T = \begin{pmatrix} v_{R,1} \\ v_{R,2} \\ \vdots \\ v_{R,N} \end{pmatrix} \quad (4.39)$$

La matrice Σ è così costruita:

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & \vdots \\ \vdots & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \vdots \\ 0 & \dots & \dots & \sigma_N \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \quad (4.40)$$

dove $\sigma_1 \geq \sigma_2 \geq \dots \sigma_{N-1} \geq \sigma_N$. I σ_i sono chiamati **singular values della matrice A**. Tutti gli altri elementi di Σ valgono 0.

Gli elementi σ_i , con $1 \leq i \leq N$, sono tali che:

$$\begin{aligned} AA^T v_{L,1} &= \sigma_1^2 v_{L,1}; & A^T A v_{R,1} &= \sigma_1^2 v_{R,1}; \\ AA^T v_{L,2} &= \sigma_2^2 v_{L,2}; & A^T A v_{R,2} &= \sigma_2^2 v_{R,2}; \\ &\vdots & &\vdots \\ &\vdots & &\vdots \\ AA^T v_{L,N} &= \sigma_N^2 v_{L,N}; & A^T A v_{R,N} &= \sigma_N^2 v_{R,N}; \end{aligned} \quad (4.41)$$

Capitolo 5

Cosimulazione: package del convertitore Buck

In questo capitolo sono riportati i risultati più importanti riguardanti **le cosimulazioni effettuate su SPICE** di un circuito di esempio, una volta ottenuti i suoi parametri S. L'obiettivo è stato quello di trovare come i **parassiti relativi ai fili di bonding del package** influiscono sul funzionamento e sulle prestazioni del **convertitore buck sincro** fornito dall'azienda. Si è mostrato necessario capire le differenze nelle forme d'onda ottenute con i diversi formati: parametri S, broadband netlist, netlist RLCG ad una frequenza e caso ideale.

5.1 Buck: brevi cenni di teoria

L'esempio analizzato è stato quello relativo al package di un **convertitore Buck sincro**. Si mostra necessario, per prima cosa, fare una breve descrizione teorica del comportamento del circuito in generale per poi passare ad uno studio più approfondito dei parassiti e delle non idealità relative al **package e ai suoi fili di bonding** con l'ausilio delle simulazioni su ADS, Ansys Q3D e Cadence Virtuoso. Nella figura sottostante è mostrato lo schematico teorico (che ancora non permette di tenere conto dei parametri parassiti) del Buck sincro:

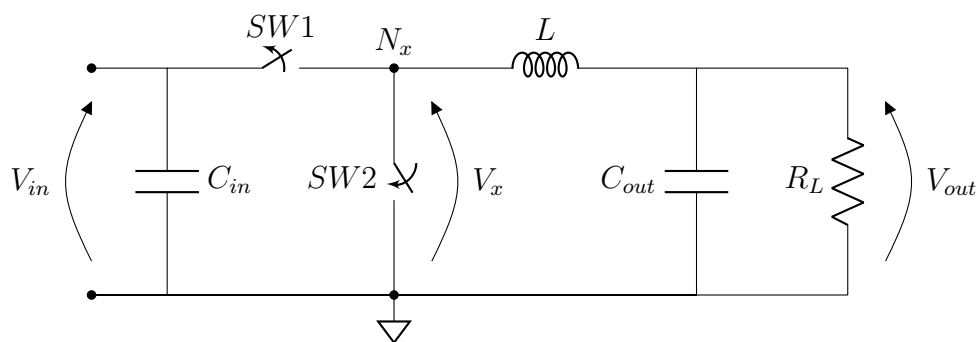


Figura 5.1: Schematico teorico di un Buck sincrono con $V_{in} = 5.5V$.

Lo studio del Buck e la cosimulazione transient su Cadence sono riferiti alla modalità di funzionamento **CCM (Continuous Current Mode)**. Come già noto, in CCM la corrente sull'induttore L non è mai nulla per tutto il periodo di ciclo T_{SW} degli interruttori $SW1$ e $SW2$.

$SW1$ e $SW2$ sono in controfase tra loro: ciò significa che quando uno è aperto l'altro sarà chiuso e viceversa. In questo modo sul **nodo di switching** N_x (avente tensione V_x) ci sarà un'onda quadra compresa tra $0V$ e V_{in} . **La tensione di ingresso V_{in} è costante nel tempo e in questo progetto vale $5.5V$.** L'onda quadra su V_x ha il duty-cycle pari a quello che fa sì che gli interruttori $SW1$ e $SW2$ si aprano e chiudano, alternandosi.

Nello spazio sottostante è riportata la funzione di trasferimento che modella il funzionamento teorico del circuito in CCM:

$$\frac{V_{out}}{V_{in}} = \frac{T_{HIGH1}}{T_{HIGH1} + T_{LOW1}} = DC < 1 \quad (5.1)$$

dove DC è il duty-cycle. T_{HIGH1} è il periodo di tempo per il quale $SW1$ rimane chiuso (e $SW2$ aperto) e quindi $V_x = V_{in}$. Invece durante T_{LOW1} , lo switch $SW1$ è aperto (e $SW2$ chiuso) e si ha che $V_x = 0V$.

I tempi T_{HIGH1} e T_{LOW1} sono tali che $T_{SW} = T_{HIGH1} + T_{LOW1}$.

Nelle figure sottostanti sono raffigurati gli andamenti teorici delle tensioni di ingresso, di uscita e sul nodo di switching N_x nel dominio del tempo:

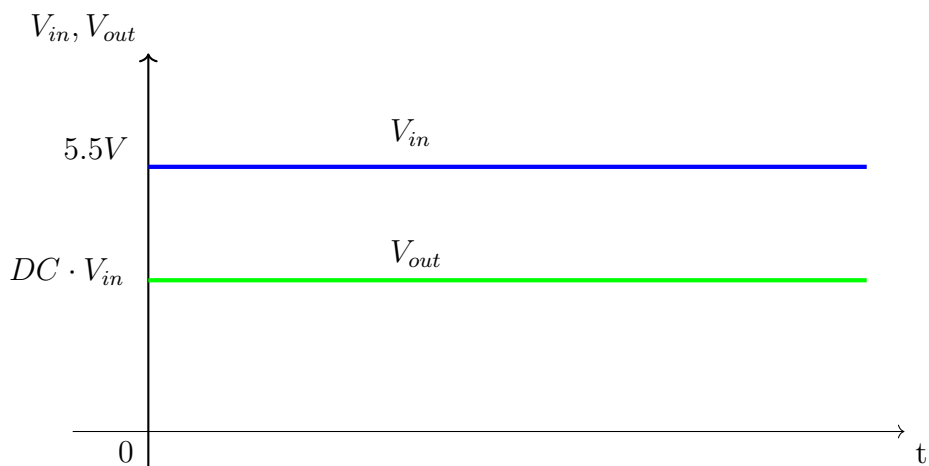


Figura 5.2: Andamento teorico delle tensioni V_{in} e V_{out} del Buck sincrono.

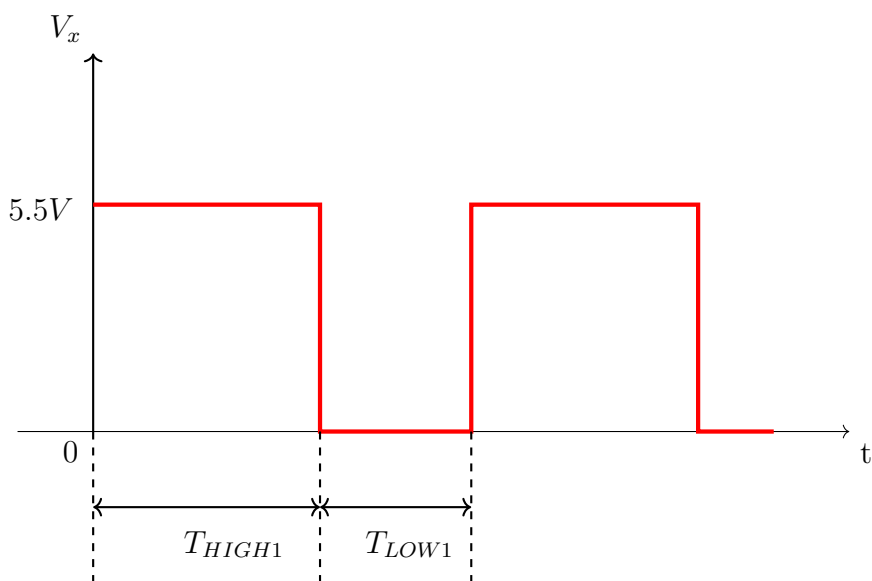


Figura 5.3: Andamento teorico della tensione V_x del Buck sincrono: onda quadra.

Quindi, noto il periodo di ciclo T_{SW} , la **frequenza di switching** è definita come:

$$f_{SW} = \frac{1}{T_{SW}} = \frac{1}{T_{HIGH1} + T_{LOW1}} \quad (5.2)$$

5.2 Simulazione elettromagnetica del package

La struttura analizzata ha 9 porte e comprende alcuni dei fili di bonding del package preso in esame. Un esempio puramente generale di fili di bonding di un certo package è mostrato nella figura seguente:

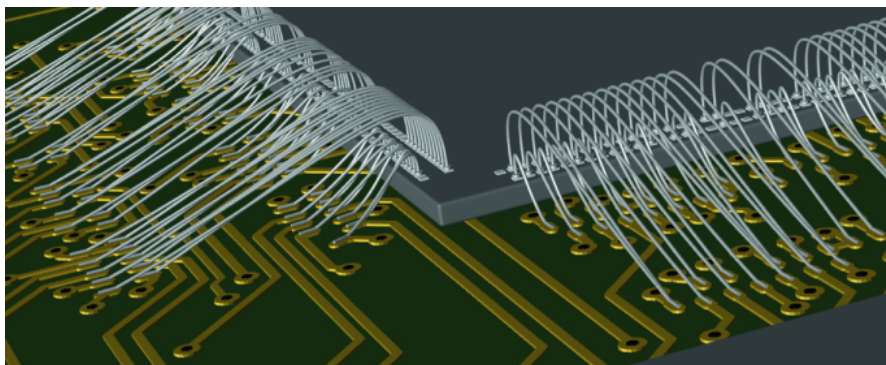


Figura 5.4: Esempio generico di fili di bonding in un package.

Per ottenere i parametri S, su Q3D, è stata effettuata la simulazione elettromagnetica **tra 1Hz e 1GHz, con step logaritmico di 50 punti/decade** del package del Buck fornito dall'azienda. Questo ha permesso di ottenere il file touchstone dei parametri S, fondamentale in questo studio.

5.3 Netlist equivalenti del package

Come già spiegato precedentemente, i due tipi di netlist utilizzati in questo studio sono:

- **Broadband netlist** su tutto il range di frequenza;
- **Netlist RLCG a 2MHz.**

Una volta ottenuti i parametri S, per generare la **broadband netlist** sul range di frequenza scelto, si sono utilizzati rispettivamente i tool descritti nei capitoli precedenti: il **Broadband SPICE model generator di ADS**, il **Broadband export tool di Q3D** e la **funzione rational() di Matlab**. Queste 3 modalità, in input richiedono il file touchstone dei parametri S. Le 3 broadband netlist ottenute sono state confrontate, per poter successivamente importare su Cadence Virtuoso la netlist migliore in termini di accuracy e utilizzarla nelle cosimulazioni nel dominio del tempo.

L'esigenza era anche quella di ottenere una topologia allo stesso tempo **accurata, passiva, stabile e causale**, come già spiegato nel Capitolo 1.

Sia con il Broadband SPICE model generator di ADS che con il Broadband export tool di Q3D non si sono riuscite ad ottenere delle broadband netlist contemporaneamente accurate, passive, stabili e causali tra 1Hz e 1GHz, e con l'ordine del circuito equivalente desiderato, per via delle limitate opzioni user defined. A causa di questo, le cosimulazioni in transient possono causare problemi di convergenza a causa della violazione della condizione di passività o, nel caso di convergenza, i valori ottenuti possono risultare largamente imprecisi.

La broadband netlist ottenuta utilizzando la funzione **rational()** in Matlab invece è risultata essere contemporaneamente passiva, causale, stabile e accurata. L'utilizzo di questa netlist ha portato alla convergenza della successiva cosimulazione transient su Cadence. Come già scritto nei capitoli precedenti, la funzione **rational()** è basata sull'algoritmo del **vector fitting**. Sono risultati sufficienti 16 poli per generare la netlist equivalente. In seguito, con la funzione **makepassive()**, il circuito è stato reso passivo, mantenendo un livello di accuratezza ottimo.

La broadband netlist generata ha degli **errori relativi percentuali massimi su R ed L (tra tutti gli S_{ij}) su tutto il range di frequenza tra 1Hz e 1GHz** i più piccoli possibile, che sono rispettivamente:

$$\text{Max} \left(\frac{\Delta R}{R} \% \right) = 7\% \quad (5.3)$$

$$\text{Max} \left(\frac{\Delta L}{L} \% \right) = 18\% \quad (5.4)$$

Questi errori relativi rappresentano la differenza tra la broadband netlist generata e i parametri S dati in ingresso.

Tali quantità permettono di capire qual'è il livello di accuratezza del modello ottenuto.

Un altro tipo di modello circuitale generato e poi utilizzato è la **netlist RL-CG a 2MHz**, sempre ottenuta con Q3D. Essa è basata sul T-model, descritto nel Capitolo 2.

5.4 Simulazioni preliminari

Per verificare che i valori dei parassiti ottenuti con i circuiti equivalenti ricalcassero in modo preciso i parametri S originari su tutto il range di frequenza, si sono misurate su SPICE, performando una **simulazione AC**, le R ed L parassite dei fili di bonding. Nell'immagine seguente è riportato il testbench relativo alla self- R e self- L del filo 5-9 che è proprio **il filo relativo al nodo di switching N_x avente tensione V_x del buck sincrono teorico (Figura 5.1)**. Le altre porte sono state lasciate flottanti perchè non influiscono sui parassiti diretti del filo 5-9. Sono stati costruiti testbench analoghi per trovare le self- R e le self- L di tutti gli altri fili e anche per misurare le **R ed L mutue (Capitolo 2)** tra due diversi fili. **Le stesse analisi sono state effettuate anche per la netlist RLCG definita a 2MHz e per i parametri S letti tramite il component nport.**

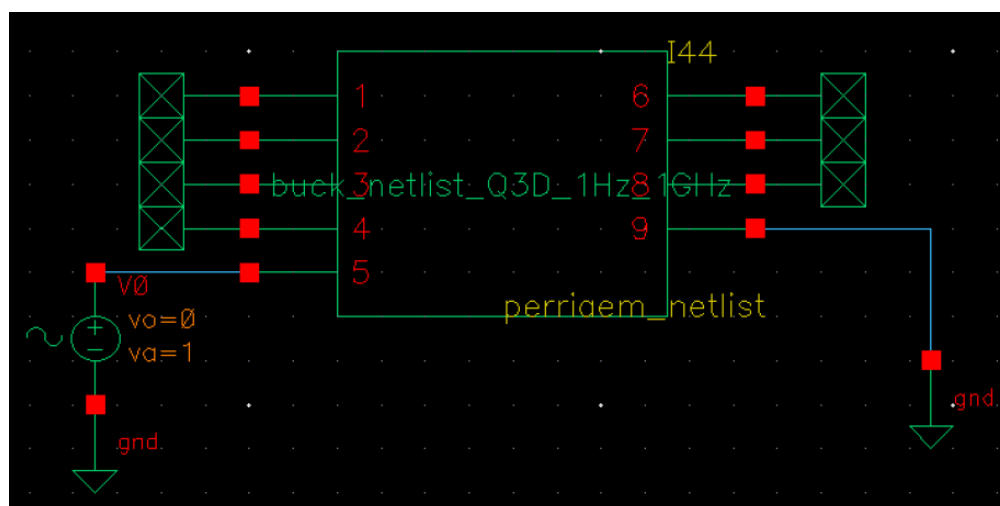


Figura 5.5: Cadence: testbench costruito per l'analisi AC di R ed L parassite riguardante il filo 5-9 - BROADBAND netlist completa a 9 porte.

5.4.1 Risultati

Gli andamenti delle self R ed L parassite ottenuti dalla simulazione AC su SPICE dei testbench appena menzionati, per il filo 5-9, sono riportati nelle figure seguenti:

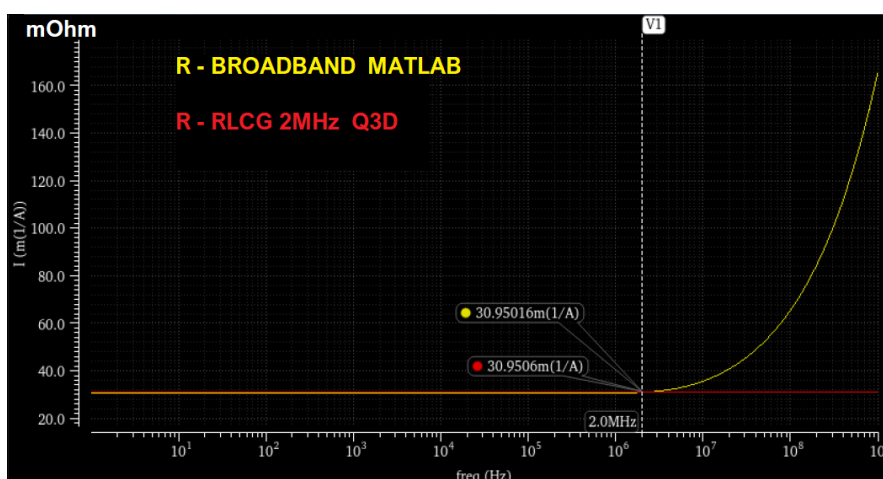


Figura 5.6: Risultati analisi AC in SPICE: confronto della R tra broadband e RLCG netlist a 2MHz - filo 5-9.

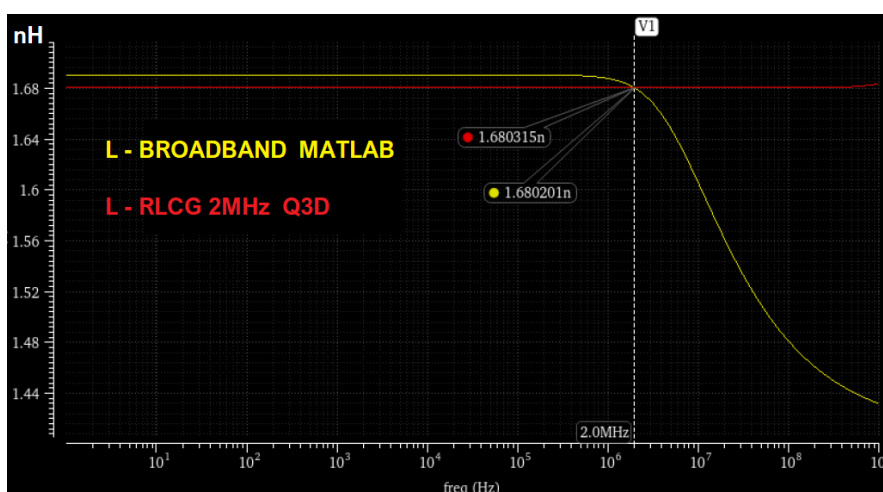


Figura 5.7: Risultati analisi AC in SPICE: confronto della L tra broadband e RLCG netlist a 2MHz - filo 5-9.

La curva rossa associata alla R (o L) del filo dell'equivalente RLCG a 2MHz, risulta corretta solo alla frequenza di 2MHz, e si interseca alla frequenza di 2MHz con la curva gialla che rappresenta la R (o L) associata alla broadband netlist. Per quanto riguarda l'equivalente RLCG a 2MHz, la R (o L) è costante al variare della frequenza e questo non rispecchia la realtà: la curva gialla invece rispecchia con buona accuratezza la realtà su tutto il range di frequenza.

Per L'equivalente RLCG a 2MHz, il fatto che la R (o L) non vari in frequenza significa che il circuito equivalente è sempre lo stesso a tutte

le frequenze. Al contrario, per la broadband netlist, il fatto che R (o L) cambi in frequenza, implica che al variare della frequenza, il circuito equivalente sia diverso. Inoltre, gli andamenti di R ed L (curve gialle) ricalcano il fenomeno dello skin effect spiegato nel Capitolo 3.

Confronto con Q3D

Per verificare che quanto ottenuto su SPICE fosse corretto, si sono confrontate le R e L della broadband netlist (curve gialle, figure 5.6, 5.7) con quelle ottenute su Q3D, sempre relativamente al filo 5-9. Gli andamenti trovati su Q3D sono riportati qui sotto:

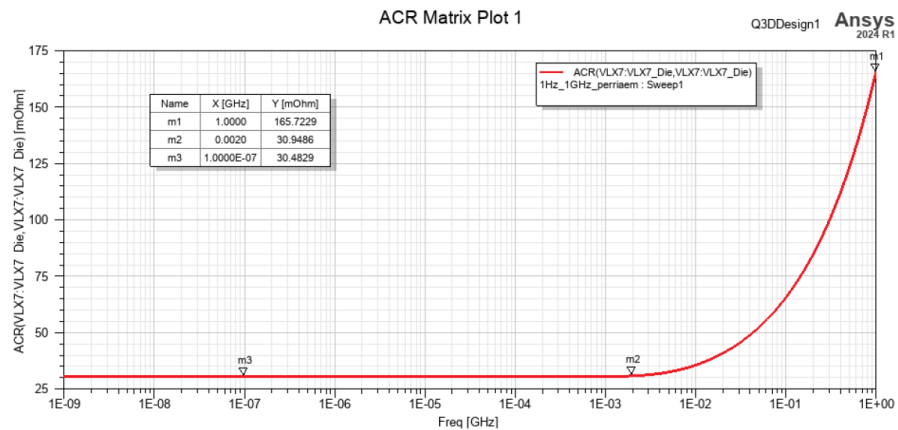


Figura 5.8: Filo 5-9: comportamento in frequenza della R parassita su Q3D.

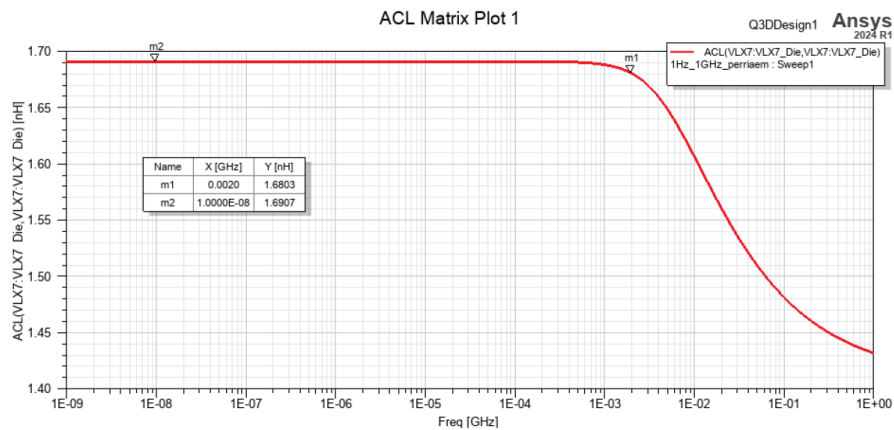


Figura 5.9: Filo 5-9: comportamento in frequenza della L parassita su Q3D.

Come si può notare, gli andamenti di R (confronto tra Figura 5.8 e 5.6) e L (confronto tra Figura 5.9 e 5.7), tra Cadence e Q3D sono identici.

Anche per tutti gli altri fili, i rispettivi andamenti degli elementi parassiti simulati con SPICE con la netlist equivalente risultano coincidere con i valori ottenuti da Q3D.

5.5 Cosimulazioni

Una volta accertati che la broadband netlist ricalca in modo preciso i parametri S originali su tutto il range di frequenza, è stata eseguita la **cosimulazione del progetto del Buck** la cui frequenza di switching è pari a 2MHz su **SPICE**, utilizzando il simulatore **Eldo**.

In particolare sono state eseguite tre tipologie di cosimulazioni con differenti modelli equivalenti del package dei fili del buck:

1. Cosimulazione con l'equivalente in termini di **parametri S**;
2. Cosimulazione con il **broadband circuit** equivalente (definito su tutto il range di frequenze scelto) esportato da Matlab a partire dai parametri S. Per generare l'equivalente broadband circuit serve prima di tutto avere i parametri S;
3. Cosimulazione con **circuito RLCG definito ad una frequenza (la frequenza di switching, 2MHz)** e ottenuto da Q3D.

In queste prove appena citate, all'interno del progetto su SPICE fornito dall'azienda, la parte dello schematico dedicata ai fili di bonding è stata sostituita rispettivamente con il suo equivalente in termini di **parametri S**, con il suo **broadband circuit equivalente** e con il **circuito equivalente RLCG definito a 2MHz**.

I risultati di queste tre cosimulazioni, sono stati confrontati per capire quale di esse fornisce i risultati più accurati. Ciò, rappresenta lo scopo finale della Tesi.

Su SPICE, i parametri S vengono integrati nel circuito elettrico fornito dall'azienda utilizzando il component nport. Invece, il broadband circuit e la netlist RLCG ad una certa frequenza, sono direttamente importati creando il file text.netlist.

Per dettagli relativi all'nport e al text.netlist si rimanda al Capitolo 4.

Quindi, all'interno del progetto finale su SPICE, il circuito che descrive il comportamento elettrico e il circuito che modella gli elementi parassiti dei fili di bonding del package si interfacciano tra di loro. In questo modo si modella il dispositivo intero che tiene conto anche dei suoi parametri parassiti. Ciò permette di performare una simulazione più realistica. Il significato di cosimulazione è proprio

questo. Nell'immagine seguente è possibile capire quanto appena spiegato:

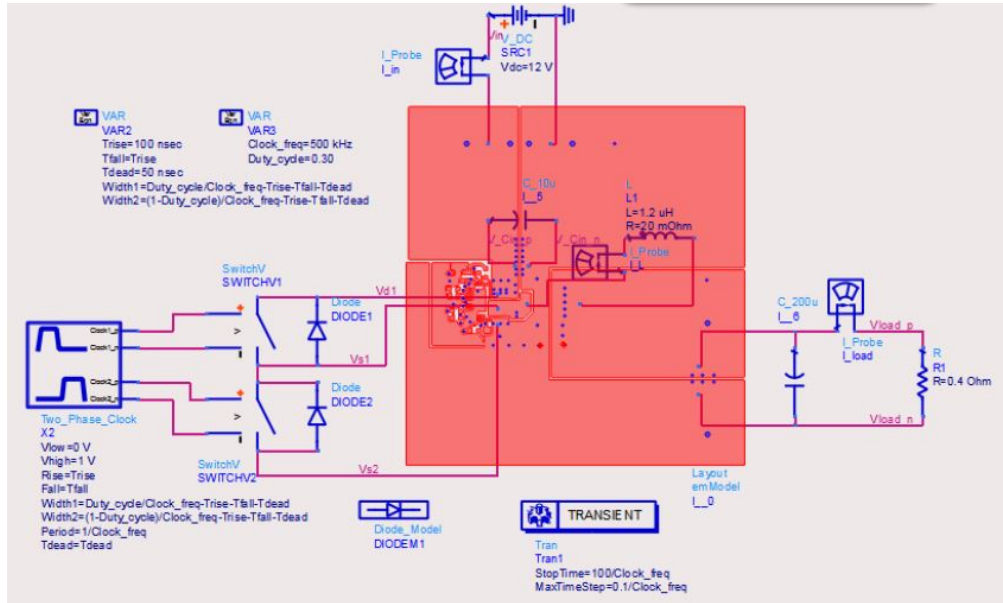


Figura 5.10: Esempio simbolico di schematico dove viene eseguita la cosimulazione tra due sotto circuiti definiti in formati diversi.

Dal punto di vista puramente simbolico, si potrebbe associare al blocchetto rosso della figura precedente i parametri S dei fili di bonding, oppure la broadband netlist o la netlist RCLG a 2MHz. Il resto dello schematico è associabile al circuito elettrico che si interfaccia con gli elementi parassiti. Si vede proprio come il modello dei parametri parassiti è inglobato nel circuito elettrico. Quindi si ha un sistema molto più realistico del semplice schematico teorico che tiene conto solamente dei componenti puramente circuitali, e che trascura le non idealità. **Le immagini seguenti mostrano rispettivamente il blocchetto relativo ai parametri S (tramite l'nport), alla broadband netlist e al circuito RLCG a 2MHz con i loro 9 pin. Nelle varie cosimulazioni, questi modelli circuitali sono stati inglobati nel circuito elettrico su Cadence e si interfacciano con esso.**

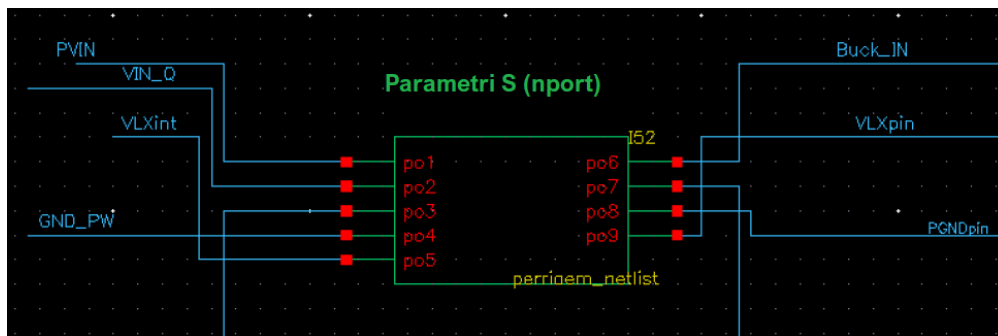


Figura 5.11: Progetto realistico su Cadence fornito dall'azienda: zoom sulla nport a 9 porte che modella i parametri S dei fili Buck che sono inglobati nel circuito elettrico.

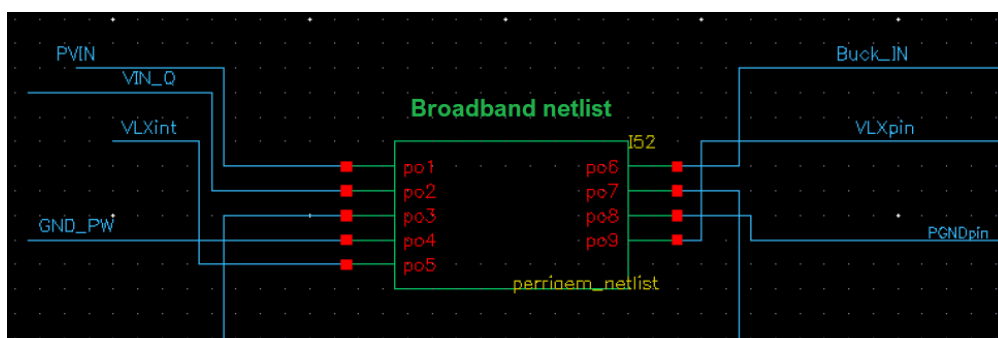


Figura 5.12: Progetto realistico su Cadence fornito dall'azienda: zoom sulla broadband netlist a 9 porte del Buck che è inglobata nel circuito elettrico.

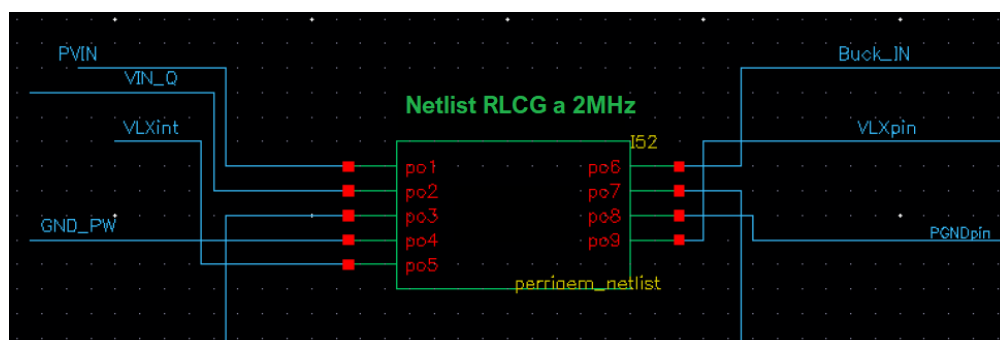


Figura 5.13: Progetto realistico su Cadence fornito dall'azienda: zoom sulla netlist RLCG a 2MHz del Buck avente 9 porte che è inglobata nel circuito elettrico.

Una volta ultimate le tre cosimulazioni in transient descritte in questa sezione, si è effettuato un confronto tra i risultati ottenuti da ciascuna di esse, sovrapponendo i segnali ottenuti per osservare eventuali differenze sia nel dominio del tempo che nel dominio della frequenza.

Le prime due cosimulazioni comunque risultano più realistiche e precise perchè utilizzano parametri S e broadband circuit definiti su tutto il range di frequenza. Infatti, nel caso delle prime due cosimulazioni se si da un'onda quadra in ingresso, **essa eccita tutti i circuiti riferiti alle frequenze delle armoniche**. Nel caso della terza cosimulazione, l'onda quadra in ingresso eccita solo il circuito che è definito ad una sola frequenza (l'armonica principale), mentre vengono persi gli effetti dovuti alle altre armoniche. La netlist RLCG definita per una sola frequenza **risulta costante su tutto il range di frequenza, quando nella realtà non è così**.

Quindi nelle prime due cosimulazioni si tiene conto delle altre armoniche mentre nell'ultima no. L'esigenza è stata anche quella di capire quanto ci si discosta dalla realtà se si utilizza il circuito RLCG definito ad una sola frequenza.

Confronto col caso ideale

Per notare con chiarezza quanto i parametri parassiti dei fili di bonding influiscano sulle prestazioni del circuito, e quindi sulle forme d'onda nel tempo e nella frequenza, si è reso necessario simulare, sempre su SPICE, lo stesso package oggetto delle cosimulazioni precedenti utilizzando però **fili puramente ideali** al posto dei circuiti equivalenti che modellizzano i parametri parassiti in termini di parametri S o broadband netlist. Osservando e analizzando le forme d'onda ottenute è stato possibile notare in questo modo le differenze tra caso ideale e le tre cosimulazioni precedenti (più realistiche).

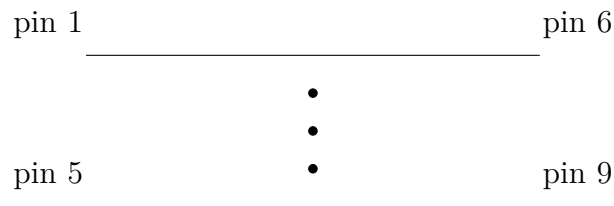


Figura 5.14: Modello puramente ideale dei fili di bonding: corto circuiti.

5.5.1 Risultati

I risultati di quanto descritto nella Sezione 5.5 sono riportati in questo paragrafo. Nella cosimulazione transient si è misurata la tensione sul pin 9 del filo 5-9 relativo al nodo di switching N_x avente tensione V_x (Figura 5.1).

Cosimulazione dei parametri S con nport

Effettuando la cosimulazione direttamente con i parametri S utilizzando il component nport non si sono ottenuti risultati precisi a causa del ridotto numero dei parametri di fitting da poter impostare. Non si ha la possibilità di riscaldare i valori su di una determinata impedenza caratteristica (come su Q3D in cui si può impostare un'impedenza caratteristica di 1 Ohm per avere un miglior fitting), e non si può scegliere il numero di poli desiderato (come invece può essere fatto su Matlab). A causa dell'errore eccessivo sul fitting degli elementi parassiti si può notare come la tensione di uscita risulti alterata rispetto al valore atteso di 1.5 V.

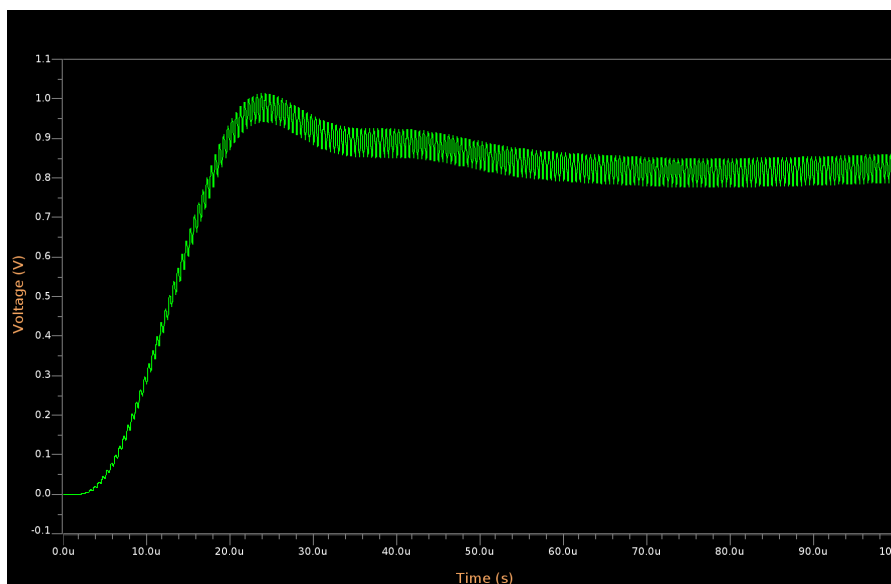


Figura 5.15: Simulazione transient con l'nport. Tensione di uscita

Con il component nport non si riesce ad avere un controllo diretto sulla precisione e sul numero di poli dei parametri S. Ciò rende le cosimulazioni in transient con l'nport molto imprecise.

Cosimulazioni con broadband netlist e circuito RLCG

Per capire meglio quali sono le differenze tra le tre versioni appena viste (**broadband netlist**, **netlist RLCG a 2MHz** e **fili ideali**), che modellizzano i fili di bonding, le forme d'onda della tensione sul nodo di switching (pin 9) sono state sovrapposte ed ingrandite, come mostrato nelle figure seguenti:

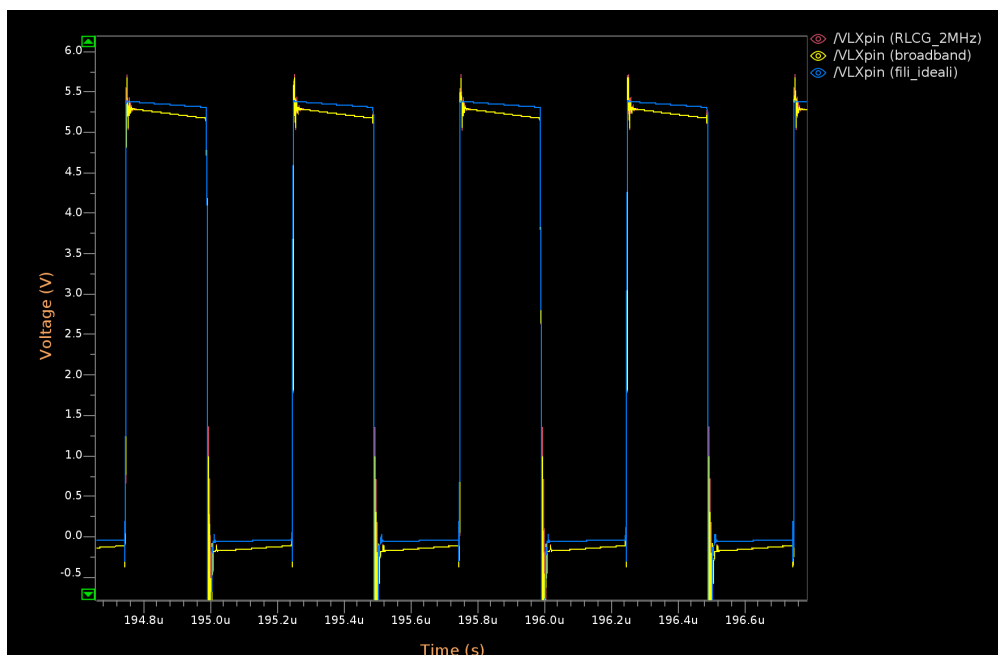


Figura 5.16: Andamento tensione VLXpin (pin 9) delle differenti versioni circuitali: zoom a steady-state.

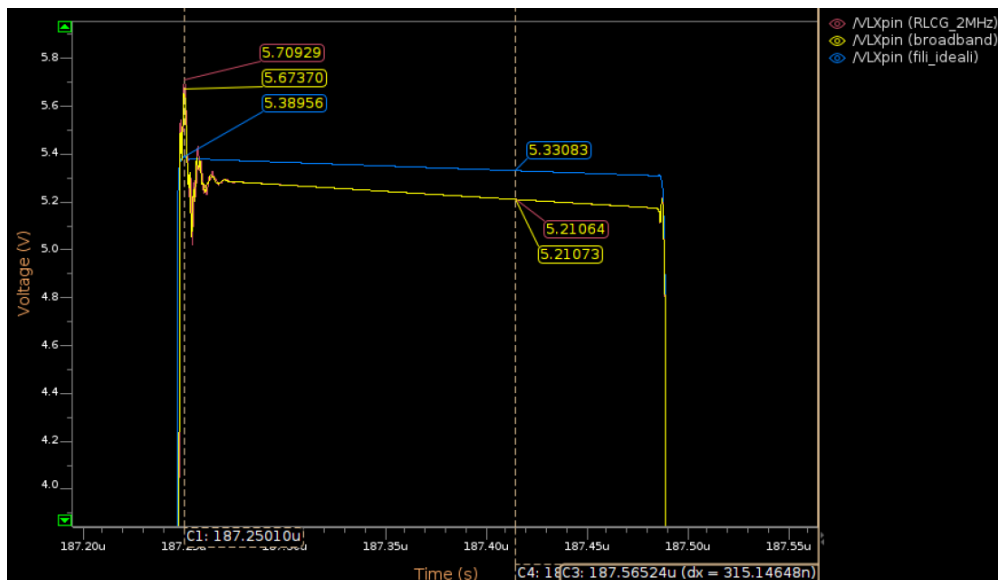


Figura 5.17: VLXpin (pin 9), zoom ulteriore su periodo alto dell'onda quadra per evidenziare il ringing alla fine del fronte di salita.



Figura 5.18: VLXpin (pin 9), zoom sul ringing dell'immagine precedente.

Si nota come sia con la **broadband netlist** che con la **netlist RLCG a 2MHz** inizi a comparire del **ringing** alla fine di ogni fronte di salita. **Questo fenomeno può provocare dei danni al dispositivo perchè la tensione può raggiungere dei picchi anche elevati.** Si ha quindi un andamento più realistico di quello che si ha con la versione dei fili ideali dove viene perso questo comportamento. Inoltre anche il valore in DC risulta essere leggermente differente.

Per valutare nel dettaglio le differenze tra la broadband netlist e il circuito RLCG si è effettuata la **FFT dei segnali**, in modo da capire a quali frequenze, le differenze sono più marcate. In bassa frequenza i valori ottenuti con i due diversi circuiti equivalenti sono molto simili, al valore della frequenza fondamentale totalmente sovrapposti, nella zona vicina a 300MHz però si notano delle differenze maggiori. Occorre ricordare che il tempo di salita dell'onda quadra è di $t_r=1ns$. Le differenze, come detto sono state notate proprio sui tempi di salita, e quindi, nel dominio della frequenza, saranno visibili ad una frequenza di circa $f = 1/(\pi \cdot t_r) = 318MHz$. La figura seguente mostra l'involuppo dello spettro di un'onda trapezoidale dove A rappresenta l'ampiezza del segnale, τ il tempo in cui il segnale assume un valore alto, τ_r rappresenta il tempo di salita e T indica il periodo del segnale :

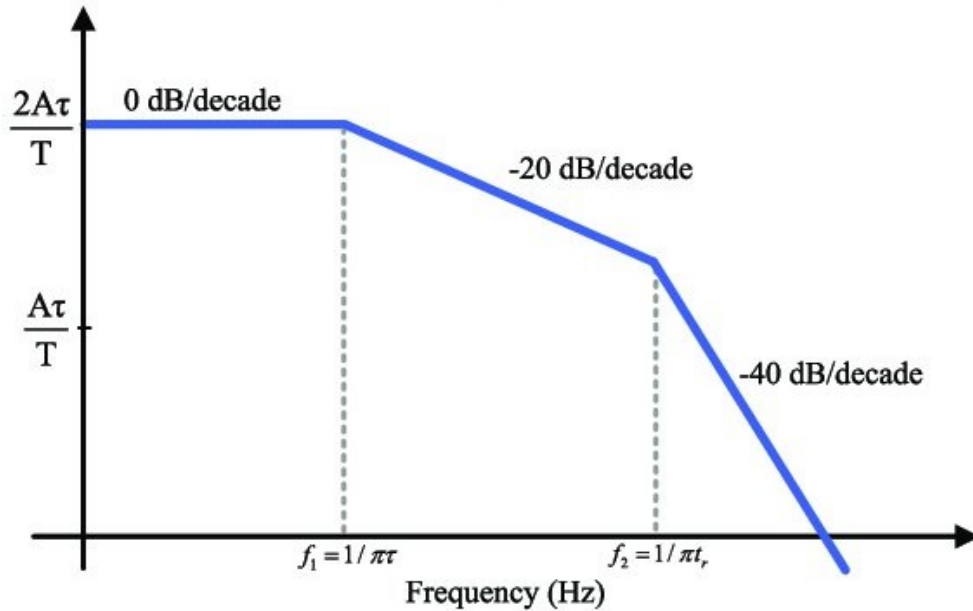


Figura 5.19: Involuppo dello spettro di un'onda trapezoidale.

Si può quindi notare come tra il circuito broadband e la netlist estratta solamente a 2MHz, ad alte frequenze sono presenti delle differenze, a causa delle componenti in frequenza che si creano in particolare dovute ai tempi di salita e discesa del segnale. Il circuito estratto ad una singola frequenza avrà lo stesso valore per tutto il range di frequenze, il circuito broadband invece riproduce l'andamento dovuto allo skin effect considerando quindi il valore corretto dei parassiti anche per quelle componenti in frequenza.

Già dalle simulazioni in AC (Sezione 5.4) si era visto che il filo 5-9 non è ideale: per dimostrarlo anche nel dominio del tempo, **nella cosimulazione riguardante la broadband netlist** (che è la più completa, in quanto tiene conto di tutte le armoniche) sono state confrontate le tensioni sui pin 5 (VLXint) e pin 9 (VLXpin) rispettivamente. La figura seguente mostra le due forme d'onda sovrapposte:

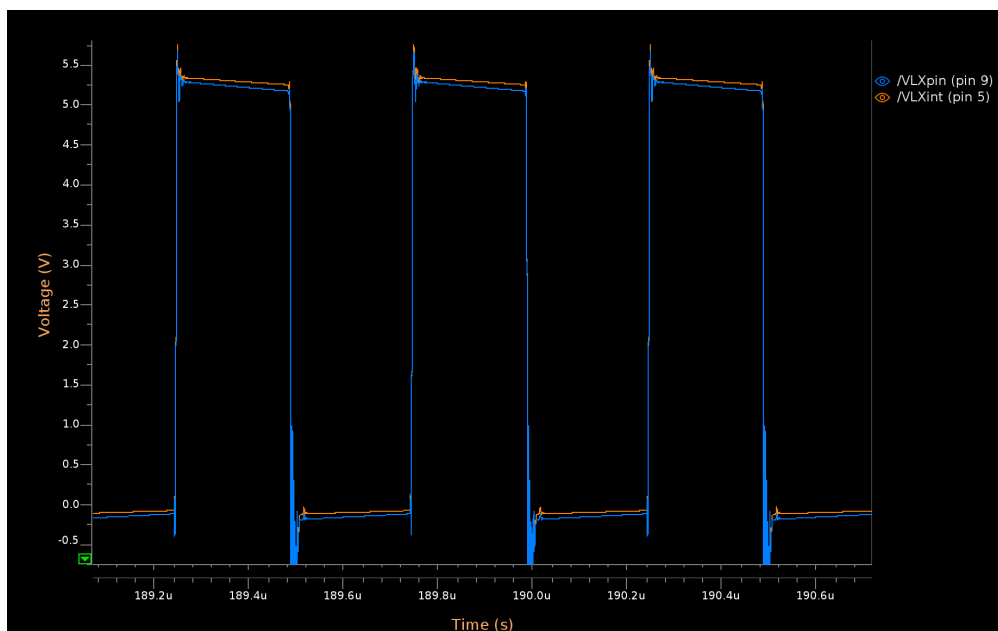


Figura 5.20: Broadband netlist del buck: tensioni sulle due estremità del filo 5-9 (pin 5 e pin 9) a confronto.

Si nota che le tensioni non sono perfettamente sovrapposte come lo sarebbero nel caso in cui il filo fosse ideale: ciò dimostra come il filo 5-9 (relativo al nodo di switching N_x) che collega i due pin **NON è un corto circuito perfetto**, ma presenta dei parametri parassiti R, L, C e G (un'impedenza non nulla). Nel caso ideale si avrebbe $VLX_{pin} = VLX_{int}$, cioè le due curve sarebbero **perfettamente coincidenti**. Si possono fare le considerazioni analoghe anche per tutti gli altri fili.

Capitolo 6

Conclusioni

Questo studio di Tesi ha permesso di capire come i vari tool utilizzati funzionano, come viene performata la simulazione elettromagnetica, la cosimulazione nel dominio del tempo e in che modo vengono costruite le netlist equivalenti (broadband netlist e circuito RLCG ad una certa frequenza). Una volta capiti gli algoritmi interni ed effettuate le cosimulazioni (circuito elettrico assieme ai parassiti) è stato possibile eseguire un confronto dei risultati ottenuti per capire quale degli equivalenti circuitali è più vicino al risultato della simulazione elettromagnetica. Osservando le forme d'onda ottenute nel Capitolo 6, si può notare come la broadband netlist fornisca i risulta migliori, in quando è definita su tutto il range di frequenza di interesse. Il circuito RLCG a 2MHz è valido solamente a 2MHz e per tale motivo offre risultati meno accurati di come li offre la cosimulazione con la broadband netlist. La cosimulazione con i parametri S (utilizzando il component nport di Cadence Virtuoso) ha offerto risultati largamente imprecisi. La presenza dei parassiti (con entrambe le netlist) la presenza del fenomeno del ringing dell'onda quadra rispetto al caso ideale in cui ciò non era presente. Detto quindi che la broadband netlist è la versione circuitala più accurata tra quelle utilizzate, diventa importante menzionare quali potrebbero essere le prospettive future riguardante questo studio di Tesi.

6.1 Prospettive future

Come scritto nei capitoli precedenti, la broadband netlist è definita in un formato che comprende oltre che resistenze e capacità, anche dei generatori pilotati. Quindi questo è ancora un formato behavioral. Questi generatori pilotati, che parametri rappresentano? In che modo è possibile tenerli sotto controllo? Lo step successivo potrebbe essere quello di ricavare un **circuito RLCG equivalente e valido su tutto il range di frequenza**, a partire proprio dalla broadband netlist analizzata ed utilizzata in questa trattazione, in modo da passare da un formato behavioral

ad un formato comprendente solo componenti passivi: R, L, C e G.

Un altro tipo di analisi potrebbe essere quello riguardante la thermal analysis, per capire come questi R, L, C, G parassiti variano al variare della temperatura.

Bibliografia

- [1] Maide Bucolo, Arturo Buscarino, Luigi Fortuna, Carlo Famoso, Mattia Frasca, Antonino Cucuccio, Gaetano Rasconà e Giovanni Vinci. «A Comparative Analysis of Computer-Aided Design Tools for Complex Power Electronics Systems». In: *Energies* 14.22 (2021). ISSN: 1996-1073. DOI: 10.3390/en14227729. URL: <https://www.mdpi.com/1996-1073/14/22/7729> (cit. alle pp. 1, 2, 5, 11, 27).
- [2] Maide Bucolo, Arturo Buscarino, Carlo Famoso, Luigi Fortuna, Mattia Frasca, Antonio Cucuccio, Gaetano Rasconá e Giovanni Vinci. «Model Identification to Validate Printed Circuit Boards for Power Applications: A New Technique». In: *IEEE Access* 10 (2022), pp. 31760–31774. DOI: 10.1109/ACCESS.2022.3160449 (cit. alle pp. 2, 4, 6).
- [3] Sabri Gokay Yildirim, Fatmanur Yurtsevenler, Faik Alan e Ibrahim Kaya. «Analysis of External Effects of Conducted Emission Measurement with Current Probe Method According to CISPR 14-1». In: *2023 7th International Electromagnetic Compatibility Conference (EMC Turkiye)*. 2023, pp. 1–5. DOI: 10.1109/EMCTurkiye59424.2023.10287441 (cit. alle pp. 2, 5).
- [4] Bainan Sun, Kasper Lüthje Jørgensen, Zhe Zhang e Michael A. E. Andersen. «Multi-physic Analysis for GaN Transistor PCB Layout». In: *2019 IEEE Applied Power Electronics Conference and Exposition (APEC)*. 2019, pp. 3407–3413. DOI: 10.1109/APEC.2019.8722166 (cit. a p. 2).
- [5] Kailin Zhang, Miao Cai, Minghui Yun, Lei Song e Daoguo Yang. «Comparison experiment of Parasitic Inductance Extraction of power module based on ANSYS Q3D software». In: *2021 18th China International Forum on Solid State Lighting 2021 7th International Forum on Wide Bandgap Semiconductors (SSLChina: IFWS)*. 2021, pp. 58–60. DOI: 10.1109/SSLChinaIFWS54608.2021.9675224 (cit. a p. 2).
- [6] Luowei Wen, Li Yang, Wensong Yu e Iqbal Husain. «PCB Busbar Optimization for High Power SiC Module». In: *2019 IEEE 7th Workshop on Wide Bandgap Power Devices and Applications (WiPDA)*. 2019, pp. 225–230. DOI: 10.1109/WiPDA46397.2019.8998882 (cit. a p. 3).

-
- [7] Tianjiao Liu, Thomas T. Y. Wong e Z. John Shen. «A Survey on Switching Oscillations in Power Converters». In: *IEEE Journal of Emerging and Selected Topics in Power Electronics* 8.1 (2020), pp. 893–908. DOI: 10.1109/JESTPE.2019.2897764 (cit. a p. 4).
- [8] Piero Triverio, Stefano Grivet-Talocia, Michel S. Nakhla, Flavio G. Canavero e Ramachandra Achar. «Stability, Causality, and Passivity in Electrical Interconnect Models». In: *IEEE Transactions on Advanced Packaging* 30.4 (2007), pp. 795–808. DOI: 10.1109/TADVP.2007.901567 (cit. a p. 9).
- [9] Piero Triverio. «An accurate, robust and intuitive technique to detect causality violations in broadband frequency measurements». In: *2014 IEEE International Symposium on Electromagnetic Compatibility (EMC)*. 2014, pp. 815–820. DOI: 10.1109/ISEMC.2014.6899080 (cit. a p. 9).
- [10] Ansys Q3D Extractor help (cit. alle pp. 11–13, 15, 26, 43, 68, 69).
- [11] Saiyd Ahyoune, Javier Sieiro, Tomás Carrasco, Neus Vidal, José M. López-Villegas, Elisenda Roca e Francisco V. Fernández. «Extending the frequency range of quasi-static electromagnetic solvers». In: *2017 14th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. 2017, pp. 1–4. DOI: 10.1109/SMACD.2017.7981612 (cit. a p. 11).
- [12] Chang Che, Huapeng Zhao, Yuandong Guo, Jun Hu e Hongseok Kim. «Investigation of Segmentation Method for Enhancing High Frequency Simulation Accuracy of Q3D Extractor». In: *2019 IEEE International Conference on Computational Electromagnetics (ICCEM)*. 2019, pp. 1–3. DOI: 10.1109/COMPEM.2019.8779105 (cit. a p. 11).
- [13] *EM Simulation Overview*. Keysight Technologies. URL: <https://docs.keysight.com/ads2024update1/simulation/electromagnetic/em-simulation-overview> (cit. a p. 26).
- [14] *Momentum Overview*. Keysight Technologies. URL: <https://docs.keysight.com/ads2024update1/simulation/electromagnetic/momentum/momentum-overview> (cit. alle pp. 27, 31).
- [15] *Theory of Operation for Momentum*. Keysight Technologies. URL: <https://docs.keysight.com/ads2024update1/simulation/electromagnetic/momentum/theory-of-operation-for-momentum#TheoryofOperationforMomentum-1106962> (cit. alle pp. 27, 30–32).
- [16] *Setting up Momentum Simulations*. Keysight Technologies. URL: <https://docs.keysight.com/ads2024update1/simulation/electromagnetic/momentum/setting-up-momentum-simulations?> (cit. alle pp. 31, 32).

-
- [17] *FEM Overview*. Keysight Technologies. URL: <https://docs.keysight.com/ads2024update1/simulation/electromagnetic/fem/fem-overview> (cit. a p. 33).
- [18] *Theory of Operation for FEM*. Keysight Technologies. URL: <https://docs.keysight.com/ads2024update1/simulation/electromagnetic/fem/theory-of-operation-for-fem> (cit. a p. 33).
- [19] *FDTD Overview*. Keysight Technologies. URL: <https://docs.keysight.com/empro2023/simulation/fdtd-simulation/fdtd-overview> (cit. a p. 40).
- [20] Eldo platform User's Manual (cit. alle pp. 42, 57).
- [21] *About Broadband SPICE Model Generator*. Keysight Technologies. URL: <https://docs.keysight.com/ads2024update1/simulation/model-creation/broadband-spice-model-generator/about-broadband-spice-model-generator> (cit. a p. 60).
- [22] *Creating Broadband SPICE Models*. Keysight Technologies. URL: <https://docs.keysight.com/ads2024update1/simulation/model-creation/broadband-spice-model-generator/creating-broadband-spice-models> (cit. a p. 64).
- [23] *Using Broadband SPICE Models*. Keysight Technologies. URL: <https://docs.keysight.com/ads2024update1/simulation/model-creation/broadband-spice-model-generator/using-broadband-spice-models> (cit. a p. 64).
- [24] B. Gustavsen e A. Semlyen. «Rational approximation of frequency domain responses by vector fitting». In: *IEEE Transactions on Power Delivery* 14.3 (1999), pp. 1052–1061. DOI: 10.1109/61.772353 (cit. a p. 71).
- [25] Jack Davies e Craig Steven Wright. «Using the Singular Value Decomposition to Generate Composite NFTs». In: *2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*. 2023, pp. 1–6. DOI: 10.1109/COINS57856.2023.10189323 (cit. a p. 71).
- [26] B. Gustavsen. «Improving the pole relocating properties of vector fitting». In: *IEEE Transactions on Power Delivery* 21.3 (2006), pp. 1587–1592. DOI: 10.1109/TPWRD.2005.860281 (cit. a p. 71).