

Introduzione a Spice

Daniele Davino - Luigi Verolino

2 - Introduzione a Spice

Indice

Prefazione	5
Capitolo 1 Introduzione	7
Capitolo 2 Reti in regime stazionario	13
Capitolo 3 Reti in regime sinusoidale	31
Capitolo 4 Reti in evoluzione dinamica	45
Capitolo 5 Modelli di dispositivi e circuiti	59
Capitolo 6 Il trasformatore	85
Capitolo 7 Linee di trasmissione	109
Capitolo 8 Dispositivi e modelli a semiconduttore	131
Capitolo 9 Altri programmi ed Opzioni	165

4 - Introduzione a Spice

Prefazione

Questo volume è stato scritto per accrescere il numero di coloro che utilizzano il codice Spice, un programma pensato per la soluzione delle reti elettriche, soprattutto quelle di grandi dimensioni, come i circuiti integrati.

Di buoni manuali in commercio ne esistono già tanti e, pertanto, a prima vista sembrerebbe inutile metterne un altro sul mercato. I prodotti esistenti soffrono, a nostro giudizio, di due gravi difetti: o sono stringatissimi manuali che non consentono di arrivare alle informazioni di cui si ha bisogno, o sono dei manuali troppo grossi, magari in lingua inglese, che, proprio per la loro mole, inibiscono lo studente.

Questa nostra introduzione a Spice vuole essere un sussidio rapido da consultare ma, al tempo stesso, completo per i comandi presentati. Qui lo studente troverà tutto ciò di cui ha bisogno, forse anche di più. Soprattutto troverà una *ricca messe di esempi* che lo aiuteranno, ne siamo sicuri, a comprendere fino in fondo il significato di ogni istruzione adoperata. Anzi, l'approccio usato in questo libro è proprio quello di parlare attraverso degli esempi e di mostrare, come partendo da casi specifici, lo studente possa da solo acquisire la capacità di costruirsi la forma generale dell'istruzione che sta usando.

Infine siamo convinti che, imparando ad usare un simulatore di reti come Spice, l'allievo sarà più motivato a studiare metodi di soluzione e proprietà delle reti elettriche poiché il ‘conforto’ numerico, spesso, è di sprone allo studio della parte teorica.

D.D. - L.V.
Settembre 2000

6 - Introduzione a Spice

Capitolo 1

Introduzione

1.1 Qualche considerazione introduttiva

1.2 La preparazione dei primi files

1.3 Utili informazioni

Sommario

In questo capitolo introduttivo esamineremo quali problemi il programma di simulazione di reti Spice ci aiuta a risolvere, dove e quando cominciò l'avventura di scrivere questo codice, come si preparano, almeno nella denominazione, i files per comunicare con esso. Troverete, infine, alcune utili informazioni sui siti Internet da cui poter prendere le versioni dimostrative (e gratuite) del programma.

1.1 Qualche considerazione introduttiva

La determinazione delle correnti e delle tensioni nei singoli lati di una rete è un problema che ben si presta ad una soluzione numerica. Nel caso lineare si riduce, per esempio, alla semplice inversione di una matrice, mentre in condizioni dinamiche si tratta di risolvere passo dopo passo un sistema di equazioni differenziali ordinarie e lineari, se tali sono i bipoli della rete; in entrambi i casi si ha a che fare con due classici problemi del calcolo numerico.

Per reti non lineari le complicazioni sono maggiori ma non insormontabili e, naturalmente, la bontà della soluzione dipende dalla criticità delle non linearità presenti.

In effetti l'uso di codici numerici si giustifica per due diversi motivi. Da una parte, per le reti lineari molto estese, cioè con un gran numero di nodi e di lati, la soluzione analitica, anche se semplice in linea di principio, può richiedere tempi di elaborazione proibitivi. Dall'altra parte, la presenza di bipoli non lineari può rendere difficile, se non impossibile, una soluzione analitica. Naturalmente la presenza di entrambi questi elementi, come accade nei circuiti integrati di grandi dimensioni, rende spesso la soluzione numerica l'unica via praticabile.

Spice è uno dei codici di calcolo per la soluzione dei circuiti elettrici e elettronici più diffuso, sia in ambiente di ricerca, sia in quello di produzione, e fu sviluppato all'Università della California negli Stati Uniti, e reso disponibile al pubblico, nel 1975. Il suo nome è formato dalle iniziali di alcune parole inglesi (Simulation Program with Integrated Circuit Emphasis) che significano Simulatore di circuiti con particolare enfasi per quelli integrati. Serve, dunque, a determinare le tensioni e le correnti nei singoli lati di una rete elettrica oppure ad effettuare prove ‘sperimentali’ su di un circuito senza realizzarlo in laboratorio, ma solo sul computer. Non è l'unico codice che realizza queste funzioni, poiché esistono molti altri codici numerici che affrontano egregiamente questo problema e ne danno una soluzione altrettanto soddisfacente.

I motivi della grande diffusione del codice Spice vanno ricercati, naturalmente, nella sue qualità e funzionalità, ma anche nella intelligente politica sostenuta dalla Università della California, che ha consentito la diffusione gratuita del prodotto, in una versione ridotta, per scopi educativi. La differenza tra la versione completa e la ridotta è nella consistenza della biblioteca di componenti prevista: nella versione completa sono contemplate le caratteristiche di moltissimi componenti diversi (circa 5000), invece la versione distribuita gratuitamente ne prevede solo pochi (circa 300), non costituendo, però, una seria limitazione. Anche le successive versioni di Spice, elaborate da società di software commerciali, prevedono generalmente una edizione didattica gratuita.

Spice è utile per un certo numero di motivi che, qui di seguito, elenchiamo schematicamente:

- può essere una utile ed istruttiva verifica degli esercizi di Elettrotecnica svolti manualmente;
- consente di valutare il comportamento di reti non lineari altrimenti difficili da trattare attraverso una analisi manuale;
- permette di studiare l'evoluzione transitoria di un circuito senza dover risolvere equazioni differenziali;
- costituisce uno standard nel mondo dell'Elettronica;
- potrebbe essere utile in futuro per il vostro lavoro.

1.2 La preparazione dei primi files

Per poter interagire con Spice bisogna acquisire un linguaggio specifico e, come per tutti gli altri linguaggi, il modo più efficiente per farlo è di provarlo a ‘parlare’. Una volta apprese le basi della conoscenza pratica, lo studio della ‘grammatica’, cioè del ‘manuale’, consentirà di approfondire la conoscenza del linguaggio in tutte le sue possibilità comunicative.

È doveroso sottolineare, però, che le grandi potenzialità di Spice emergono chiaramente solo quando si affrontano circuiti complessi ed in presenza di non linearità dei componenti. Dovendoci limitare ai semplici circuiti trattati in un testo introduttivo come il nostro, tali potenzialità non sempre verranno sfruttate; si potrà avere a volte l'impressione di sparare ad una mosca con il proverbiale cannone. Altre volte invece saremo costretti a delle piccole forzature per indurre Spice a fornire risposte indubbiamente molto elementari. Si ricordi che Spice è stato progettato, come dice l'acronimo, con ‘particolare enfasi ai circuiti integrati’.

Possiamo ora introdurre un po’ più dettagliatamente Spice, evidenziando innanzitutto che del programma esistono, oramai, numerose versioni per diverse piattaforme di computer. Potete trovare, ad esempio, versioni di Spice per PC, Macintosh o per i grandi sistemi funzionanti con il sistema operativo UNIX. Se paragonassimo Spice a una lingua, potremmo dire che esistono diversi ‘dialetti’, il più famoso dei quali è PSpice, un prodotto dalla Microsim Corporation di cui esistono versioni per ogni personal computer o work station. In questo caso ci troviamo di fronte ad un prodotto commerciale, di cui, per fortuna, esiste una versione ‘evaluation’, ovvero una versione gratuita con alcune limitazioni, che per i nostri scopi attuali sono irrilevanti. Questa versione del programma è stata

realizzata apposta per gli studenti (è anche detta ‘student edition’) per cui potete installarla senza indugio sui vostri personal computer!

Nel seguito ci riferiremo ad una ‘grammatica’ di base di Spice esportabile su ogni versione, costituita da un insieme di comandi comune a quasi tutte le versioni del programma. Sappiate che i grafici che incontrerete nel seguito sono stati ottenuti con Pspice 6.0 (evaluation version) per Macintosh.

Per utilizzare Spice occorre percorrere fondamentalmente tre passi:

- preparare un *file di ingresso* che spieghi come è strutturato il circuito, quali siano i valori che caratterizzano ciascun componente e il tipo di analisi richiesta (continua, alternata, transitoria);
- far esaminare questo file al programma Spice che ne produrrà uno nuovo, detto *file di uscita*, che contiene i principali risultati;
- elaborare, eventualmente, il file prodotto da Spice per mostrare in forma grafica alcuni dei risultati ottenuti.

Il file di ingresso deve avere l'estensione ‘.cir’ (non in tutte le versioni strettamente necessaria): ciò vuol dire che se si decide di chiamare con il nome ‘Esempio’ il nostro file di ingresso, sul calcolatore dovrà essere memorizzato come ‘Esempio.cir’ affinché il compilatore Spice lo riconosca.

La prima riga del file contiene necessariamente il titolo della analisi, ovvero dell'esercizio. Perciò se non volete usare un titolo dovete lasciare libera la prima riga, altrimenti la vostra prima riga di programma diverrà ... il titolo dell'esercizio!

Esaminiamo l'esempio qui di seguito proposto.

Esempio 1.1

* Primo esempio di semplice circuito

R1 1 2 8

R2 2 0 4

VE 1 0 2

.END

Le istruzioni vi saranno più chiare nel seguito ma, per il momento, potete cominciare a intuire che tra i nodi 1 e 2 vi è un resistore di 8 ; tra i nodi 2 e 0 vi è un altro resistore di 4 ; infine tra i nodi 1 e 0 vi è un generatore di tensione di

2 V. Notate pura la seconda riga, quella che inizia con l'asterisco, che rappresenta un commento.

Altre informazioni riguardanti le modalità di utilizzo saranno fornite direttamente discutendo alcuni esempi nei capitoli seguenti.

1.3 Utili informazioni

Per coloro che non disponessero di una versione del programma Spice, esso può essere gratuitamente preso dal sito Internet (americano)

www.orcad.com .

Tuttavia, la versione Spice presente all'indirizzo indicato, è molto grande e potrebbe richiedere parecchio tempo per essere scaricata. Per evitare questo problema, abbiamo messo sul sito del nostro editore alcune versioni Spice per i principali sistemi operativi. Provate, dunque, all'indirizzo

www.liguori.it/materiali

e troverete delle versioni Spice più piccole, ma sicuramente adatte alle vostre esigenze e a provare tutti gli esempi sviluppati in questo testo.

12 - Introduzione a Spice

Capitolo 2

Reti in regime stazionario

2.1 Introduzione

2.2 Un primo esempio

2.3 Resistenze e generatori indipendenti

2.4 Generatore usato come amperometro

2.5 Potenza erogata dai generatori

2.6 L’istruzione ‘.DC’

2.7 L’istruzione ‘.TF’

2.8 Altri esempi

Sommario

Introdurremo in questo capitolo i primi comandi Spice per simulare reti in regime stazionario. Esamineremo, alla fine del capitolo, anche l’istruzione che consente di ottenere il circuito equivalente secondo Thévenin.

2.1 Introduzione

In questo primo capitolo mostreremo come si possa realizzare la simulazione di una rete funzionante in regime stazionario. Preferiamo iniziare con i circuiti in corrente continua poiché, data la loro relativa semplicità, è possibile introdurre in maniera più comprensibile i primi comandi del simulatore.

2.2 Un primo esempio

La prima domanda che sorge nell'utilizzo di un simulatore per circuiti è come inserire la topologia della rete e le caratteristiche dei rami. La topologia viene inserita tramite i *nodi*: a differenza dei classici risultati della teoria dei circuiti, in Spice, ogni terminale di bipolo è considerato come un nodo e va, quindi, numerato.

Per essere il più possibile chiari, consideriamo il segmento di rete costituito dalle due resistenze R_1 e R_2 mostrate in Figura 2.1.

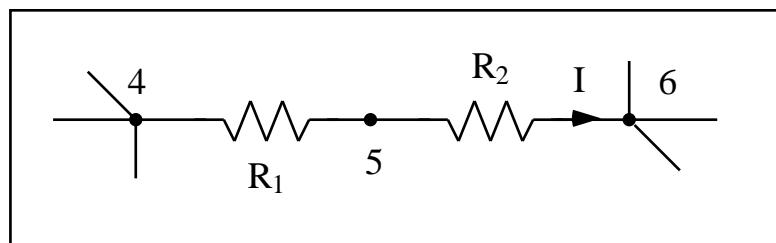


Figura 2.1: due resistenze in serie.

Pur essendo chiaro che esse sono in serie, e pertanto riducibili ad un'unica resistenza, per la codifica Spice è necessario introdurre il nodo 5.

Una volta stabilito da quanti nodi è costituita una rete, è necessario numerarli. Uno dei nodi, scelto a caso, va numerato con lo zero (0), e sarà considerato come il nodo di riferimento per i potenziali. Tutti gli altri possono essere etichettati con dei numeri interi positivi qualsiasi. È ovvio che utilizzare qualche criterio nella scelta dei nodi semplifica la leggibilità e l'eventuale verifica del circuito.

Infine, bisogna specificare le caratteristiche dei bipoli.

Per cominciare a esemplificare, consideriamo il circuito mostrato in Figura 2.2, e supponiamo di voler determinare la corrente I che circola attraverso il generatore di tensione.

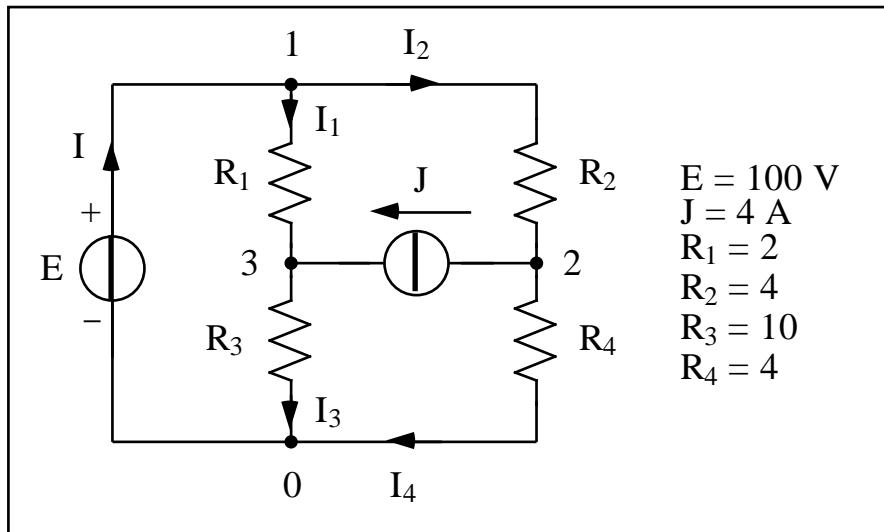


Figura 2.2: primo esempio di circuito.

Il problema può, ad esempio, essere risolto per mezzo delle istruzioni che seguono.

Esempio 2.1

* Primo esempio di circuito in continua

```
R1    1    3    2
R2    1    2    4
R3    3    0    10
R4    2    0    4
VE    1    0    100
IJ      2    3    4
.END
```

La *prima linea* assegna semplicemente il nome al pacchetto di istruzioni che si conclude con il comando ‘.END’. Quest’ultimo costituisce un primo esempio di istruzione di controllo, che in Spice è sempre individuata dal punto iniziale.

In particolare, l’istruzione ‘.END’ informa il compilatore Spice che la lettura è finita, o, se si vuole, che non vi sono altre istruzioni da esaminare.

La *seconda linea* è un commento, individuato dall’asterisco iniziale, che viene semplicemente ignorato da Spice.

Le *righe successive* descrivono il circuito individuandone i bipoli presenti tra i diversi nodi, numerati a partire dal nodo 0 che, necessariamente presente, è il

nodo scelto quale riferimento per i potenziali. Così la prima riga afferma che tra il nodo 1 e il nodo 3 è presente un resistore di 2 : la lettera R nella prima posizione del nome scelto per individuarlo ne è testimone. Gli altri caratteri del nome sono, invece, del tutto arbitrari. Per quanto riguarda le unità di misura, se non diversamente specificato, le resistenze si intendono assegnate in ; altrimenti, come per tutte le altre grandezze che introdurremo, si possono utilizzare multipli e sottomultipli, secondo il simbolismo specificato nella tabella mostrata in basso, dove naturalmente, ‘1e-3’ (anche maiuscolo ‘1E-3’) indica il valore 10^{-3} .

Simbolo	Esponente
f	1e-15
p	1e-12
n	1e-9
u	1e-6
m	1e-3
k	1e+3
meg	1e+6
g	1e+9
t	1e+12

Quindi, una resistenza di $R = 2000$ può essere data a Spice nelle tre forme equivalenti: 2000, 2e3, 2k.

Notiamo che Spice non distingue mai i caratteri minuscoli da quelli maiuscoli.

L'ordine con cui vengono indicati i due nodi estremi del bipolo non è indifferente; esso specifica, infatti, l'orientazione scelta per il ramo, che va sempre dal primo nodo menzionato al secondo.

La *penultima riga* del pacchetto di istruzioni, prima del ‘.END’, comunica a Spice che tra il nodo 1 ed il nodo 0, nel verso precisato dall'ordine, è inserito un generatore ideale che eroga una tensione di 100V; esso è individuato dalla lettera V al primo posto del nome scelto per indicare il bipolo. Il nome di un generatore ideale di corrente dovrà, ovviamente, cominciare con la lettera I. Notiamo che per il generatore ideale di corrente del nostro esempio la corrente scorre dal nodo 2 al nodo 3.

Fatta eccezione per la prima e l'ultima istruzione, e per poche altre che incontreremo in seguito, l'ordine delle diverse righe è del tutto arbitrario.

Se forniamo a Spice un tale pacchetto di istruzioni, esso produce in risposta un file, denominato ‘Esempio1.out’, dal contenuto mostrato nel riquadro che segue.

```
**** 11/22/99 10:20:02 **** Evaluation PSpice (July 1993) ****
Esempio 2.1
**** CIRCUIT DESCRIPTION
*****
* Primo esempio di circuito in continua
R1      1      3      2
R2      1      2      4
R3      3      0      10
R4      2      0      4
VE      1      0      100
IJ      2      3      4
.END
**** 11/22/99 10:20:02 **** Evaluation PSpice (July 1993) ****
Esempio 1
**** SMALL SIGNAL BIAS SOLUTION    TEMPERATURE = 27.000 DEG C
*****
NODE VOLTAGE NODE VOLTAGE NODE VOLTAGE NODE VOLTAGE NODE VOLTAGE
( 1) 100.0000 ( 2) 42.0000 ( 3) 90.0000
VOLTAGE SOURCE CURRENTS
NAME      CURRENT
VE        -1.950E+01
TOTAL POWER DISSIPATION 1.95E+03 WATTS
JOB CONCLUDED
TOTAL JOB TIME .17
```

Nella prima parte della risposta, viene riproposto il file in ingresso; ciò può essere evitato, con una opportuna istruzione, ma in generale non conviene farlo.

Nella seconda parte, sotto il titolo ‘SMALL SIGNAL BIAS SOLUTION’ vengono forniti i valori dei potenziali associati ai nodi della rete. Il motivo di una tale intitolazione sta nel fatto che Spice valuta sempre il punto di lavoro di tutti i componenti, dovuto alla presenza dei generatori in continua, prima di valutare l’eventuale dinamica prodotta da segnali di piccola ampiezza intorno a tali punti di lavoro. Per questo motivo non è stato necessario, nel nostro caso, indicare alcuna specifica richiesta nel file di ingresso: la valutazione del *punto di lavoro* è sempre fornita.

In pratica si può immaginare che Spice abbia applicato il metodo dei potenziali ai nodi ed invertito la matrice corrispondente. Tra le altre informazioni fornite, che non commenteremo perché di intuitiva interpretazione, si nota la presenza del

valore della corrente erogata dal generatore di tensione: Spice calcola sempre tali correnti allo scopo di valutarne la potenza erogata.

2.3 Resistenze e generatori indipendenti

Riassumiamo, dunque, le tre istruzioni già mostrate per codificare resistori, generatori indipendenti di tensione e di corrente.

Il formato per identificare un *resistore* di 100 , connesso, ad esempio, tra i nodi 2 e 3, è

RAB 2 3 100

in cui RAB è il nome scelto (Figura 2.3). Il nome può contenere al massimo una combinazione di 8 lettere e numeri. La prima lettera del nome individua il tipo di bipolo e per un resistore deve essere una R. L'ordine con cui scriviamo i nodi non è importante.

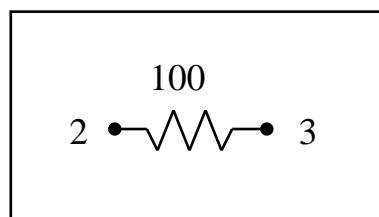


Figura 2.3: generico resistore.

Un *generatore indipendente di tensione* di 35 V, di nome SORG e posto tra i nodi 7 e 8, è descritto dall'istruzione (Figura 2.4)

VSORG 7 8 DC 35

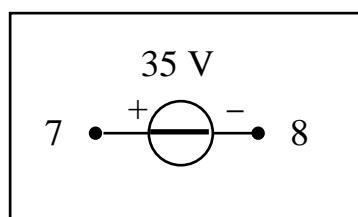


Figura 2.4: generatore indipendente di tensione.

La prima lettera del nome deve essere una V e 'DC' indica il tipo di sorgente (opzionale in regime stazionario). Se l'ordine dei nodi viene invertito nella descrizione della sorgente, il valore va cambiato di segno

VSORG 8 7 DC - 35

Analogamente un *generatore indipendente di corrente* tra i nodi 1 e 5, che eroga una corrente di 20 A, è descritto dall'istruzione (Figura 2.5)

IS2 5 1 DC 20

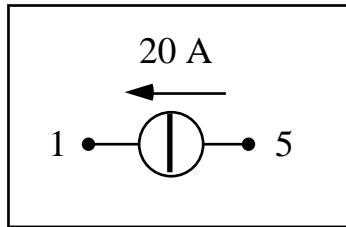


Figura 2.5: generatore indipendente di corrente.

Qui IS2 è il nome scelto per la sorgente e la prima lettera del nome deve essere una I. Di nuovo si presti attenzione all'ordine in cui sono stati posti i nodi.

2.4 Generatore usato come amperometro

Esiste un metodo molto semplice per ottenere la valutazione diretta nel file di uscita della corrente in un ramo: basta inserire in quel ramo un generatore ideale di tensione che eroghi una tensione nulla, che, ovviamente, non influenza il funzionamento del circuito.

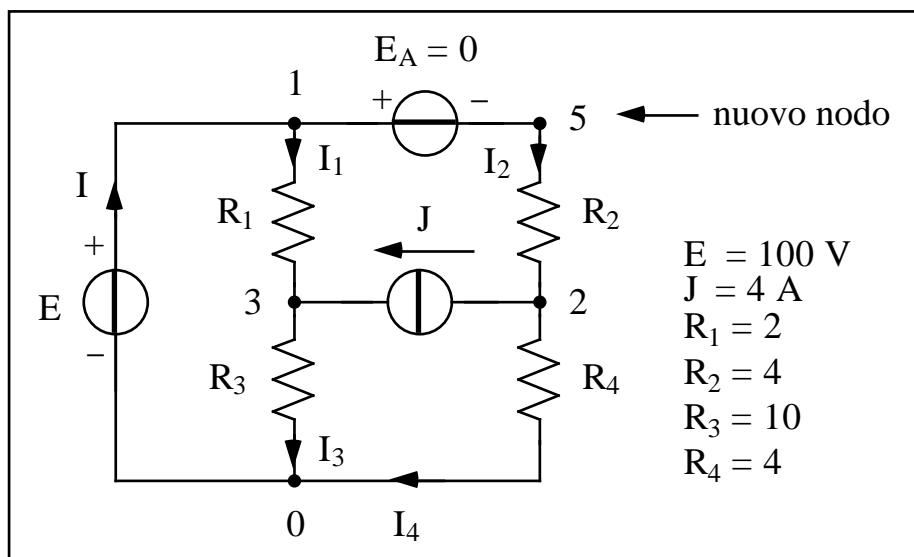


Figura 2.6: generatore usato come amperometro.

Questo generatore svolge il ruolo di un vero e proprio *amperometro*. Così, ad esempio, se modifichiamo il file di ingresso come mostrato in Figura 2.6, possiamo ottenere la corrente che circola nel resistore R_2 . Si noti che per inserire il generatore si è dovuto introdurre un altro nodo.

In queste condizioni nella risposta di Spice troviamo anche

$$I_1 = I - I_2 = (19.5 - 14.5) \text{ A} = 5 \text{ A},$$

come potete controllare mediante il file seguente.

Esempio 2.2

* Generatore usato come amperometro

R1	1	3	2
R2	5	2	4
R3	3	0	10
R4	2	0	4
VE	1	0	100
IJ	2	3	4
VA	1	5	0

.END

Notate che la corrente I che interessa il generatore di tensione E viene fornita con il segno cambiato, rispetto a quella indicata in Figura 2.6, dato che il compilatore fa, su ogni bipolo, la convenzione dell'utilizzatore. Per lo stesso motivo, la corrente che scorre in VA è positiva, dato che il generatore si lascia attraversare dalla corrente come un qualsiasi bipolo passivo.

VOLTAGE SOURCE CURRENTS	
NAME	CURRENT
VE	-1.950E+01
VA	1.450E+01

2.5 Potenza erogata dai generatori

Vale la pena notare che l'indicazione del file di uscita 'TOTAL POWER DISSIPATION' non rappresenta la potenza assorbita dai resistori della rete (come la dizione potrebbe erroneamente suggerire), ma la potenza erogata dai *soli* generatori di tensione. Nell'esempio riportato, l'indicazione è relativa alla sola

$$P_E = E I = 1.95 \text{ kW} .$$

Per cui, per ottenere la potenza assorbita da tutti i resistori, *mancava* la potenza erogata dai generatori indipendenti di corrente.

2.6 L'istruzione '.DC'

Fino ad ora abbiamo imparato a risolvere una rete in continua con Spice e a ricavare alcune informazioni come la potenza erogata dai generatori indipendenti di tensione.

L'istruzione '.DC' permette di risolvere una rete quando un generatore indipendente varia entro un intervallo di valori. Consideriamo l'esempio 2.1 in cui inseriamo l'istruzione

```
.DC VE 90 110 1
```

in tal caso l'analisi in corrente continua sarà effettuata considerando il generatore di tensione VE che assume valori tra 90 e 110V con incrementi di 1V.

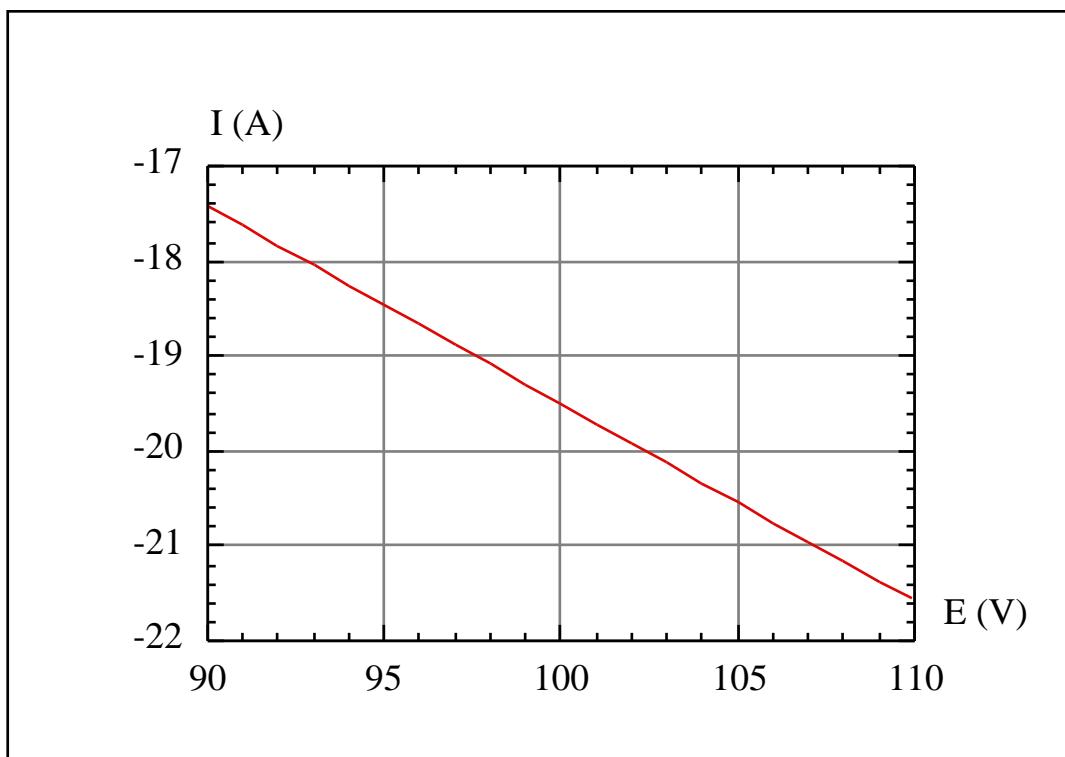


Figura 2.7: analisi di sensibilità.

Per ogni valore del generatore VE potete richiedere, mediante l'istruzione '.PRINT DC' di cui parleremo tra poco, la stampa di una o più grandezze di

interesse e otterrete nel file di uscita una tabella in cui la prima colonna contiene i valori assunti dal generatore che varia e le successive contengono le grandezze richieste. La Figura 2.7 mostra questa analisi di sensibilità che, per il caso in esame, fornisce in uscita un grafico rettilineo.

Si noti che non conta il valore (100V nell'esempio 2.1) che compare nella riga in cui definiamo il generatore che faremo variare in quanto esso sarà sostituito dal valore di voltaggio imposto dall'istruzione '.DC'. Infatti, è buona norma porre tale valore a zero proprio per ricordare che il generatore è incluso in un istruzione '.DC'.

L'istruzione '.DC' permette anche di far variare due generatori indipendenti contemporaneamente; questa possibilità è particolarmente utile per ricavare le caratteristiche di uscita di un transistore (come vedremo nel seguito). La riga

```
.DC VE 90 110 5 IJ 0 4 1
```

consente di variare *anche* il generatore IJ nell'intervallo (0 ÷ 4) A con incrementi di 1 A.

Come dicevamo prima, la stampa dei risultati dell'analisi .DC si ottiene mediante un'istruzione del tipo:

```
.PRINT DC I(R2)
```

con la quale otteniamo nel file .OUT la stampa di una tabella in cui la prima colonna contiene i valori assunti dal generatore e la seconda colonna è la corrente che scorre nel resistore R2. Ricordiamo che Spice usa la convenzione dell'utilizzatore, per cui la corrente I(R2) nella tabella sarà positiva se la rete è tale da far scorrere la corrente dal primo al secondo nodo nella riga che definisce il resistore R2.

Nel caso in cui variano contemporaneamente due generatori, il file di uscita conterrà ancora una tabella in cui i valori assunti dal generatore sono ripetuti per ogni valore assunto dal secondo generatore. Purtroppo questi ultimi non sono forniti esplicitamente per cui bisognerà aggiungere la richiesta mediante una istruzione del tipo

```
.PRINT DC I(IJ) I(R2)
```

Questa istruzione, applicata all'esempio 2.1, fornisce la seguente tabella (abbiamo riportato solo una parte della tabella, relativa ai primi due valori assunti da IJ)

VE	I(IJ)	I(R2)
9.000E+01	0.000E+00	1.125E+01
9.500E+01	0.000E+00	1.188E+01
1.000E+02	0.000E+00	1.250E+01
1.050E+02	0.000E+00	1.313E+01
1.100E+02	0.000E+00	1.375E+01
9.000E+01	1.000E+00	1.175E+01
9.500E+01	1.000E+00	1.238E+01
1.000E+02	1.000E+00	1.300E+01
1.050E+02	1.000E+00	1.363E+01
1.100E+02	1.000E+00	1.425E+01

2.7 L'istruzione '.TF'

Sappiamo la grande importanza che riveste il circuito equivalente di Thévenin (o di Norton) nella soluzione delle reti lineari. In Spice, l'istruzione '.TF' fornisce, tra le altre cose, la resistenza equivalente che occorre per la realizzazione di tale circuito. Letteralmente '.TF' è l'acronimo di 'Transfer Function', una istruzione che permette di calcolare la funzione di trasferimento tra una grandezza di uscita, che può essere una tensione di lato (oppure una corrente di ramo che attraversa un generatore indipendente di tensione) e una di ingresso che deve essere un generatore indipendente di tensione o di corrente. Per comprendere sino in fondo come funziona, consideriamo di nuovo la rete di Figura 2.2. Supponiamo di voler conoscere la corrente I_4 e di voler usare il teorema di Thévenin, come suggerito dalla Figura 2.8.

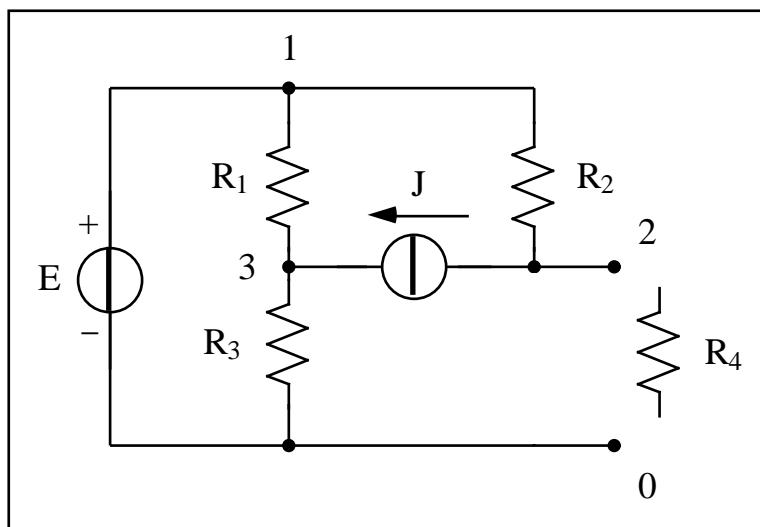


Figura 2.8: applicazione del comando '.TF'.

Per determinare questa corrente vogliamo costruire il circuito equivalente di Thévenin della rete visto dai morsetti 2 e 0. Simuliamo, allora, la rete mostrata in Figura 2.8 in cui il ramo contenente la resistenza R_4 è stato rimosso ed usiamo la nuova istruzione ‘.TF’, secondo il listato qui di seguito riportato.

Esempio 2.3
 * Uso dell’istruzione ‘.TF’
 R1 1 3 2
 R2 1 2 4
 R3 3 0 10
 VE 1 0 100
 IJ 2 3 4
 .TF V(2,0) VE
 .END

Notate che dal listato è stata rimossa la linea che introduceva la resistenza R_4 . L’istruzione ‘.TF’ richiede a Spice di valutare il rapporto tra la tensione di lato $V(2,0)$, grandezza di uscita, e il generatore indipendente di tensione VE , grandezza di ingresso. Nel file di uscita ritroviamo alcune informazioni.

**** SMALL-SIGNAL CHARACTERISTICS
 $V(2,0)/VE = 1.000E+00$
 INPUT RESISTANCE AT $VE = 1.200E-01$
 OUTPUT RESISTANCE AT $V(2,0) = 4.000E+00$

Fate attenzione al fatto che il rapporto tra la variabile di uscita e quella di ingresso viene calcolato considerando la rete in cui gli altri eventuali generatori di tensione sono stati cortocircuitati e i generatori di corrente sono stati aperti, eccezion fatta per il generatore che definisce la grandezza di ingresso; questa informazione è dunque inutile ai nostri scopi attuali.

La seconda riga definisce la resistenza di ingresso vista attraverso i nodi, che definiscono il generatore di ingresso e, ancora una volta, questa informazione è inutile al circuito di Thévenin. La *resistenza di uscita* tra i morsetti, o nodi, che definiscono la grandezza di uscita è calcolata considerando la rete con tutti i generatori di tensione cortocircuitati e tutti i generatori di corrente aperti. Questa, finalmente, è la resistenza equivalente R_0 del circuito di Thévenin!

Infine, dall’analisi in continua abbiamo tutti i potenziali ai nodi per cui la tensione a vuoto del circuito di Thévenin sarà la differenza di potenziale tra i nodi che

abbiamo scelto come uscita. Nel nostro caso ricordiamo che l'analisi fornisce $E_0 = V(2) = 84 \text{ V}$. Con ciò disponiamo dei due parametri che ci consentono di realizzare il circuito equivalente (Figura 2.9) e, quindi, calcolare la corrente desiderata:

$$I_4 = \frac{E_0}{R_0 + R_4} = 10.5 \text{ A} .$$

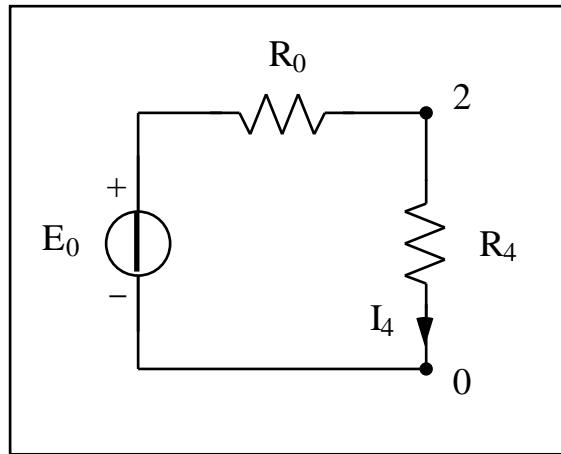


Figura 2.9: circuito equivalente secondo Thévenin.

Notiamo che questo metodo può essere applicato a qualsiasi ramo della rete che non lasci nodi isolati, cioè che non vi siano bipoli con uno dei due nodi non connesso ad altri bipoli della rete. In tal caso, onde evitare errori, si applicherà Thévenin a tutto il ramo in modo da non lasciare bipoli isolati.

2.8 Altri esempi

In questo paragrafo conclusivo riportiamo alcuni esempi risolti sui quali potete verificare se avete ben compreso quanto detto in precedenza.

La Figura 2.10 mostra un esempio di rete con più generatori.

Si vuole determinare la corrente I che passa attraverso il generatore di tensione E_1 , assumendo $E_1 = 12 \text{ V}$, $E_2 = 6 \text{ V}$, $J = 5 \text{ A}$, $R_1 = 15 \Omega$, $R_2 = 3 \Omega$, $R_3 = 6 \Omega$.

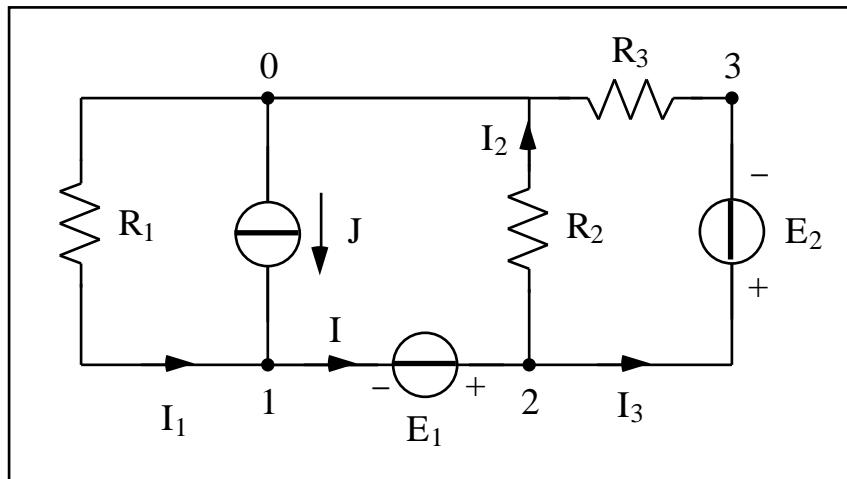


Figura 2.10: circuito con più generatori.

Ecco il listato Spice che risolve il problema.

Esempio 2.4

* Circuito con più generatori

```
R1    0    1    15
R2    2    0    3
R3    3    0    6
VE1    2    1    DC    12
VE2    2    3    DC    6
IJ    0    1    DC    5
.END
```

Risposta: $I = 5 \text{ A}$.

Ripetete l'esempio precedente adoperando l'istruzione '.TF'.

Ancora un esempio che consideri una rete un po' più grande.

Per la rete di Figura 2.11 calcolare le tensioni V_1 , V_2 , V_3 , V_4 , assumendo che $J = 2 \text{ A}$, $R_1 = 150 \Omega$, $R_2 = 100 \Omega$, $R_3 = 150 \Omega$, $R_4 = 100 \Omega$, $R_5 = 25 \Omega$, $R_6 = 50 \Omega$, $R_7 = R_8 = 20 \Omega$.

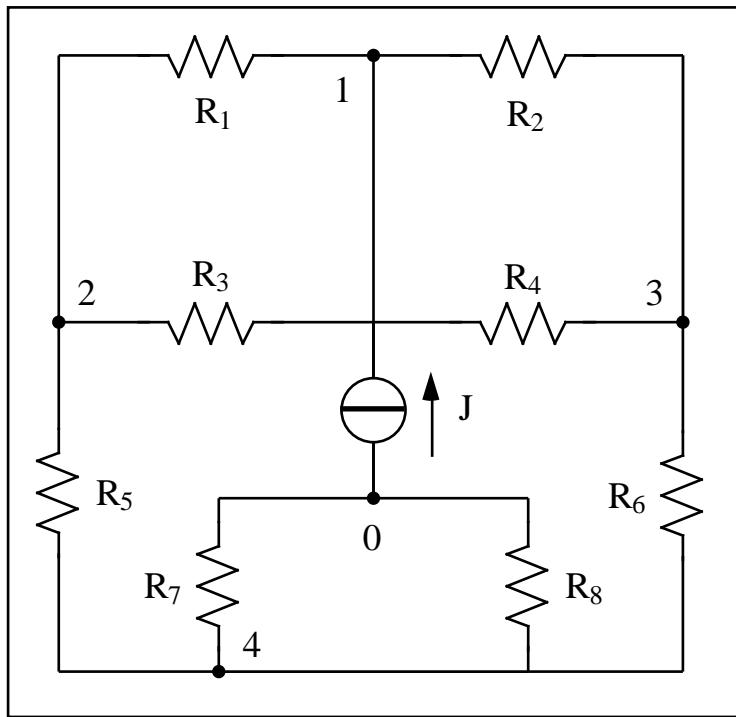


Figura 2.11: un esempio di rete con diversi resistori.

Ecco il listato Spice.

Esercizio 2.5

* Rete con diversi resistori

R1	2	1	150
R2	3	1	100
R3	2	1	150
R4	3	1	100
R5	2	4	25
R6	3	4	50
R7	4	0	20
R8	4	0	20
IJ	0	1	DC 2
.END			

Risposta : $V_1 = 120 \text{ V}$, $V_2 = 45 \text{ V}$, $V_3 = 70 \text{ V}$, $V_4 = 20 \text{ V}$.

Ancora, determinate la corrente I della rete mostrata in Figura 2.12, assumendo che $E = 50 \text{ V}$, $J = 1 \text{ A}$, $R_1 = 80$, $R_2 = 50$, $R_3 = 40$, $R_4 = 800$, $R_5 = 200$.

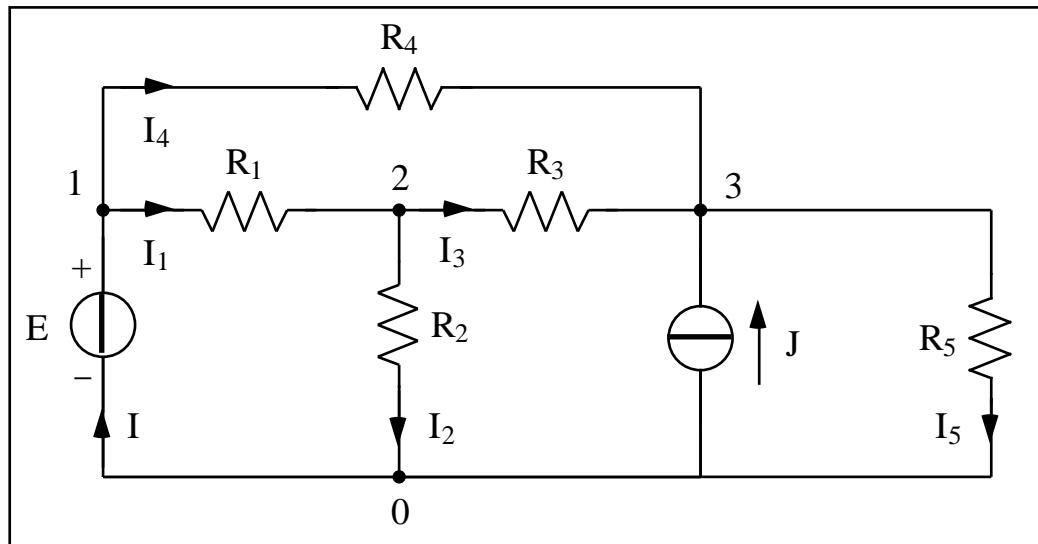


Figura 2.12: esempio finale.

Il listato Spice soluzione del problema proposto.

Esercizio 2.6

* Circuito con due generatori

R1	1	2	80
R2	2	0	50
R3	2	3	40
R4	1	3	800
R5	3	0	200
VE	1	0	DC 50
IJ	0	3	DC 1

.END

Risposta: $I = 0.114$ A.

Ancora una volta, vi consigliamo di controllare il risultato usando l'istruzione '.TF'.

Prima di concludere è opportuno fare un cenno alle reti spaziali, quelle, per intenderci, non planari. Immaginate di avere un cubo di resistenze, tutte uguali a R (sulle diagonali non ci sono resistori). Verificate, per mezzo di Spice, che la resistenza tra due nodi contigui vale

$$R_1 = \frac{7}{12} R .$$

Il listato che segue può darvi qualche buon suggerimento su come procedere.

Esempio 2.7

* Cubo di resistenze uguali

R01 0 1 24

R12 1 2 24

R23 2 3 24

R03 0 3 24

R45 4 5 24

R56 5 6 24

R67 6 7 24

R47 4 7 24

R04 0 4 24

R15 1 5 24

R26 2 6 24

R37 3 7 24

V0 0 1 1

.END

Provate, poi, a verificare che la resistenza equivalente tra due nodi posti diagonalmente sulla stessa faccia e due nodi posti sulla diagonale del cubo valgono, rispettivamente

$$R_2 = \frac{3}{4} R, \quad R_3 = \frac{5}{6} R.$$

Capitolo 3

Reti in regime sinusoidale

3.1 Introduzione

3.2 Un esempio introduttivo

3.3 Il condensatore e l'induttore

3.4 Sintassi per l'analisi in frequenza

3.5 Un classico esempio di analisi in frequenza

3.6 Esempi

Sommario

L'analisi delle reti in regime sinusoidale costituisce l'oggetto del presente capitolo. Dopo aver introdotto la sintassi fondamentale per simulare un condensatore e un induttore, introdurremo l'analisi in frequenza di una rete che operi in regime sinusoidale.

3.1 Introduzione

Nel corso di questo capitolo ci occuperemo dell'analisi delle reti funzionanti in regime sinusoidale. Per raggiungere questo obiettivo introdurremo, per la prima volta, le istruzioni per la codifica degli induttori e dei condensatori e mostreremo come si possa ottenere l'analisi in frequenza esaminando alcuni esempi di reti.

Mostreremo pure come si possano ottenere i diagrammi del modulo e della fase (oppure della parte reale e di quella immaginaria) di una certa tensione o di una corrente al variare della frequenza.

3.2 Un esempio introduttivo

Per cominciare, consideriamo il circuito, operante in regime sinusoidale, mostrato in Figura 3.1.

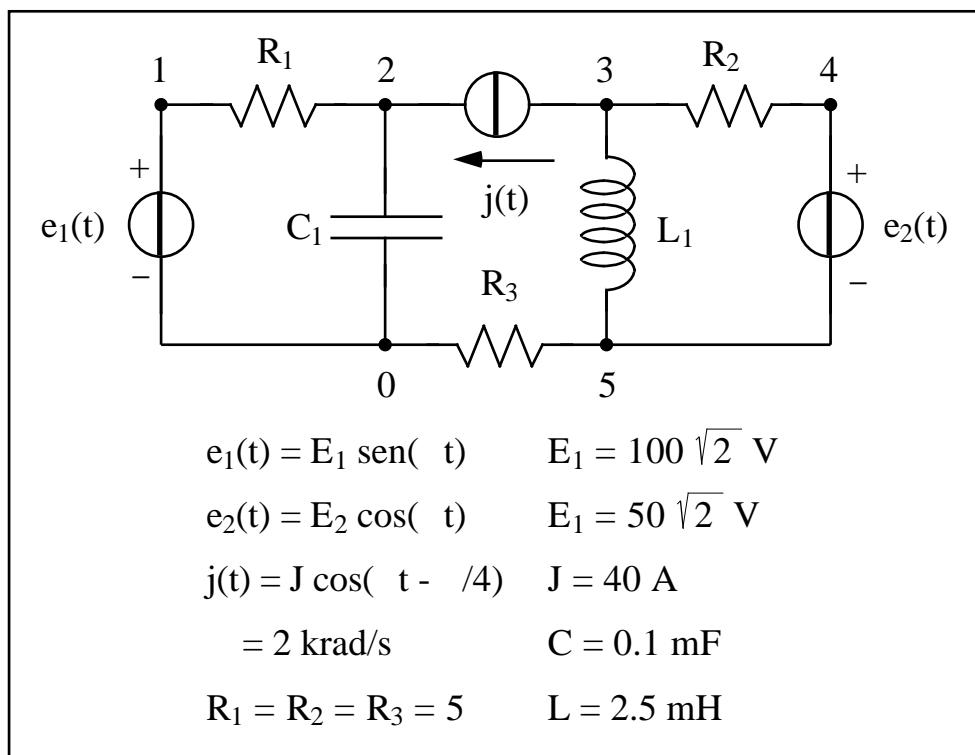


Figura 3.1: primo esempio di circuito.

La rete è costituita da tre generatori sinusoidali, due di tensione ed uno di corrente, e da alcuni bipoli, più precisamente tre resistori, un condensatore ed un induttore.

Il problema può essere risolto per mezzo delle istruzioni che seguono.

Esempio 3.1

* Primo esempio di circuito in alternata

```
R1    1    2    5
R2    4    3    5
R3    5    0    5
C1    2    0    1e-4
L1    3    5    2.5m
VE1   1    0    AC    141.42    -90
VE2   4    5    AC    70.71     0
IJ    3    2    AC    40     -45
.AC   LIN    1    318.35    318.35
.PRINT AC    VM(2)    VP(2)    VR(2)    VI(2)
.PRINT AC    VM(3,5)  VP(3,5)  VR(3,5)  VI(3,5)
.END
```

Esaminiamo in dettaglio questo file Spice. Dopo le linee relative alla definizione degli elementi passivi, introduciamo i generatori indipendenti. Ad esempio, l'ottava linea stabilisce che tra i nodi 1 e 0 vi è un generatore sinusoidale di tensione di ampiezza 141.42 V e fase -90°. Notiamo esplicitamente che Spice adotta come riferimento le cosinusidi, per cui una sinusoide è rappresentata da una cosinusoide con fase pari a -90°. A titolo di esempio riportiamo dei possibili generatori sinusoidali, che supporremo connessi tra i nodi 1 e 0, e la loro codifica Spice

$$v_1(t) = 10 \sin(t)$$

V1 1 0 AC 10 -90 ,

$$i_0(t) = 100 \sqrt{2} \sin(t - \pi/4)$$

I0 1 0 AC 141.42 -135 ,

$$v_{x1}(t) = 7 \sin(t + \pi/6)$$

VX1 1 0 AC 7 -60 .

L'undicesima linea del pacchetto di istruzioni

.AC LIN 1 318.35 318.35

richiede a Spice di eseguire una analisi in regime sinusoidale del circuito per un sol valore della frequenza (il terzo termine nell'istruzione in questione) pari a 318.35 Hz (gli ultimi due termini rappresentano il valore iniziale e finale del campo di frequenze da analizzare, e sono eguali). Ciò significa che i tre generatori sinusoidali oscilleranno sulla frequenza 318.35 Hz.

Le frequenze dovranno variare linearmente dal valore iniziale a quello finale, come specificato dal secondo termine (LIN) dell'istruzione. Naturalmente avendo richiesto nel nostro caso una sola frequenza - successivamente vedremo il caso più generale - questa informazione è pleonastica.

La *dodicesima e tredicesima riga* indicano che si richiede in uscita la stampa dei valori in modulo (VM) e fase (VP) ed in forma cartesiana (VR + j VI), dei potenziali dei nodi 2 (prima istruzione ‘.PRINT’) e della differenza di potenziale fra 3 e 5 (seconda istruzione).

La risposta di Spice contiene, nel file di uscita, quanto di seguito riportato.

FREQ	VM(2)	VP(2)	VR(2)	VI(2)
	3.183E+02	2.236E+02	-1.084E+02	-7.071E+01
FREQ	VM(3,5)	VP(3,5)	VR(3,5)	VI(3,5)
	3.183E+02	1.118E+02	1.616E+02	-1.061E+02
				3.536E+01

Notiamo esplicitamente che anche nell'analisi in regime sinusoidale è possibile richiedere in uscita la corrente che attraversa un qualsiasi elemento della rete, compresa quella dei generatori indipendenti. Riportiamo di seguito la tabella che illustra i codici utilizzabili nell'istruzione ‘.PRINT’ e il tipo di informazione che si ottiene in uscita nel file ‘.OUT’.

Tipo di codice	Informazione in uscita
nessun codice	ampiezza
M	ampiezza
DB	ampiezza in dB
P	fase in gradi
R	parte reale
I	parte immaginaria

Per utilizzare a pieno le potenzialità dell'istruzione di controllo ‘.AC’, sostituiamo le ultime quattro linee di programma:

```
.AC LIN 31 50 1k
.PLOT AC I(C1)
.PROBE
.END
```

La prima di tali istruzioni chiede a Spice di sviluppare l'analisi per frequenze variabili linearmente, in 31 passi, a partire da 50 Hz ed a finire a 1 kHz. L'istruzione successiva richiede di riportare il grafico della corrente che attraversa il condensatore C1 in funzione della frequenza. Sotto queste condizioni nel file di uscita si ritrovano i dati come di seguito riportato.

FREQ	I(C1)					
(*)-----	1.0000E+00	1.0000E+01	1.0000E+02	1.0000E+03	1.0000E+04	
5.000E+01	9.814E+00	*	.	.	.	
8.167E+01	1.572E+01	.	*	.	.	
1.133E+02	2.121E+01	.	*	.	.	
1.450E+02	2.622E+01	.	*	.	.	
1.767E+02	3.069E+01	.	*	.	.	
2.083E+02	3.464E+01	.	*	.	.	
2.400E+02	3.808E+01	.	*	.	.	
2.717E+02	4.106E+01	.	*	.	.	
3.033E+02	4.363E+01	.	*	.	.	
3.350E+02	4.585E+01	.	*	.	.	
3.667E+02	4.776E+01	.	*	.	.	
3.983E+02	4.941E+01	.	*	.	.	
4.300E+02	5.083E+01	.	*	.	.	
4.617E+02	5.207E+01	.	*	.	.	
4.933E+02	5.314E+01	.	*	.	.	
5.250E+02	5.408E+01	.	*	.	.	
5.567E+02	5.490E+01	.	*	.	.	
5.883E+02	5.563E+01	.	*	.	.	
6.200E+02	5.626E+01	.	*	.	.	
6.517E+02	5.683E+01	.	*	.	.	
6.833E+02	5.733E+01	.	*	.	.	
7.150E+02	5.778E+01	.	*	.	.	
7.467E+02	5.818E+01	.	*	.	.	
7.783E+02	5.854E+01	.	*	.	.	
8.100E+02	5.886E+01	.	*	.	.	
8.417E+02	5.916E+01	.	*	.	.	
8.733E+02	5.942E+01	.	*	.	.	
9.050E+02	5.966E+01	.	*	.	.	
9.367E+02	5.988E+01	.	*	.	.	
9.683E+02	6.008E+01	.	*	.	.	
1.000E+03	6.027E+01	.	*	.	.	

Ciò rappresenta un dignitoso tentativo di ottenere un grafico con un terminale non grafico, quindi disponibile a chiunque, o con una stampante a caratteri. Ma, naturalmente, si può ottenere di più.

Infatti, l'istruzione ‘.PROBE’, contenuta nel file di ingresso, ha richiesto a Spice di creare un file in uscita, individuato con l'estensione ‘.DAT’, che contiene tutti i dati delle elaborazioni effettuate. Un tale file può essere letto dal post-processore grafico ‘PROBE’ che è in grado di produrre grafici di alta qualità sulla base dei

dati forniti da Spice. La stessa curva precedente assume ora la forma mostrata in Figura 3.2.

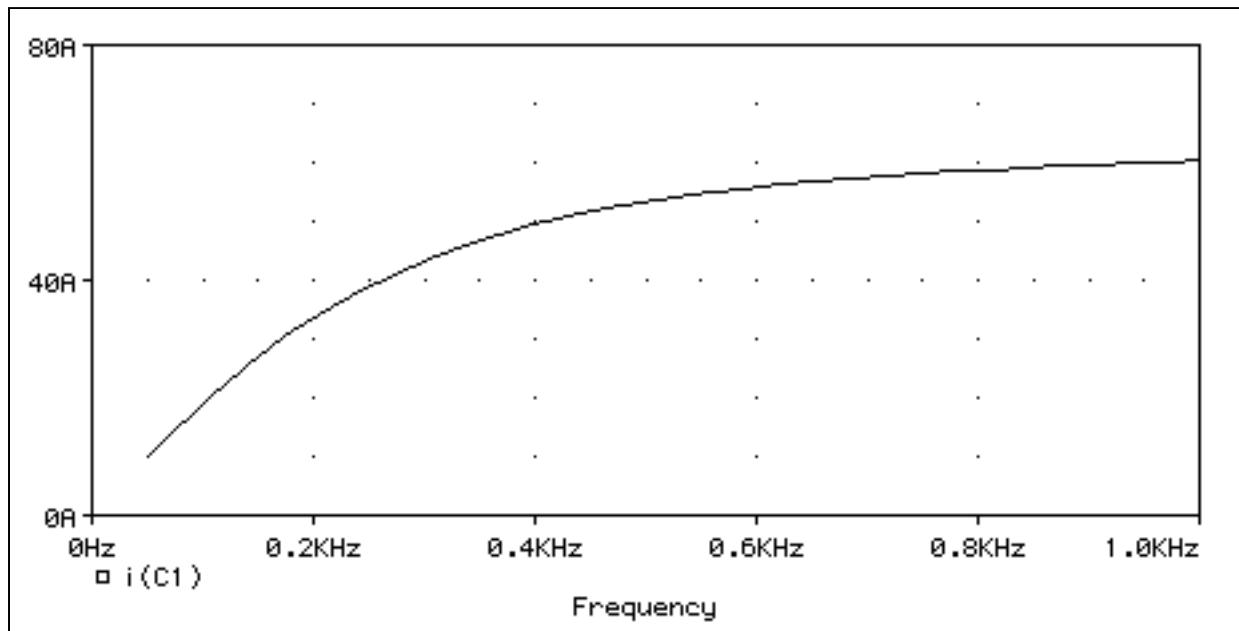


Figura 3.2: grafico della corrente del condensatore.

Le possibilità del post-processore grafico Probe sono notevoli ed il suo uso rende estremamente agevole la visualizzazione dei dati elaborati da Spice, essendo possibile effettuare operazioni algebriche e funzionali sulle grandezze da rappresentare. Daremo qualche altro cenno a Probe nell'ultimo capitolo, nella convinzione che soltanto l'uso, peraltro assai intuitivo ed immediato, vi aiuterà a scoprire tutte le sue possibilità.

3.3 Il condensatore e l'induttore

Il formato per individuare un condensatore di 1 mF, connesso tra i nodi 2 e 0 è

```
C1      2      0      1e-3
```

in cui C1 è il nome scelto. La prima lettera del nome deve essere sempre una C. In maniera analoga, un induttore di valore 7 mH connesso tra i nodi 1 e 2 è individuato dalla riga

```
L23lp    1      2      7m .
```

In tal caso il nome scelto è L23lp. Notate che per individuare un induttore la prima lettera è sempre una L. Ovviamente, anche per induttori e condensatori valgono le modalità per la scrittura dei multipli e sottomultipli del valore che abbiamo già incontrato per i resistori.

Nel capitolo riguardante l'analisi delle reti in evoluzione dinamica vedremo come inserire anche le condizioni iniziali per tali elementi 'a memoria'.

Infine, nell'utilizzo di condensatori e induttori in circuiti dinamici bisogna prestare attenzione a non incappare in due errori abbastanza comuni: creare un nodo del circuito che connette solo condensatori oppure disegnare una maglia della rete che contiene solo induttori e generatori di tensione, sia indipendenti che dipendenti. Il primo caso produrrebbe un errore durante l'analisi del punto di lavoro in quanto tale analisi ha bisogno di un percorso per la corrente continua che permetta di accedere a tutti i nodi del circuito e ciò sarebbe evidentemente impossibile per un nodo che 'vede' solo condensatori.

* Primo caso errato (un esempio)

VE	1	0	AC	1	0
R0	1	2		1	
C0	2	3		1	
C1	3	0		1	
.AC	LIN	1		1	

ERROR -- Node 3 is floating

Nel secondo caso una maglia di induttori e generatori di tensione produrrebbe un assurdo fisico nell'analisi del punto di lavoro, in quanto per correnti continue l'induttore si comporterebbe da cortocircuito in contrasto con la tensione imposta dal generatore.

* Secondo caso errato

VE	1	0	AC	1	0
L0	1	0		1	
.AC	LIN	1		1	

ERROR -- Voltage source and/or inductor loop involving VE
You may break the loop by adding a series resistance

Il primo problema si può risolvere agevolmente mediante un resistore di valore elevatissimo (rispetto agli altri resistori del circuito) che connetta il nodo isolato a un'altra parte del circuito, mentre il secondo problema si può risolvere mediante

un resistore di valore molto piccolo da porre in serie tra l'induttore e il generatore di tensione.

3.4 Sintassi per l'analisi in frequenza

Riassumiamo ora le istruzioni relative all'analisi in regime sinusoidale. Nelle linee riguardanti generatori indipendenti di tensione e di corrente:

```
VE    1    0    DC    1    AC    100    90
```

il campo relativo all'analisi DC diventa opzionale se seguito dalla voce AC cui segue il campo con il valore dell'ampiezza massima e il campo con la fase del generatore espressa in gradi. Attenzione al fatto che in questo caso la voce AC è obbligatoria e la sua assenza provoca un errore. Ricordiamo ancora una volta che Spice usa come riferimento le funzioni cosinusoidali per cui una sinusoide è rappresentata per mezzo della relazione:

$$\text{sen} = \cos(-\pi/2).$$

L'istruzione ‘.AC’, che chiede a Spice di effettuare l'analisi al variare della frequenza, può essere di tre tipi:

```
.AC    LIN     NP    FSTART    FSTOP ,
.AC    DEC     ND    FSTART    FSTOP ,
.AC    OCT     NO    FSTART    FSTOP .
```

Il secondo campo indica il tipo di scansione in frequenza, il terzo campo indica il numero di punti e infine FSTART e FSTOP indicano, rispettivamente, la frequenza iniziale e finale. Più precisamente:

- ‘LIN’ significa che la scansione in frequenza tra ‘FSTART’ e ‘FSTOP’ è lineare con passo

$$\text{FSTEP} = \frac{\text{FSTOP} - \text{FSTART}}{\text{NP} - 1};$$

- ‘DEC’ significa che la scansione in frequenza tra FSTART e FSTOP è a decadi (una decade è un intervallo la cui frequenza finale è dieci volte quella iniziale).

Ogni decade contiene ND punti e la generica frequenza contenuta nella decade che parte dalla frequenza FSD è

$$F_k = FSD \cdot 10^{k/ND}, \quad k = 1, \dots, ND;$$

- ‘OCT’ significa che la scansione in frequenza tra ‘FSTART’ e ‘FSTOP’ è a ottave (un’ottava è un intervallo la cui frequenza finale è il doppio di quella iniziale). Ogni ottava contiene NO punti e se, ad esempio, l’ottava comincia con la frequenza FSO, i punti saranno del tipo

$$F_k = FSO \cdot 2^{k/NO}, \quad k = 1, \dots, NO.$$

L’utilità principale dell’analisi in frequenza a decadi e a ottave è di permettere lo stesso dettaglio per tutta la banda di frequenze esaminata. Infine, come già mostrato negli esempi precedenti, i risultati dell’analisi in frequenza possono essere stampati e mostrati mediante un grafico nel file di uscita; oppure possono essere visualizzati mediante il post-processore grafico Probe.

3.5 Un classico esempio di analisi in frequenza

In questo paragrafo svilupperemo un classico esempio per il quale è semplice introdurre i concetti di banda e risonanza: l’analisi in frequenza della corrente che attraversa un circuito RLC serie. Considerazioni analoghe potrebbero essere sviluppate anche nel caso RLC parallelo.

Consideriamo, allora, il circuito di Figura 3.3 in cui abbiamo rappresentato un circuito RLC serie, alimentato da un generatore di tensione alternata di assegnata frequenza.

È ben noto che la frequenza di risonanza di questo circuito è pari a

$$f_0 = \frac{1}{2\sqrt{LC}} = 159.15 \text{ Hz},$$

e, pertanto, il diagramma del modulo della corrente al variare della frequenza dovrà avere un massimo in corrispondenza di questo valore.

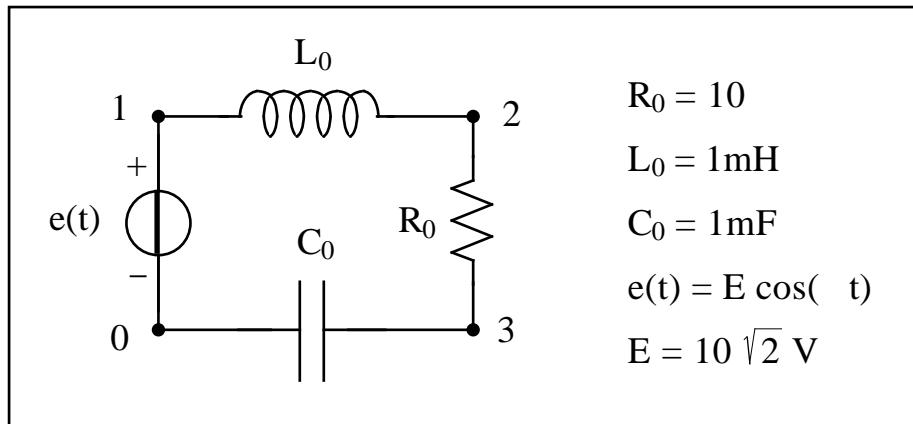


Figura 3.3: circuito RLC serie.

Il listato Spice che segue rappresenta la codifica del circuito di Figura 3.3

```

Esempio 3.2
*Circuito RLC serie
R0    2    3    10
L0    1    2    1m
C0    3    0    1m
VE    1    0    AC    14.142    0
.AC    LIN    10000    10    10k
.PROBE
.END

```

Osservate che, per l'analisi in frequenza, è importante anche prevedere quale sarà l'intervallo di frequenze di interesse, in maniera da non 'sprecare' tempo di calcolo. Nel nostro caso, abbiamo richiesto un'analisi in frequenza di tipo lineare con 10000 punti.

Il grafico del valore efficace della corrente del circuito si presenta come mostrato in Figura 3.4, confermando quanto probabilmente già sapevate dalla teoria, ovvero che la corrente ha un 'carattere' di tipo passa-banda.

Provate a completare il lavoro iniziato: fate un diagramma della fase della corrente e di qualche altra grandezza rilevante del circuito, come la tensione ai capi del condensatore, e scopritene gli andamenti in frequenza. Infine, se adoperate (usando con giudizio il numero di punti) le analisi DEC e OCT, troverete ... sempre lo stesso grafico del valore efficace della corrente al variare della frequenza.

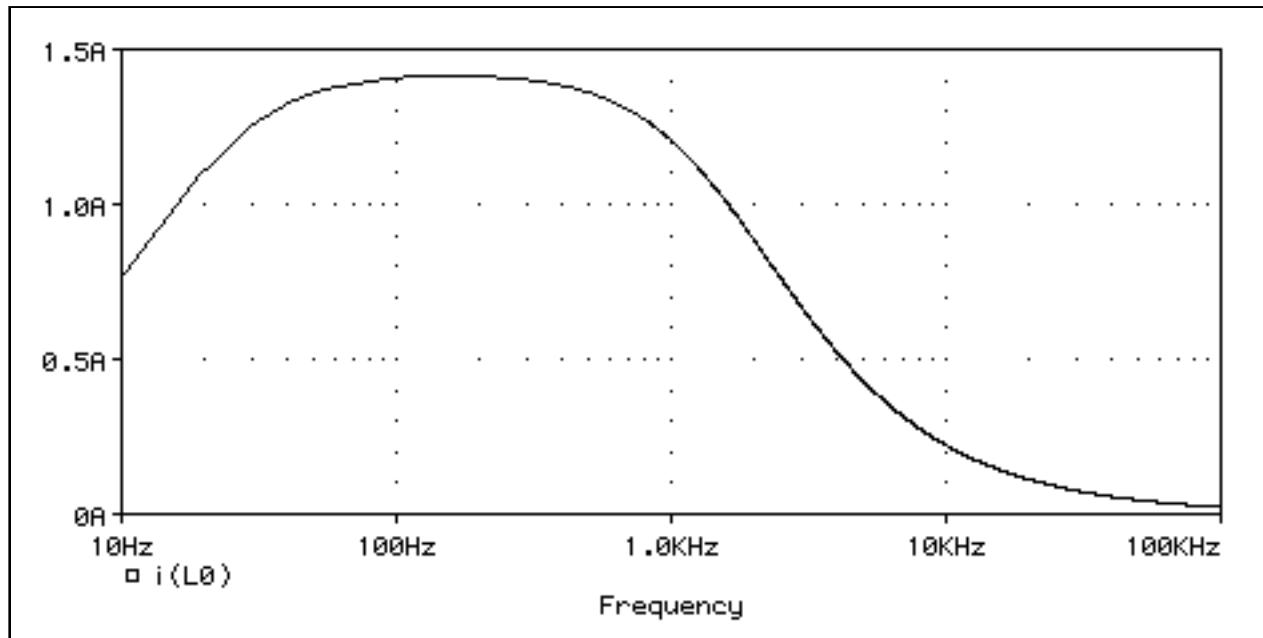


Figura 3.4: modulo della corrente in un circuito RLC serie.

3.6 Esempi

Per continuare ad approfondire l'analisi in regime sinusoidale, esaminate gli esempi che vi proponiamo.

- Supponendo che la rete mostrata in Figura 3.5 operi in regime sinusoidale, determinare la corrente $i(t)$.

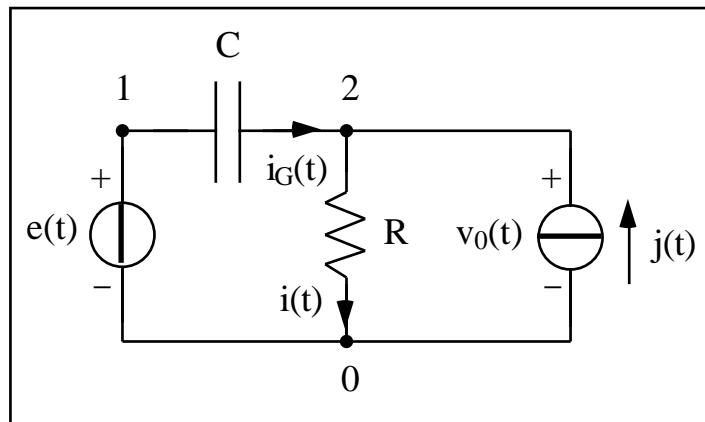


Figura 3.5: esempio di circuito in alternata.

Assumiamo che $e(t) = E \sqrt{2} \sin(\omega t)$, $j(t) = -I \sqrt{2} \cos(\omega t)$, $E = 10 \text{ V}$, $I = 2 \text{ mA}$, $\omega = 200 \text{ rad/s}$, $R = 5 \text{ k}\Omega$, $C = 2 \mu\text{F}$.

Risposta: risulta

$$i(t) = I \sin\left(t + \arctan \frac{1}{2}\right), \text{ con } I = \frac{2}{5} \sqrt{10} \text{ mA.}$$

Esempio 3.3

* Rete operante in regime sinusoidale

```
C0    1    2    2u
R0    2    0    5k
VE    1    0    AC    14.142    -90
IJ    0    2    AC    2.8284m    180
.AC    LIN    1    31.83    31.83
.PRINT  AC    IM(R0)    IP(R0)    IR(R0)    II(R0)
.END
```

- Determinare le correnti che fluiscono nei tre rami della rete mostrata in Figura 3.6.

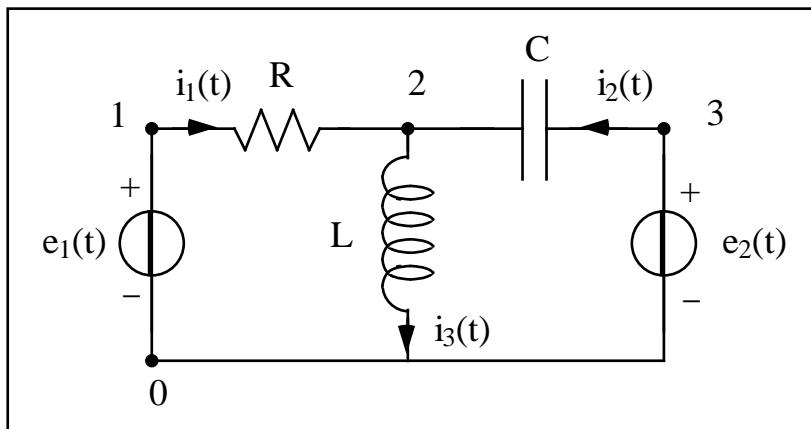


Figura 3.6: altro esempio in alternata.

Si assuma che $e_1(t) = E \sqrt{2} \sin(\omega t)$, $e_2(t) = -E \sqrt{2} \cos(\omega t)$, $E = 10 \text{ V}$, $\omega = 1 \text{ krad/s}$, $R = 5 \Omega$, $L = 5 \text{ mH}$, $C = 0.2 \text{ mF}$.

Risultati: posto

$$I = 2 \sqrt{2} \text{ A} \quad \omega = \arctan 2,$$

possiamo scrivere

$$\begin{cases} i_1(t) = -I \sin(1000t), \\ i_2(t) = I \sqrt{5} \sin(1000t - \phi), \\ i_3(t) = 2I \cos(1000t). \end{cases}$$

Esempio 3.4

* Due generatori in regime sinusoidale

```
R1    1    3    5
C1    2    3    0.2m
L1    3    0    5m
VE1   1    0    AC    14.14    -90
VE2   2    0    AC    14.14    180
.AC   LIN   1    159.15    159.15
.PRINT AC    IM(R1)    IP(R1)
.PRINT AC    IM(C1)    IP(C1)
.PRINT AC    IM(L1)    IP(L1)
.END
```

- La rete in Figura 3.7 opera in regime sinusoidale. Si determini la corrente $i(t)$ che fluisce nel condensatore.

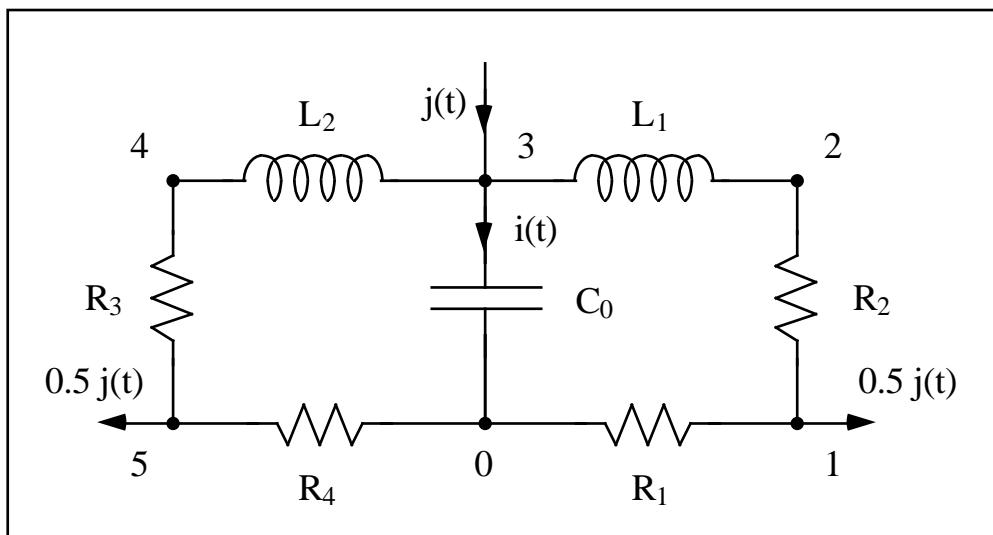


Figura 3.7: ultimo esempio di circuito operante in regime sinusoidale.

Si assuma che $j(t) = I \sqrt{2} \cos(\omega t)$, $\omega = 200 \text{ rad/s}$, $I = 2 \text{ A}$, $L_1 = L_2 = L = 5 \text{ mH}$, $C_0 = 10 \text{ mF}$, $R_1 = R_2 = R_3 = R_4 = R = 1 \Omega$.

Risultato: risulta

$$i(t) = I \cos\left(200t + \frac{\pi}{4}\right),$$

con $I = 2$ A.

Esempio 3.5

*Ancora un circuito in regime sinusoidale

```
R1    0    1    1
R2    1    2    1
L1    2    3    5m
C0    3    0    10m
L2    3    4    5m
R3    4    5    1
R4    5    0    1
I1    5    3    AC    1.414    0
I2    1    3    AC    1.414    0
.AC    LIN        1      31.83    31.83
.PRINT AC        IM(C0)    IP(C0)
.END
```

Capitolo 4

Reti in evoluzione dinamica

4.1 Introduzione

4.2 Analisi in transitorio

4.3 Sintassi per l'analisi in transitorio

4.4 Generatori di tensione e di corrente di forma qualsiasi

4.5 Generatori di tensione e di corrente di forma canonica

Sommario

In questo capitolo esamineremo le istruzioni Spice che consentono di realizzare l'analisi dinamica di una rete elettrica lineare. Particolare attenzione verrà dedicata alle più comuni forme d'onda dei generatori.

4.1 Introduzione

Uno degli aspetti più vantaggiosi dell'uso di Spice nella risoluzione di problemi su reti elettriche e elettroniche è proprio la possibilità di risolvere agevolmente problemi di evoluzione dinamica nel dominio del tempo. Questa possibilità, come vedremo nel seguito, si estende anche allo studio di reti che presentano elementi non lineari e questo, come si può intuire, facilita molto il lavoro del progettista che, altrimenti, solo in alcuni semplici casi potrebbe arrivare ad una soluzione esprimibile in forma analitica o in forma semplificata. In questo capitolo introdurremo le istruzioni Spice che permettono l'analisi in transitorio delle reti lineari per mezzo di alcuni esempi che, speriamo, servano a chiarire le istruzioni generali introdotte.

Queste istruzioni includono innanzitutto la riga '.TRAN' che permette l'analisi in transitorio e poi numerose possibilità per definire generatori indipendenti di tensione e di corrente di forme canoniche (seno smorzato, onda quadra e così via) e il generatore 'Pice Wise Linear' (costante a tratti) che permette di costruire un segnale di forma qualsiasi. L'uso accorto di queste possibilità consente sia la simulazione di semplici reti, magari comprendenti interruttori ideali, sia la simulazione di dinamiche complesse forzate da generatori di forma assegnata.

4.2 Analisi in transitorio

Cominciamo con una semplice rete in evoluzione dinamica, analizzando il circuito mostrato in Figura 4.1.

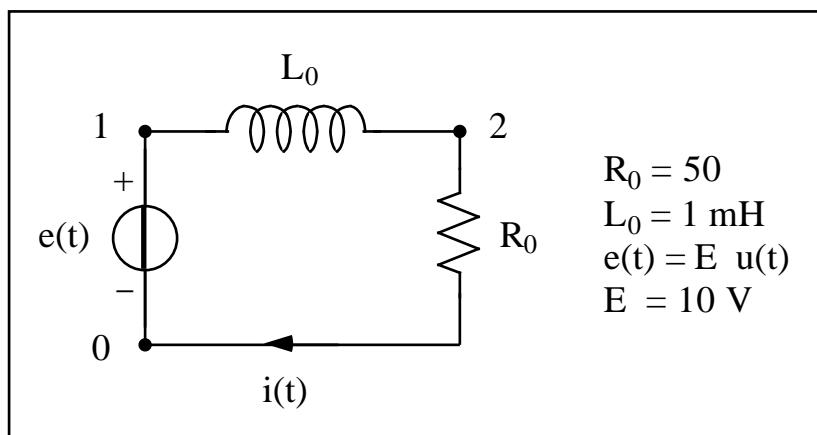


Figura 4.1: circuito per la carica di un induttore.

Esso descrive un fenomeno abbastanza semplice, la carica di un induttore per mezzo di un generatore di tensione (costante). Questa rete può essere codificata in Spice per mezzo del seguente insieme di istruzioni.

Esempio 4.1

* Carica di un circuito RL

```
R0    2    0    50
L0    1    2    1m    IC=0
VE    1    0    DC    10
.TRAN    5u    0.2m  UIC
.PROBE
.END
```

Notiamo subito che la codifica dell'induttore, alla quarta linea, segue quanto già detto nel Capitolo 3 solo che, stavolta, si assegna alla corrente che scorre nell'induttore un valore iniziale nullo all'istante iniziale ($IC = 0$, dove IC sta per 'Initial Condition').

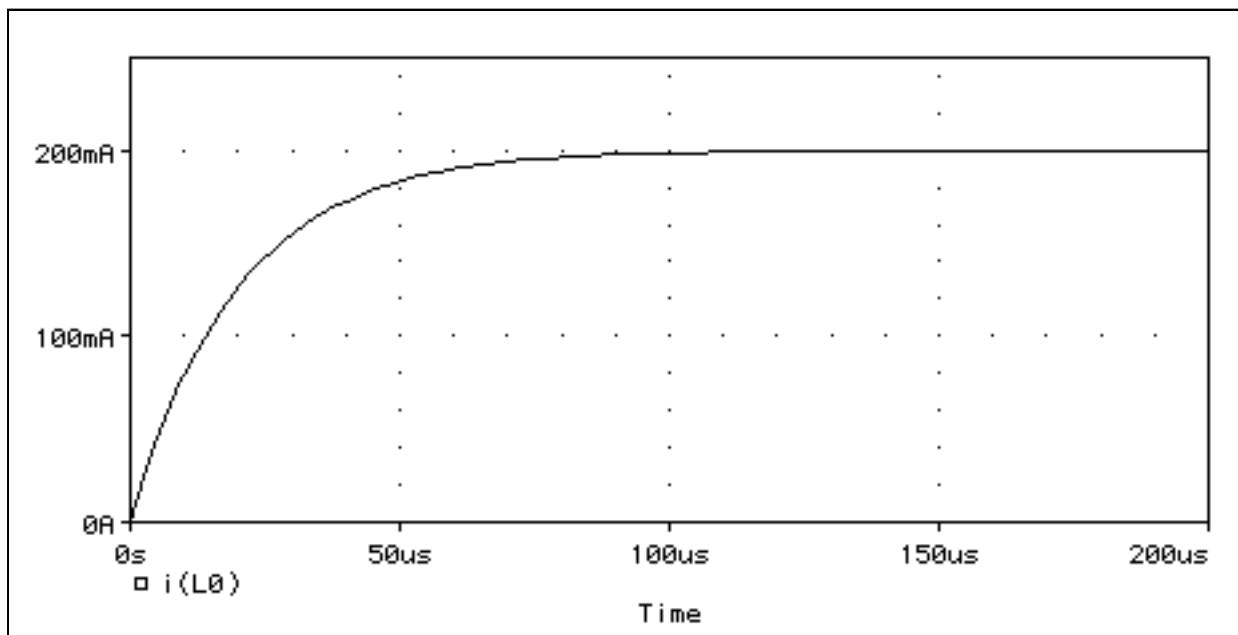


Figura 4.2: corrente di carica dell'induttore.

La quinta linea, invece, riguarda il tipo di analisi da effettuare e richiede di valutare il transitorio a partire dall'istante $t = 0$ fino all'istante $t = 0.2$ ms, utilizzando le condizioni iniziali (UIC sta per 'Use Initial Conditions') e con un passo di visualizzazione di $5 \mu\text{s}$.

La parola chiave 'UIC' indica a Spice che le condizioni iniziali specificano completamente il punto di lavoro iniziale della rete nell'analisi in transitorio.

La risposta, che d'ora in poi esamineremo prevalentemente utilizzando Probe, fornisce il grafico della corrente di Figura 4.2, che descrive il caratteristico andamento della corrente in un induttore durante un processo di carica.

Si noti che non è stato necessario introdurre esplicitamente un interruttore nei dati forniti a Spice, infatti l'analisi in transitorio effettua automaticamente 'la chiusura dell'interruttore' a partire dall'istante d'inizio.

Inoltre la presenza delle condizioni iniziali svolge un ruolo fondamentale: cambiando infatti tali condizioni l'andamento della corrente cambia di conseguenza.

Esempio 4.2

```
* Scarica di un induttore carico
R0    2    0    50
L0    1    2    1m    IC=0.4
VE    1    0    DC    10
.TRAN    5u    0.2m    UIC
.PROBE
.END
```

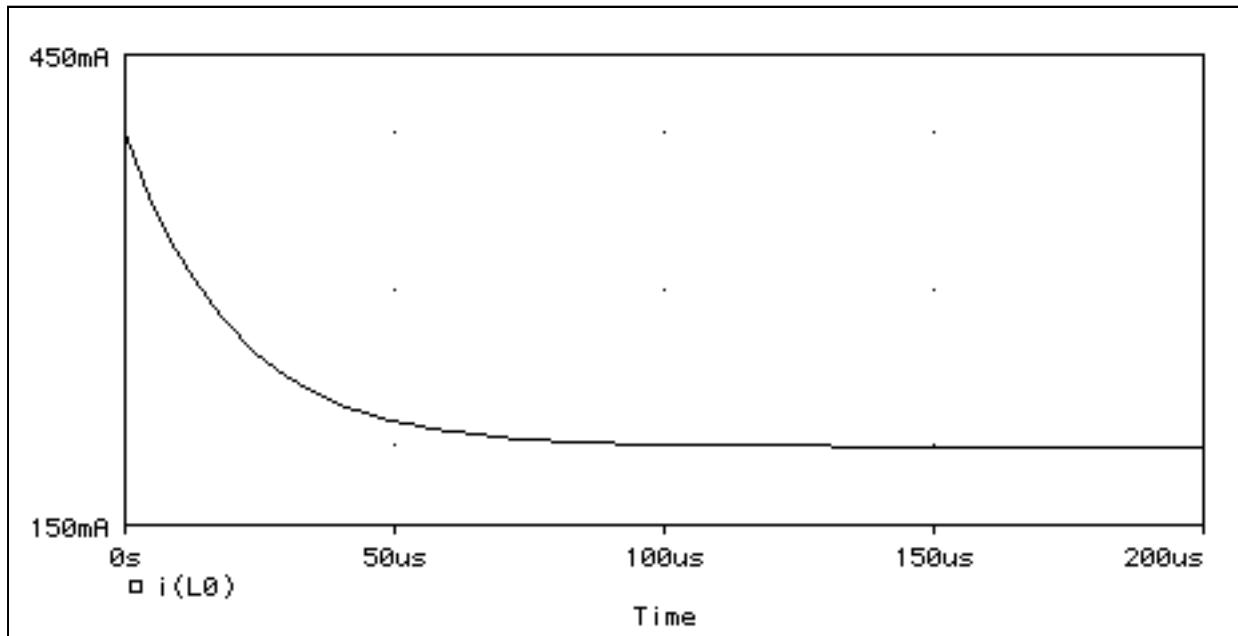


Figura 4.3: corrente di carica in presenza di condizioni iniziali non nulle.

Nell'esempio 4.2 abbiamo considerato il caso in cui all'istante iniziale scorra nell'induttore una corrente di 0.4 A. L'andamento si modifica come mostrato in Figura 4.3.

Infine, facciamo notare che, per i nostri scopi, è praticamente obbligatorio usare le condizioni iniziali dato che non abbiamo definito alcun generatore in transitorio e abbiamo sfruttato il valore 10 V del generatore VE. Tale modo di procedere produce un risultato significativo finché imponiamo una condizione iniziale (IC) che implichi la presenza di un transitorio. Provate per esercizio a rimuovere le condizioni iniziali e l'opzione UIC, ritroverete che la corrente nell'induttore non si muove mai dal valore di regime ($t = 0$) pari a 200 mA.

4.3 Sintassi per l'analisi in transitorio

In questo paragrafo commenteremo in maniera dettagliata la sintassi dell'istruzione '.TRAN' che serve a effettuare l'analisi in transitorio di una rete. Il formato più generale possibile per la linea '.TRAN' è del tipo

```
.TRAN      TSTEP      TSTOP      <TSTART>      <TMAX>      <UIC> .
```

dove i termini compresi tra '<>' sono opzionali. Tale istruzione specifica un'analisi transitoria effettuata dall'istante ' $t = \text{TSTART}$ ' fino all'istante ' $t = \text{TSTOP}$ ', con un passo di visualizzazione pari a TSTEP.

Se il campo ' $t = \text{TSTART}$ ' è specificato, allora la presentazione dei risultati dell'analisi transitoria nel file di uscita o la sua visualizzazione mediante Probe parte proprio dall'istante TSTART; mentre il calcolo parte, in ogni caso, dall'istante ' $t = 0$ '. TMAX, invece, specifica il massimo passo che Spice deve adottare per l'elaborazione. Se TMAX non è presente, Spice usa TSTEP per il calcolo (e ovviamente per la visualizzazione dei risultati) oppure

TSTOP - TSTART,
50

se quest'ultima grandezza fosse più piccola di TSTEP; questo vuol dire che tra TSTART e TSTOP sono visualizzati almeno 50 punti. TMAX è utile qualora si volesse garantire un passo di calcolo più piccolo del passo di visualizzazione richiesto e questa opzione è molto utile, come vedremo in seguito, nel calcolo di transitori molto delicati come quelli che coinvolgono linee di trasmissione o più semplicemente segnali con fronti di salita e discesa molto ripidi.

La visualizzazione dei risultati dell'analisi transitoria segue gli stessi criteri visti per gli altri casi. Considerando l'esempio 4.1, è possibile stampare nel file di uscita gli istanti temporali e la corrente dell'induttore mediante l'istruzione

```
.PRINT TRAN I(L1) .
```

Analogamente è sempre possibile ottenere un grafico nel file di uscita mediante

```
.PLOT TRAN I(L1) .
```

Notate che il campo TRAN specifica che i valori da stampare sono solo quelli del transitorio. Ovviamente una eventuale riga ‘.PROBE’ fa generare un file di dati leggibile dal post-processore grafico Probe.

Esiste un’altra maniera per introdurre le condizioni iniziali nel circuito secondo l’istruzione:

```
.IC V(NODO)=VALORE1 .
```

L’istruzione ‘.IC’ consente quindi di introdurre direttamente i potenziali dei nodi all’istante iniziale. Ovviamente, per rendere efficace tale istruzione, bisognerà sempre usare l’opzione ‘UIC’ nella riga ‘.TRAN’. Bisogna precisare, però, che le funzionalità dell’opzione UIC e delle condizioni iniziali possono dipendere dalla versione di Spice, per cui preferibile specificare sempre le condizioni iniziali degli elementi a memoria anche se queste sono nulle ($IC=0$) in modo da evitare di incappare in errori dovuti a valori di ‘default’ non controllati.

Infine, sottolineiamo che nell’analisi transitoria è fondamentale scegliere il passo di visualizzazione e l’intervallo di tempo corretti per studiare il fenomeno sotto esame. Il primo dovrebbe essere almeno un decimo della più piccola costante di tempo del circuito in modo da poter discernere con sufficiente accuratezza il comportamento della rete, laddove il secondo dovrebbe essere almeno dieci volte la più grande costante di tempo del circuito, in maniera da essere sicuri che il transitorio si sia completamente esaurito.

4.4 Generatori di tensione e di corrente di forma qualsiasi

La presenza di interruttori in una rete elettrica può essere simulata anche in maniera diversa da quanto visto nel primo esempio fornito in questo capitolo. Infatti nella realizzazione dell’esempio di Figura 4.4, utilizziamo un generatore ideale di tensione con forma d’onda lineare a tratti.

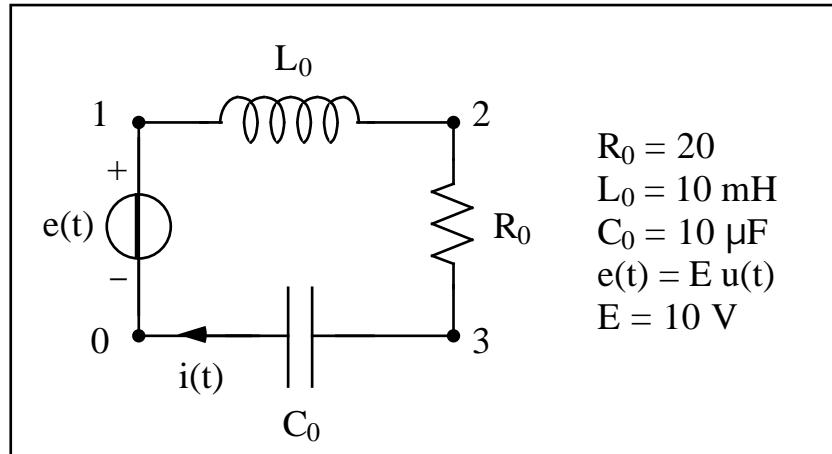


Figura 4.4: circuito con generatore di tensione a gradino.

Esempio 4.3

* Generatore di tensione a gradino

```

R0    2    3    20
L0    1    2    10m    IC=0
C0    3    0    1e-6   IC=0
VE    1    0    PWL(0 0 1u 10 2m 10)
.TRAN  1n    2m
.PROBE
.END

```

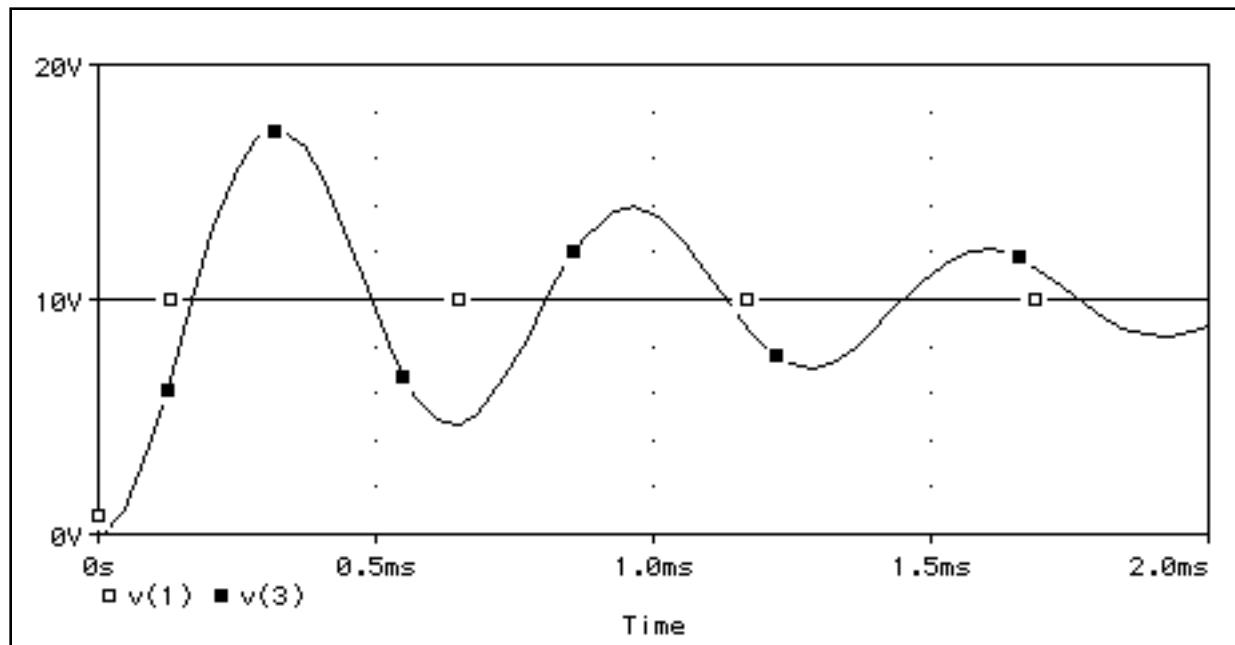


Figura 4.5: tensione ai capi del condensatore.

La sesta linea del pacchetto di istruzioni comunica a Spice che tra i nodi 1 e 0 è inserito un generatore che, in transitorio, fornisce ai suoi morsetti una tensione che al tempo $t = 0$ vale 0 V, al tempo $t = 1 \mu\text{s}$ vale 10 V, ed ancora al tempo $t = 2 \text{ ms}$ è pari a 10 V. Spice interpola linearmente tra questi punti.

In pratica con questa istruzione è possibile simulare un qualsiasi andamento della tensione approssimandolo con un andamento lineare a tratti ('Pice Wise Linear'). Nel caso in esame, essendo il fronte di salita della tensione estremamente breve rispetto ai tempi caratteristici del circuito, la forma d'onda simulata sarà quella di una gradino di tensione.

La risposta di Spice, analizzata con Probe, fornisce il grafico della tensione sul condensatore riportato in Figura 4.5, che possiamo ritenere una soddisfacente risposta al gradino, anche se la forma d'onda da noi assegnata in ingresso è, a rigore, un trapezio con tempo di salita molto piccolo.

Ma naturalmente è possibile simulare andamenti più complessi, come nel caso del seguente esempio, mostrato in Figura 4.6, che fa riferimento ad un generatore di tensione tipo rampa.

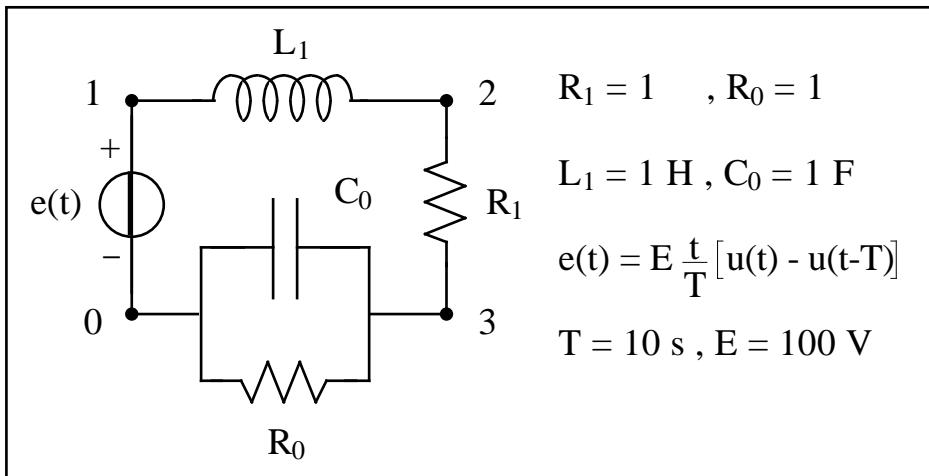


Figura 4.6: circuito con generatore a rampa.

Esempio 4.4

* Generatore a rampa

```
R0    3    0    1
C0    3    0    1
R1    2    3    1
L1    2    1    1
VE    1    0    PWL(0 0 10 100 10.0001 0 100 0)
.TRAN    1m    20
.PROBE
.END
```

La risposta di Spice fornisce il seguente andamento per la tensione ai capi del condensatore; nello stesso diagramma si è riportato anche l'andamento della tensione ai morsetti del generatore.

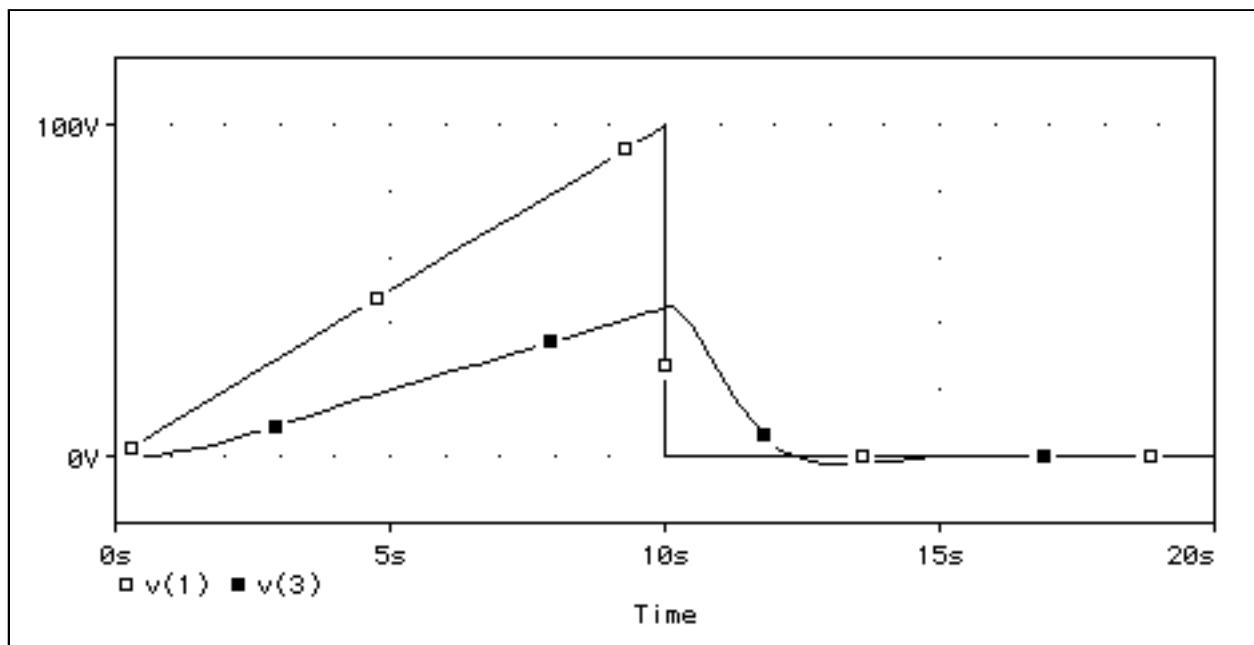


Figura 4.7: generatore a rampa e tensione del condensatore.

La forma generale dell'istruzione appena introdotta è

VPWL 1 0 PWL(T1 V1 T2 V2 ...) ,

che rappresenta un generatore di tensione lineare a tratti, posto tra i nodi 1 e 0, che al tempo T1 assume il valore V1, al tempo T2 assume il valore V2, e così via. Come già detto in precedenza, possiamo concludere che con il generatore PWL è possibile simulare praticamente qualsiasi forma d'onda, anche se nel caso di segnali periodici o di forma canonica, Spice ci mette a disposizione soluzioni più agevoli, come vedremo nel prossimo paragrafo.

4.5 Generatori di tensione e di corrente di forma canonica

Oltre ai generatori lineari a tratti, Spice offre la possibilità di introdurre generatori di tensione e di corrente di forma canonica: impulso periodico, sinusoida smorzata, impulso esponenziale e singola frequenza FM.

La sintassi per un generatore di tensione a *impulso periodico* di nome VIP connesso tra i nodi 1 e 0 è

```
VIP    1    0    PULSE(V1 V2 TD TR TF PW PER) ,
```

in cui i diversi parametri rappresentano

- V1 il valore iniziale,
- V2 il valore di picco,
- TD il tempo di ritardo,
- TR il tempo di salita,
- TF il tempo di discesa,
- PW la durata del picco,
- PER il periodo.

Per rendere ancora più chiaro il significato dei parametri appena introdotti, ecco il grafico di un impulso con

$V1 = 2 \text{ V}$, $V2 = 4 \text{ V}$, $TD = 5 \text{ ms}$, $TR = 2.5 \text{ ms}$, $TF = 2.5 \text{ ms}$,
 $PW = 5 \text{ ms}$, $PER = 10 \text{ ms}$.

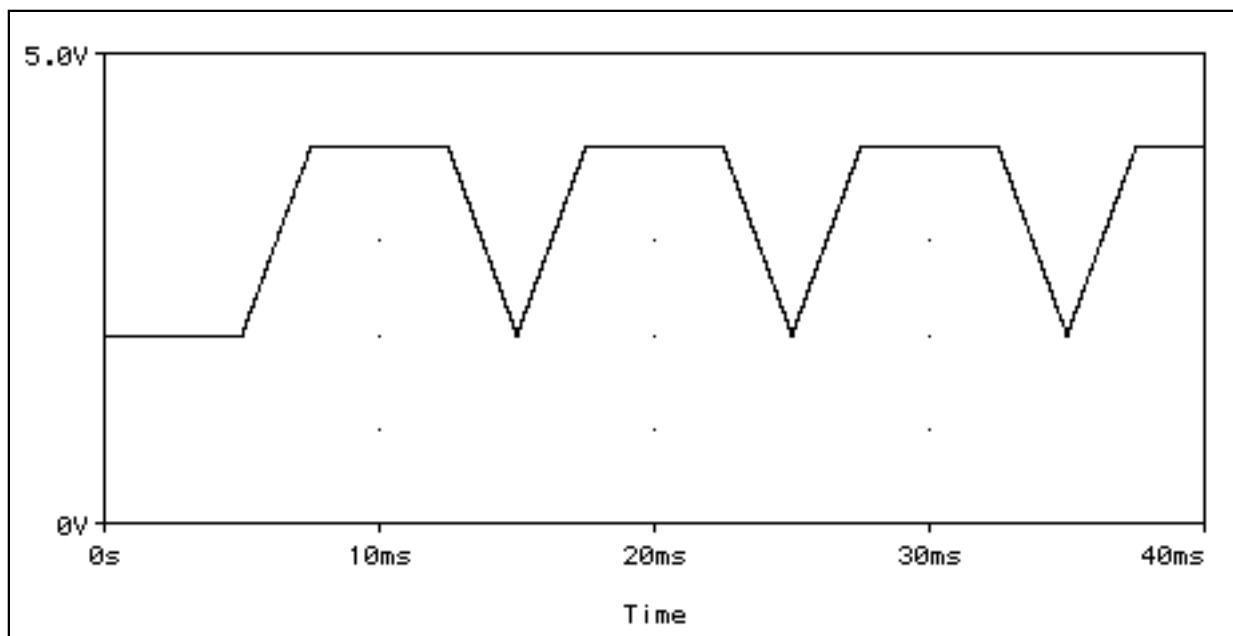


Figura 4.8: generatore di tensione di forma tipo impulso periodico.

L'uso opportuno dei parametri a disposizione consente la realizzazione di onde triangolari e quadre con differenti rapporti d'aspetto (modulazione degli

intervalli pieno-vuoto), fino a giungere alla realizzazione di un treno di impulsi, dove la parola impulso non deve far pensare all'irrealizzabile entità matematica di Dirac, ma piuttosto a una sequenza di segnali la cui durata temporale è molto piccola e l'ampiezza è molto grande (rispetto alle grandezze caratteristiche del circuito).

L'espressione per un generatore di tensione tipo *sinusoide smorzata* è

$$VSS = \begin{cases} V1 , & 0 \leq t \leq TD , \\ V1 + V2 \exp[-(t - TD)] \sin[2 \pi FREQ(t - TD)] , & t > TD . \end{cases}$$

La sintassi per un tal generatore connesso tra i nodi 1 e 0 è

VSS 1 0 SIN(V1 V2 FREQ TD) ,

dove stavolta i parametri rappresentano:

- V1 il valore iniziale,
- V2 il valore di picco,
- FREQ la frequenza,
- TD il tempo di ritardo,
- il fattore di attenuazione.

La Figura 4.9 mostra il caso particolare

$$V1 = 2 \text{ V} , \quad V2 = 2 \text{ V} , \quad FREQ = 500 \text{ Hz} , \quad TD = 2.5 \text{ ms} , \quad = 500 \text{ s}^{-1} ,$$

che rende più evidente il significato dei diversi parametri.

Osserviamo, infine, che ponendo $= 0$, possiamo ottenere una sinusoide non smorzata.

L'espressione per un generatore di tensione tipo *impulso esponenziale* è

$$VIS = \begin{cases} V1 , & 0 \leq t \leq TD1 , \\ V1 + (V2 - V1) \left[1 - \exp\left(-\frac{t - TD1}{1}\right) \right] , & TD1 \leq t \leq TD2 , \\ V1 + (V2 - V1) \left[\exp\left(-\frac{t - TD2}{2}\right) - \exp\left(-\frac{t - TD1}{1}\right) \right] , & t > TD2 . \end{cases}$$

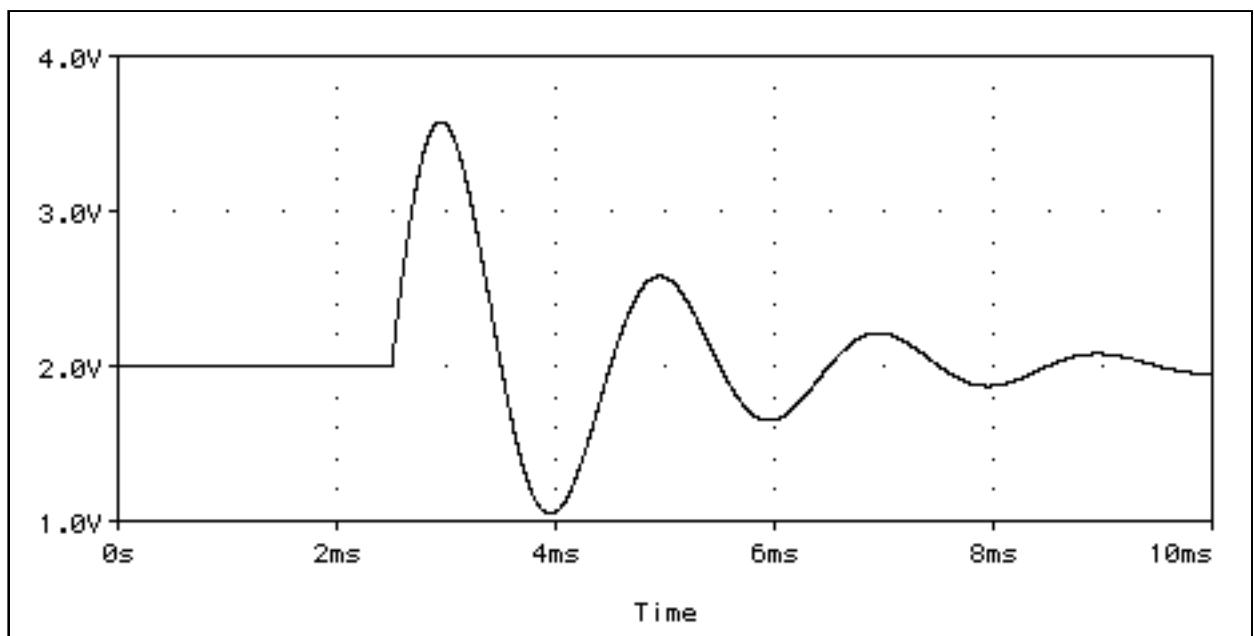


Figura 4.9: generatore di tensione di forma tipo sinuosoide smorzata.

La sintassi per un tal generatore connesso tra i nodi 1 e 0 è

VIS 1 0 EXP(V1 V2 TD1 1 TD2 2),

dove, ovviamente, i parametri rappresentano

- V1 il valore iniziale,
- V2 il valore asintotico,
- TD1 il tempo di ritardo per il primo impulso,
- 1 la costante di tempo del primo impulso,
- TD2 il tempo di ritardo per il secondo impulso,
- 2 la costante di tempo del secondo impulso.

In Figura 4.10 mostriamo un impulso esponenziale con

$V1 = 2 \text{ V}$, $V2 = 40 \text{ V}$, $TD1 = 2.5 \text{ ms}$, $1 = 1 \text{ ms}$, $TD2 = 6 \text{ ms}$,
 $2 = 2 \text{ ms}$.

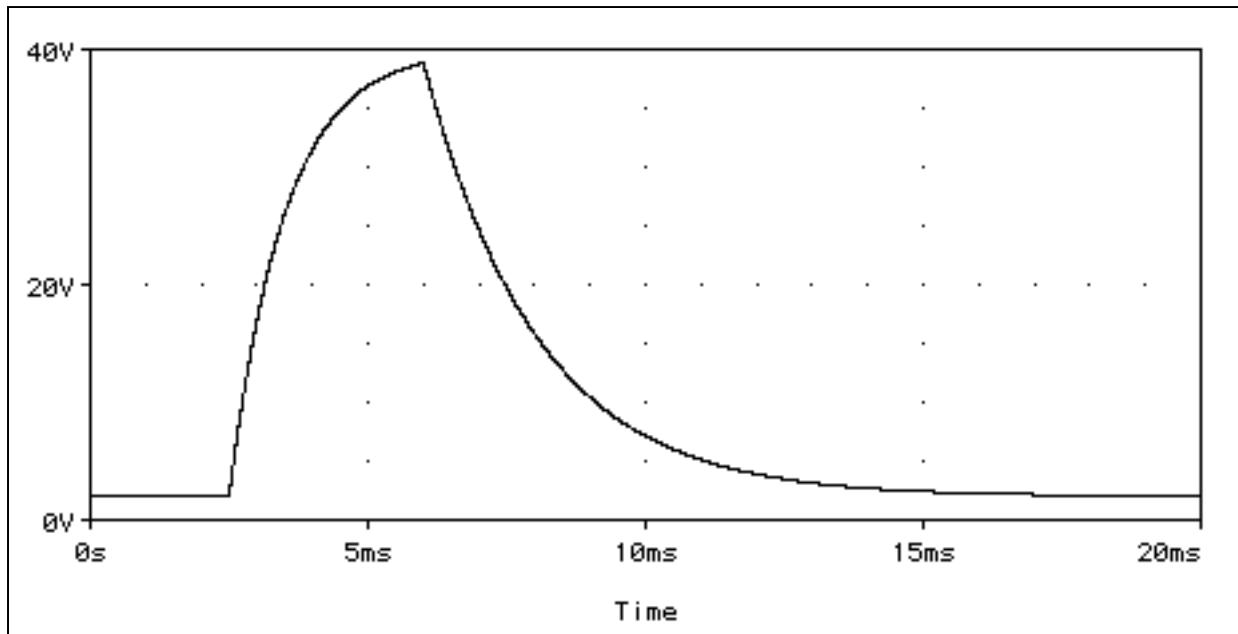


Figura 4.10: generatore di tensione di forma tipo impulso esponenziale.

L'ultimo tipo di generatore di segnale di forma canonica disponibile in Spice è un generatore che presenta una sinusoide modulata in fase da un'altra sinusoide. Tutto questo serve ovviamente a simulare un singolo tono di un segnale FM (Frequency Modulation). L'espressione funzionale per un generatore di tensione di questo tipo è la seguente

$$VFM = V1 + V2 \sin[2 \pi FREQP t + MOD \sin(2 \pi FREQM t)] .$$

La sintassi per un tal generatore connesso tra i nodi 1 e 0 è

VFM 1 0 SFFM(V1 V2 FREQP MOD FREQM)

la dicitura SFFM sta per ‘Single Frequency FM’ e i parametri dell’istruzione Spice rappresentano

- V1 il valore medio,
- V2 il valore di picco,
- FREQP la frequenza della portante,
- MOD l’indice di modulazione,
- FREQM la frequenza della modulante.

Un’interpretazione grafica dei parametri rende ancora più chiaro il significato introdotto nella tabella precedente. La Figura 4.11 mostra un esempio con

$V1 = 2 \text{ V}$, $V2 = 2 \text{ V}$, $\text{FREQP} = 500 \text{ Hz}$, $\text{MOD} = 5$, $\text{FREQM} = 50 \text{ Hz}$.

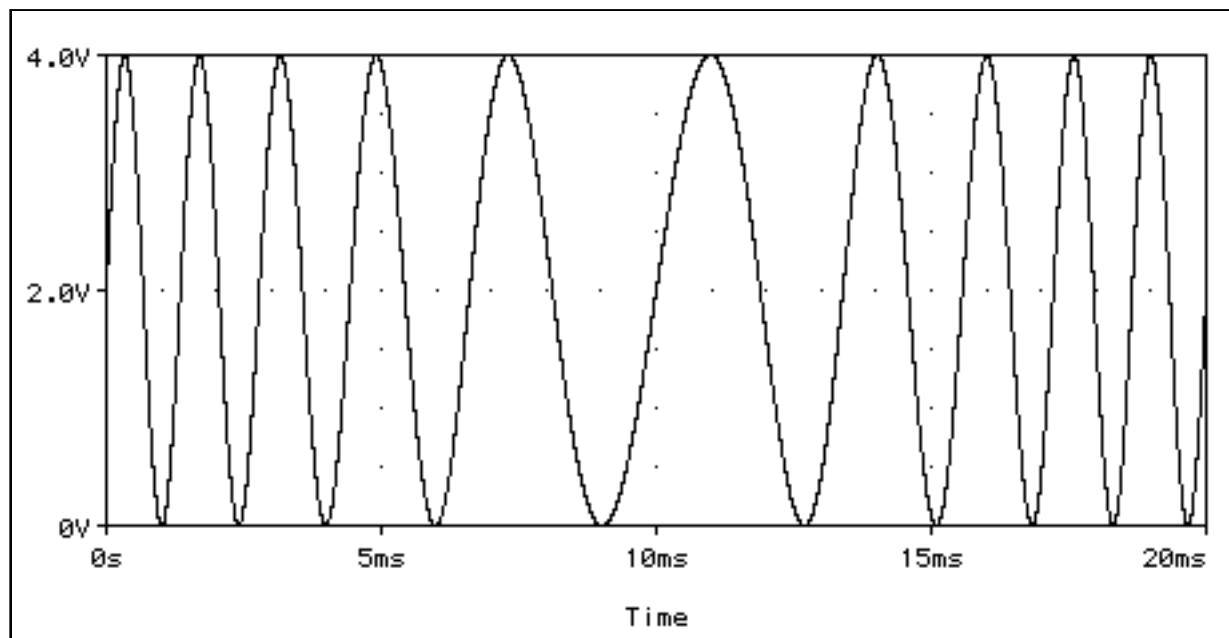


Figura 4.11: generatore di tensione di forma tipo SFFM.

Capitolo 5

Modelli di dispositivi e circuiti

5.1 Introduzione

5.2 Generatori dipendenti lineari

5.2.1 Sintassi dei generatori dipendenti controllati in tensione

5.2.2 Sintassi dei generatori dipendenti controllati in corrente

5.2.3 Un esempio: il circuito equivalente del transistore

5.3 Generatori dipendenti non lineari

5.3.1 Generatori non lineari di tipo polinomiale

5.3.2 Generatori dipendenti controllati in tensione di vario tipo

5.4 Sottocircuiti

5.4.1 Sintassi dei sottocircuiti

5.4.2 Sintassi della chiamata ai sottocircuiti

5.4.3 Sottocircuiti con parametri

5.4.4 Semplice modulatore AM

5.5 L'istruzione ‘.MODEL’

5.5.1 Modello di resistore

5.5.2 Modello di condensatore

5.5.3 Modello di induttore

Sommario

In questo capitolo studieremo la sintassi dei generatori controllati, lineari e non lineari. Poi discuteremo le istruzioni che ci consentono di costruire i sottocircuiti. Infine ci soffermeremo sulla istruzione ‘.MODEL’.

5.1 Introduzione

In questo capitolo descriveremo quei modelli che la sintassi Spice mette a disposizione per rappresentare singoli dispositivi a due o più morsetti e parti di una rete che possono essere inglobate in una ‘scatola’ che poi viene vista dall’esterno solamente attraverso i nodi di interfacciamento. Per rispondere a quest’ultima esigenza Spice mette a disposizione l’istruzione ‘.SUBCKT’ che serve a definire un sottocircuito, utile per descrivere una rete, che realizza una funzione desiderata e di cui abbiamo bisogno spesso.

Il sottocircuito, attraverso i suoi nodi di interfacciamento, può essere utilizzato e inserito in più punti di un’altra rete e questo semplifica notevolmente la comprensione della rete stessa sia dal punto di vista logico e funzionale, sia dal punto di vista più pratico, ma non meno importante, della ricerca di eventuali errori.

Invece, per rispondere alla prima esigenza, Spice offre l’istruzione ‘.MODEL’ che permette di variare i parametri di alcuni tipi di dispositivi di cui esiste già un ‘sottocircuito’ all’interno del programma Spice. Attraverso l’istruzione ‘.MODEL’ noi possiamo cambiare i parametri costruttivi e dal tipo di dispositivo generico (diodo e transistore ad esempio) ottenere proprio quello con le caratteristiche di ingresso e uscita che ci interessano.

Infine, laddove non esista un modello interno a Spice e nemmeno sia possibile realizzare la funzione logica o la caratteristica che ci interessano, esistono i generatori dipendenti di tensione o corrente che permettono di definire la loro caratteristica di funzionamento, quasi in maniera arbitraria, attraverso numerose opzioni.

Cominceremmo il capitolo con la descrizione dei generatori dipendenti, seguiranno poi i sottocircuiti e infine introdurremo l’istruzione ‘.MODEL’, di cui forniremo come esempi la sintassi dei modelli per il resistore, il condensatore e l’induttore. La sintassi e le ulteriori possibilità dell’istruzione ‘.MODEL’ per i dispositivi a semiconduttore saranno poi spiegate in un prossimo capitolo.

5.2 Generatori dipendenti lineari

Spice offre la notevole possibilità di inserire nei circuiti dei bipoli generatori (lineari) di corrente e di tensione controllati. Cosa vuol dire? Che il valore del generatore è proporzionale alla differenza di potenziale di una coppia di nodi oppure è proporzionale alla corrente che scorre in un generatore indipendente di tensione.

Nel seguito del presente paragrafo illustriamo la sintassi di tali istruzioni e una classica applicazione: il circuito equivalente per piccoli segnali di un transistore. Nel prossimo paragrafo presenteremo anche la sintassi per i generatori dipendenti non lineari.

5.2.1 Sintassi dei generatori dipendenti controllati in tensione

Il nome di un generatore di tensione controllato in tensione deve cominciare con la lettera E. Ad esempio un generatore di nome E1 connesso tra i nodi 2 e 0 è individuato dalla linea

E1 2 0 NC1 NC2 VALUE .

Il valore in tensione di E1 è

$$V(E1) = [V(NC1) - V(NC2)] \text{ VALUE ,}$$

in cui NC1 e NC2 sono i due nodi del circuito che individuano la differenza di potenziale di controllo e VALUE è una costante adimensionale (guadagno in tensione).

Il nome di un generatore di corrente controllato in tensione deve cominciare con la lettera G. Ad esempio, un generatore di nome G1 connesso tra i nodi 0 e 2 è individuato dall'istruzione

G1 0 2 NC1 NC2 VALUE .

Il valore in corrente di G1 è

$$I(G1) = [V(NC1) - V(NC2)] \text{ VALUE ,}$$

dove NC1 e NC2 sono due nodi del circuito e VALUE è una costante dimensionale (trans-conduttanza). Come ulteriore esempio consideriamo la riga

GLIN5 0 5 10 15 0.1 .

Essa definisce un generatore di corrente di nome GLIN5 connesso tra i nodi 0 e 5. I nodi 10 e 15 sono quelli su cui il programma va a rilevare la differenza di potenziale di controllo. La trans-conduttanza vale 0.1 S.

5.2.2 Sintassi dei generatori dipendenti controllati in corrente

Il nome di un generatore di tensione controllato in corrente deve cominciare con la lettera H. Ad esempio un generatore di nome H1 connesso tra i nodi 2 e 0 è rappresentato dall'istruzione

```
H1    2    0    VCONTR    VALUE .
```

Il valore di tensione erogato da H1 vale

$$V(H1) = I(VCONTR) \cdot VALUE ,$$

in cui VCONTR è il nome di un generatore di tensione indipendente attraverso il quale scorre la corrente di controllo (attenzione al verso della corrente nel generatore!) e VALUE è una costante dimensionale (trans-resistenza). Ovviamente, se la corrente di controllo scorre in un ramo senza generatori è necessario introdurre un generatore di tensione fittizio usato come amperometro. Il nome di un generatore di corrente controllato in corrente deve cominciare con la lettera F. Ad esempio, un generatore di nome F1 connesso tra i nodi 0 e 2 è individuato da

```
F1    0    2    VCONTR    VALUE .
```

Il valore di corrente erogata da F1 è

$$I(F1) = I(VCONTR) \cdot VALUE ,$$

dove VCONTR è il nome di un generatore di tensione indipendente attraverso il quale scorre la corrente di controllo e VALUE è una costante adimensionale (guadagno in corrente). Anche in questo caso, ove mai nel ramo di controllo non vi sia un generatore di tensione indipendente, è necessario aggiungere uno che faccia da amperometro erogando una tensione nulla.

5.2.3 Un esempio: il circuito equivalente del transistore

Per mostrare l'utilità dei generatori controllati lineari, consideriamo il circuito in Figura 5.1, in cui abbiamo rappresentato il circuito equivalente, detto a , di un transistore.

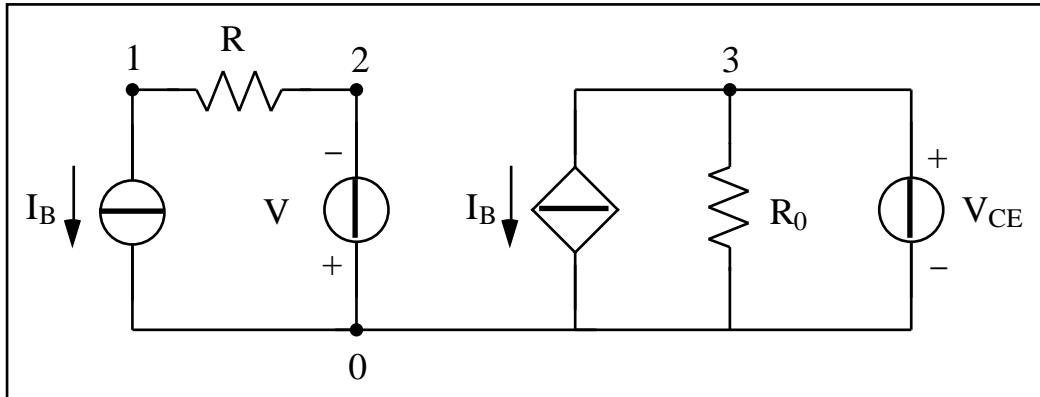


Figura 5.1: circuito equivalente per piccoli segnali di un transistore (pnp).

Il generatore di corrente tra i nodi 3 e 0 (che in un transistore sono detti ‘collettore’ ed ‘emettitore’ rispettivamente) è controllato dalla corrente I_B che scorre nella resistenza R (il morsetto 1 è detto ‘base’).

Ecco il listato Spice che permette di determinare le caratteristiche di uscita, ovvero la corrente che scorre nel collettore in funzione di $V(3)$ utilizzando come parametro la corrente di base I_B .

Esempio 5.1

* Il circuito ibrido a del transistore, $R = 1k$, $= 300$, $R_0 = 10k$

```

Rpi   1    2    1k
Vpi   0    2    DC   0
Fbeta 3    0    Vpi   300
R0    3    0    10k
IB    1    0    DC   0
VCE   3    0    DC   0
.DC   VCE    2    10    1    IB    2u    10u    2u
.PROBE
.END

```

Notate la presenza del generatore fittizio ‘ Vpi ’ che serve a misurare la corrente di base. Le caratteristiche di uscita si ottengono effettuando un’analisi ‘.DC’ in cui variano sia la tensione VCE che la corrente di base IB .

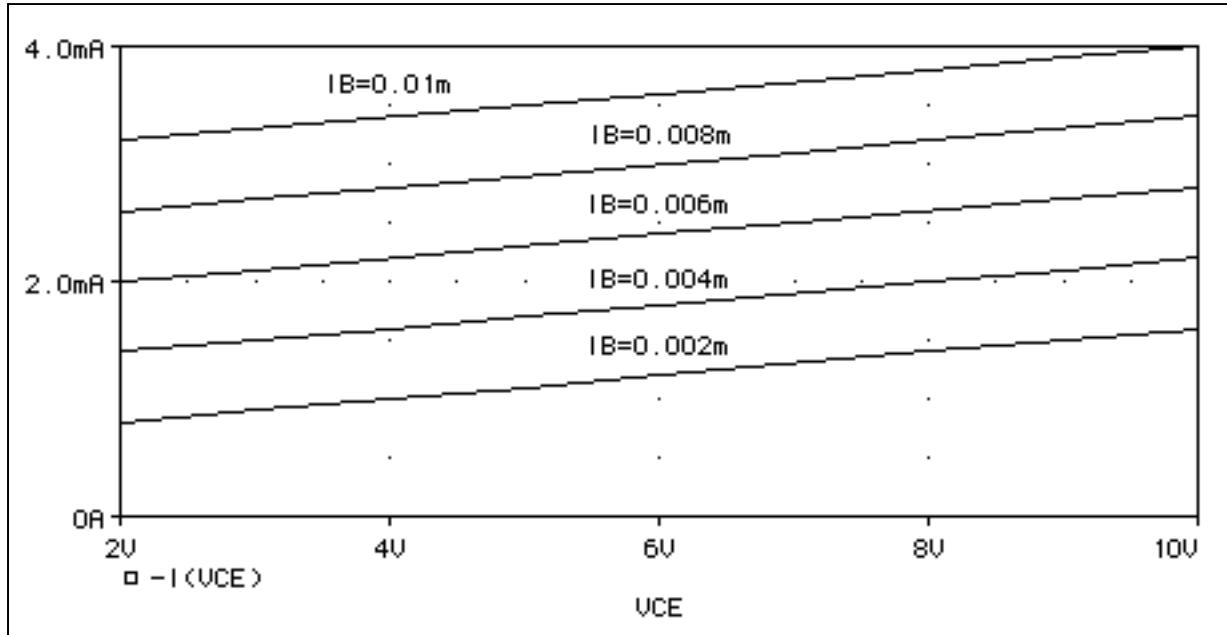


Figura 5.2: caratteristiche di uscita del transistore.

In Figura 5.2 osserviamo le caratteristiche di uscita a partire da una tensione collettore-emettitore di 2V. In realtà il comportamento del transistore è molto più complesso come avremo modo di scoprire nel capitolo dedicato allo studio dei dispositivi a semiconduttore dove parleremo, tra l'altro, più diffusamente del transistore e dei suoi modelli simulabili con Spice.

Sicuramente nel precedente esempio è da apprezzare la semplicità del modello che mette in evidenza il ‘guadagno del transistore’ rappresentato proprio dal generatore controllato e dal parametro che vale 300.

5.3 Generatori dipendenti non lineari

In Spice i quattro tipi di generatori dipendenti, controllati in tensione e in corrente, possono anche essere configurati in modo da presentare una caratteristica non lineare. Il legame tra la grandezza in uscita, tensione o corrente, e le grandezze di controllo (per i generatori non lineari possono essere più di una!) può essere espresso da una funzione polinomiale o da una combinazione di funzioni elementari o da una tabella di valori o, infine, può essere espressa nel dominio trasformato di Laplace.

5.3.1 Generatori non lineari di tipo polinomiale

In questo caso il legame tra la grandezza di uscita (tensione o corrente erogata dal generatore controllato) e le grandezze di controllo (tensioni o correnti) è espresso

mediante un polinomio in cui le grandezze di controllo agiscono da variabili indipendenti; il numero di ingressi e il grado sono stabiliti dall'utente. Cominciamo subito con un esempio chiarificatore:

EM2 1 0 POLY(2) 2 3 6 7 0 0 0 0 1 .

Il generatore di tensione controllato in tensione di nome EM2 è connesso tra i nodi 1 e 0. La sigla POLY indica, per l'appunto, che il generatore ha un legame non lineare di tipo polinomiale con due tensioni di controllo e, ovviamente, il numero 2 tra parentesi ne è testimone. Subito dopo seguono i nodi che identificano le tensioni di controllo: la coppia 2 e 3 e poi la coppia 6 e 7. Infine, seguono i coefficienti del polinomio (0 0 0 0 1) che specifica il legame non lineare: in questo caso vedremo che la tensione di uscita è uguale al prodotto delle tensioni di controllo.

La regola che assegna i coefficienti al polinomio multidimensionale è piuttosto complessa. Per un polinomio con n grandezze di controllo, il primo coefficiente è un termine costante, i successivi n coefficienti stabiliscono un legame lineare con le grandezze di controllo e a partire dal coefficiente n + 2 cominciano i termini non lineari che coinvolgono prodotti tra coppie di grandezze di controllo, in ogni combinazione possibile e con tutte le potenze possibili. Resta inteso che nella pratica non sono mai necessari prodotti di ordine elevato per cui anche noi arresteremo l'esposizione della regola al grado secondo, ritenendo che questo sia sufficiente a coprire la maggior parte delle necessità. Consideriamo la riga generica

ENAME N1 N2 POLY(n) NC1 NC2 ... P0 P1

Questa è la sintassi da utilizzare per un generatore di tensione controllato da n tensioni V(NC1, NC2), ..., V(NCn-1, NCn), i coefficienti P0, P1, ... sono quelli del polinomio. Analogamente, per un generatore di corrente controllato da n tensioni

GNAME N1 N2 POLY(n) NC1 NC2 ... P0 P1

Invece, per un generatore di corrente controllato da n correnti

FNAME N1 N2 POLY(n) VCTRL1 VCTRL2 ... P0 P1 ... ,

dove VCTRL1, ..., VCTRLn sono gli n generatori di tensione in cui scorrono le n correnti di controllo (attenzione ai versi delle correnti nei generatori!). Ricordiamo che è sempre possibile ricorrere a un generatore fittizio con tensione nulla in quei rami in cui non sono disponibili generatori indipendenti. Infine, per un generatore di tensione controllato da n correnti

HNAME N1 N2 POLY(n) VCTRL1 VCTRL2 ... P0 P1

Il legame funzionale tra la grandezza di uscita OUT e, ad esempio, le n tensioni di controllo è

$$\begin{aligned} \text{OUT}(V_1, V_2, \dots, V_n) = & P_0 + P_1 V_1 + P_2 V_2 + \dots + P_n V_n + \\ & P_{n+1} V_1 V_1 + P_{n+2} V_1 V_2 + \dots + P_{2n} V_1 V_n + \\ & P_{2n+1} V_2 V_2 + P_{2n+2} V_2 V_3 + \dots + P_{3n-1} V_2 V_n + \\ & \vdots \\ & P_{n!/(2(n-2)!)+2n} V_n V_n + \\ & P_{n!/(2(n-2)!)+2n+1} V_1^2 V_1 + P_{n!/(2(n-2)!)+2n+2} V_1^2 V_2 + \\ & \vdots \end{aligned}$$

per cui P_0 è un termine costante indipendente dalle grandezze di controllo, $P_1 \dots P_n$ sono i coefficienti che legano linearmente le grandezze di controllo alla grandezza di uscita. Da P_{n+1} cominciano i termini non lineari. Nell'esempio precedente $P_{n+2} = 1$ indicava il prodotto tra le due grandezze di controllo, infatti per due tensioni di controllo il legame sarà del tipo

$$\begin{aligned} \text{OUT}(V_1, V_2) = & P_0 + P_1 V_1 + P_2 V_2 + \\ & P_3 V_1^2 + P_4 V_1 V_2 + P_5 V_2^2 + \\ & P_6 V_1^3 + P_7 V_1^2 V_2 + P_8 V_1 V_2^2 + P_9 V_2^3 + \dots \end{aligned}$$

Vale la pena notare che è necessario includere tutti i coefficienti nulli fino al primo diverso da zero, poi i successivi all'ultimo diverso da zero saranno ovviamente considerati nulli.

Infine, nel caso in cui si abbia una sola grandezza di controllo, il legame funzionale si semplifica nel seguente modo

$$\text{OUT}(V_1) = P_0 + P_1 V_1 + P_2 V_1^2 + \dots + P_n V_1^n,$$

in cui P_0 rappresenta una costante, P_1 esprime il legame lineare, P_2 la presenza di un termine quadratico e così via.

Quest'ultimo caso, di una sola grandezza di controllo, permette di realizzare generatori dipendenti le cui caratteristiche sono funzioni sviluppabili in serie di potenze. Basta infatti considerare, come coefficienti del polinomio, proprio quelli dello sviluppo in serie di potenze della funzione da realizzare.

Ad esempio, il seguente listato realizza un generatore di tensione controllato in tensione la cui caratteristica approssima la funzione $V_{out} = \log(1+V_{in})$.

Esempio 5.2

* Generatore controllato logaritmico

```

Vin      1    0    DC    0
Eout     2    0    POLY(1)  1  0  0  1  -0.5   0.333  -0.25
.DC      Vin    0    1    0.05
.PROBE
.END

```

Abbiamo introdotto i primi quattro coefficienti dello sviluppo in serie di $\log(1+x)$ e ovviamente ciò introduce un'approssimazione (che sappiamo essere tanto più piccola quanto più x è piccolo rispetto a 1), ma il vantaggio è che, con tale tecnica, è possibile costruire tutte le forme d'onda sviluppabili in serie di potenze.

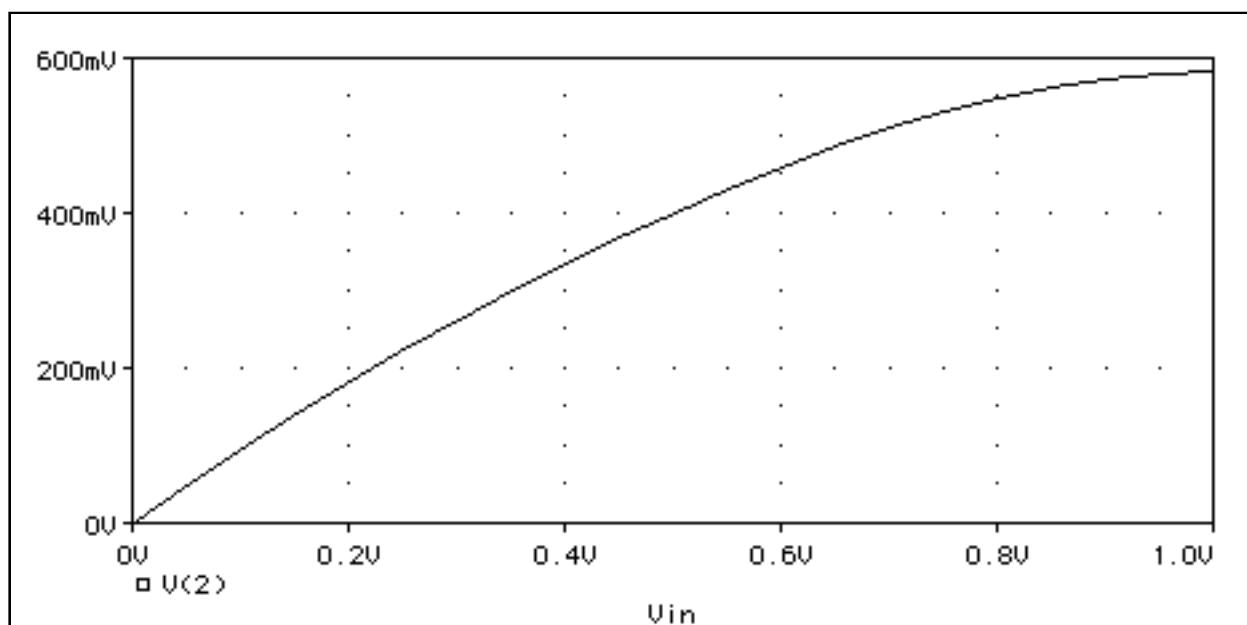


Figura 5.3: generatore non lineare che realizza la funzione $\log(1+V_{in})$.

Nella Figura 5.3 notiamo che la tensione di uscita ottenuta è una buona approssimazione di $\log(1+V_{in})$ laddove la tensione V_{in} è piccola rispetto a 1V. Come vedremo nel prossimo paragrafo, bisogna dire che per la funzione logaritmo naturale esiste anche la possibilità di realizzare direttamente un generatore dipendente con tale caratteristica, ma ripetiamo che la precedente tecnica trova la sua migliore applicazione con le funzioni che si possono espandere in serie di potenze, non incluse nel bagaglio di funzioni elementari di Spice.

5.3.2 Generatori dipendenti controllati in tensione di vario tipo

Per i generatori dipendenti controllati in tensione (e non per quelli controllati in corrente!) esiste la possibilità di definire la caratteristica di funzionamento attraverso altre ‘strade’. Cominciamo con l’opzione TABLE: essa definisce la caratteristica del generatore attingendo i valori da una tabella. Si stabilisce una corrispondenza tra i valori assunti da una espressione di tensioni del circuito e i corrispondenti valori assunti dal generatore dipendente. Tutto quello che occorre fare è fornire le coppie di valori fino a un massimo di $2^{11} = 2048$ coppie. Con l’esempio che segue introduciamo anche la sintassi del comando.

Esempio 5.3

```
* Generatore dipendente di tipo TABLE
Vin      1    0    DC    1
Eout     2    0    TABLE {V(1)} = (0,0) (2.4,0) (2.6,5) (5,5)
.DC      Vin   0    5    0.25
.PRINT   DC    V(2)
.PROBE
.END
```

L’esempio 5.3 mostra un legame tra i valori assunti dalla tensione al nodo 1 e il generatore Eout. Come si evince dalle coppie di valori in tabella, quando la tensione al nodo 1 vale 0 V allora l’uscita vale 0 V, quando la tensione al nodo 1 vale 2.6 V allora quella al generatore Eout vale 5 V, e così via. Per provare quanto abbiamo realizzato, utilizziamo un’analisi in continua ‘.DC’ e, in Figura 5.4, possiamo notare il legame ingresso uscita che riusciamo a realizzare. La funzione ottenuta potrebbe essere quella di un comparatore a soglia semplificato.

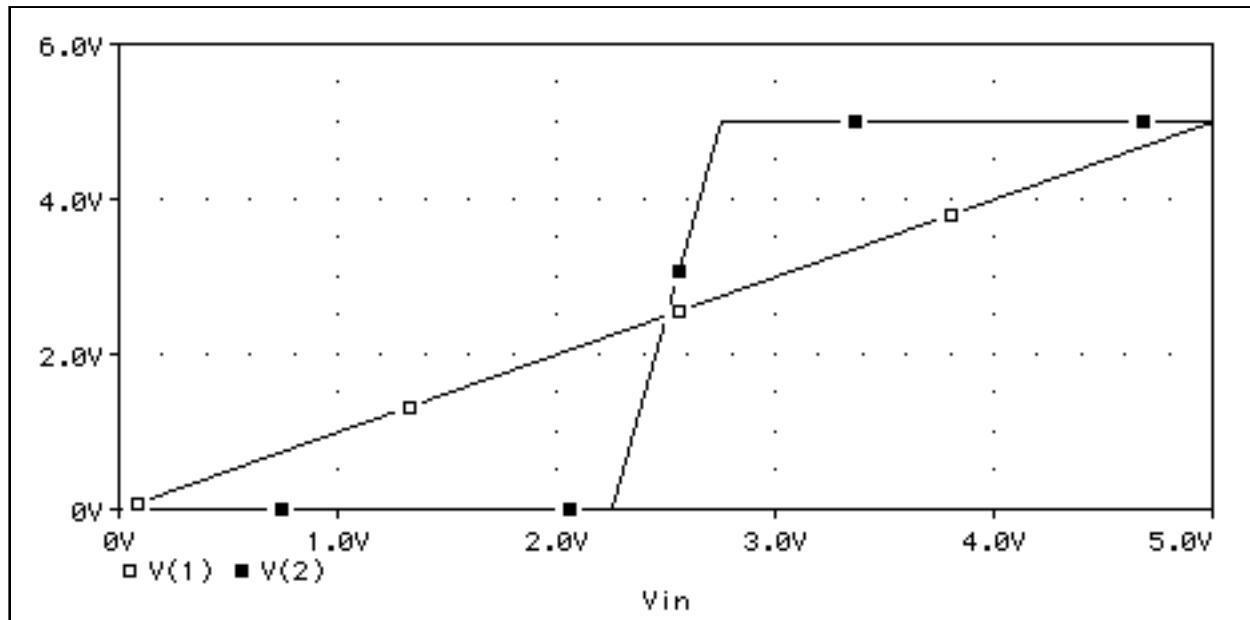


Figura 5.4: generatore dipendente di tipo TABLE.

Riportiamo di seguito anche la tabella di valori così come la fornisce l'istruzione '.PRINT DC' nel file '.OUT'.

Vin	V(2)
0.000E+00	0.000E+00
2.500E-01	0.000E+00
5.000E-01	0.000E+00
7.500E-01	0.000E+00
1.000E+00	0.000E+00
1.250E+00	0.000E+00
1.500E+00	0.000E+00
1.750E+00	0.000E+00
2.000E+00	0.000E+00
2.250E+00	0.000E+00
2.500E+00	2.500E+00
2.750E+00	5.000E+00
3.000E+00	5.000E+00
3.250E+00	5.000E+00
3.500E+00	5.000E+00
3.750E+00	5.000E+00
4.000E+00	5.000E+00
4.250E+00	5.000E+00
4.500E+00	5.000E+00
4.750E+00	5.000E+00
5.000E+00	5.000E+00

Questo tipo di generatore è utile in tutti quei casi in cui il risultato di un altro programma ha fornito un file con i valori che assume una grandezza. Con qualche piccola modifica al file questo può diventare la definizione di un generatore tipo

TABLE, può essere incluso in un file Spice e permettere quindi altre elaborazioni utilizzando Spice.

Una possibilità ancora più comoda è quella offerta dall'opzione VALUE che permette di utilizzare, per la caratteristica del generatore dipendente, una qualsiasi espressione di funzioni elementari comprese in Spice, funzioni delle tensioni e correnti del circuito.

Esempio 5.4

```
* Generatore dipendente di tipo VALUE
Vin 1 0 DC 1
Eout 2 0 VALUE = {LOG(1+V(1))}
.DC Vin 0 2 0.05
.PROBE
.END
```

L'esempio 5.4 mostra un generatore che realizza la funzione $\log(1+V_{in})$ (la funzione 'log' rappresenta il logaritmo in base 'e', detto anche neperiano) però, stavolta, senza alcuna limitazione nei valori assunti dall'ingresso.

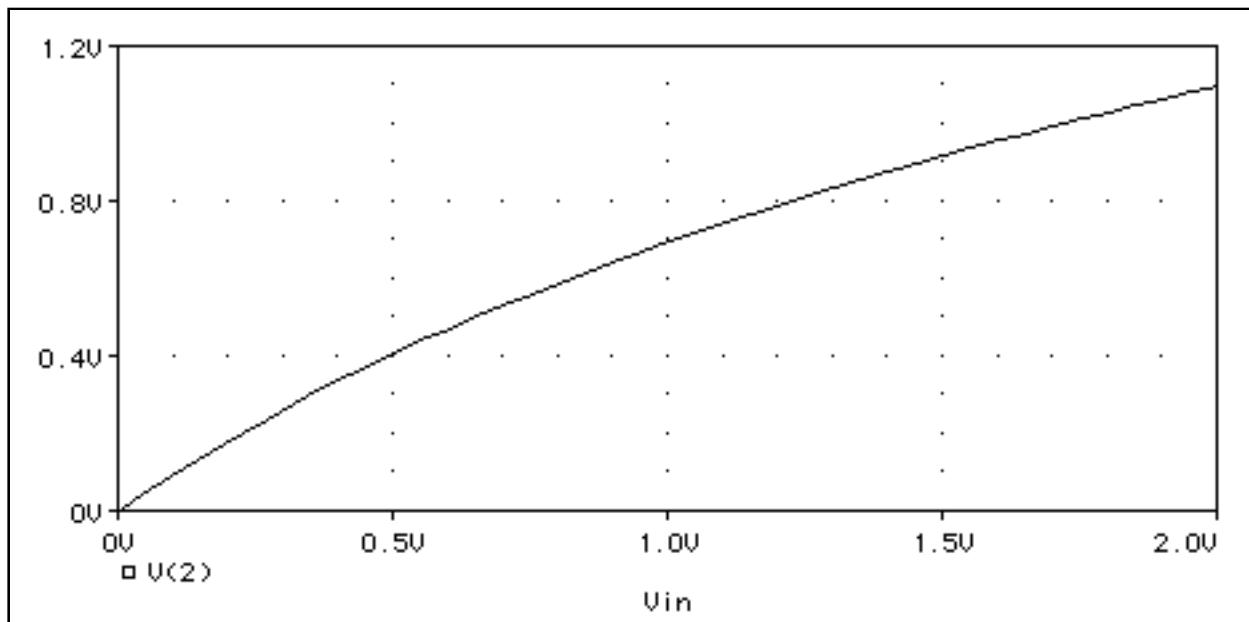


Figura 5.5: generatore dipendente di tipo VALUE.

In Figura 5.5 abbiamo riportato l'andamento della tensione sul generatore. Provate voi a realizzare un circuito che contenga questo generatore e quello approssimato che abbiamo introdotto nel precedente paragrafo: potrete notare la differenza tra i due quando la tensione di controllo diventa circa 1.

Questa opzione può anche essere utile per modellare il comportamento di dispositivi che hanno una caratteristica ingresso-uscita espressa da una funzione elementare. Ad esempio, possiamo modellare la caratteristica diretta di un diodo che è espressa dalla formula

$$I_D = I_S (e^V - 1),$$

in cui I_D e V sono, rispettivamente, la corrente che passa attraverso il diodo e la tensione applicata ai suoi capi. I parametri I_S e V_{TH} sono delle costanti caratteristiche per ogni diodo, come avremo modo di discutere in maggior dettaglio in seguito. In Figura 5.6 possiamo notare il circuito equivalente che realizza la precedente formula.

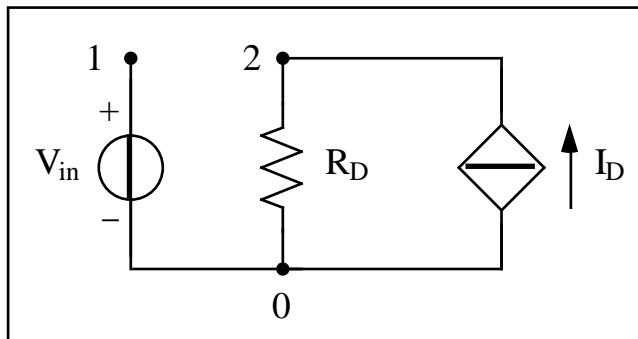


Figura 5.6: circuito equivalente in continua di un diodo.

Consideriamo allora il seguente listato Spice.

Esempio 5.5

```
* Circuito equivalente in continua di un diodo
.PARAM IS=3M, VTH=0.7
Vin 1 0 DC 1
GD 0 2 VALUE={IS*(EXP(V(1)/VTH)-1)}
RD 2 0 1
.DC Vin 0.05 5 0.05
.PROBE
.END
```

Esso definisce proprio un generatore di corrente che rappresenta la caratteristica di un diodo; notate che i valori I_S e V_{TH} sono stati impostati ricorrendo all'istruzione '.PARAM' che permette di specificare delle costanti che possono essere utilizzate nelle espressioni presenti nel listato. Il vantaggio è che la formula

conserva l'aspetto della ‘teoria’ e che le costanti possono essere modificate su una sola riga senza entrare nel listato. V_{th} è detta tensione di soglia ed è l'inverso del parametro introdotto in precedenza. Operiamo una analisi .DC che permette di ottenere la caratteristica del diodo. In Figura 5.7 possiamo vedere il risultato ottenuto.

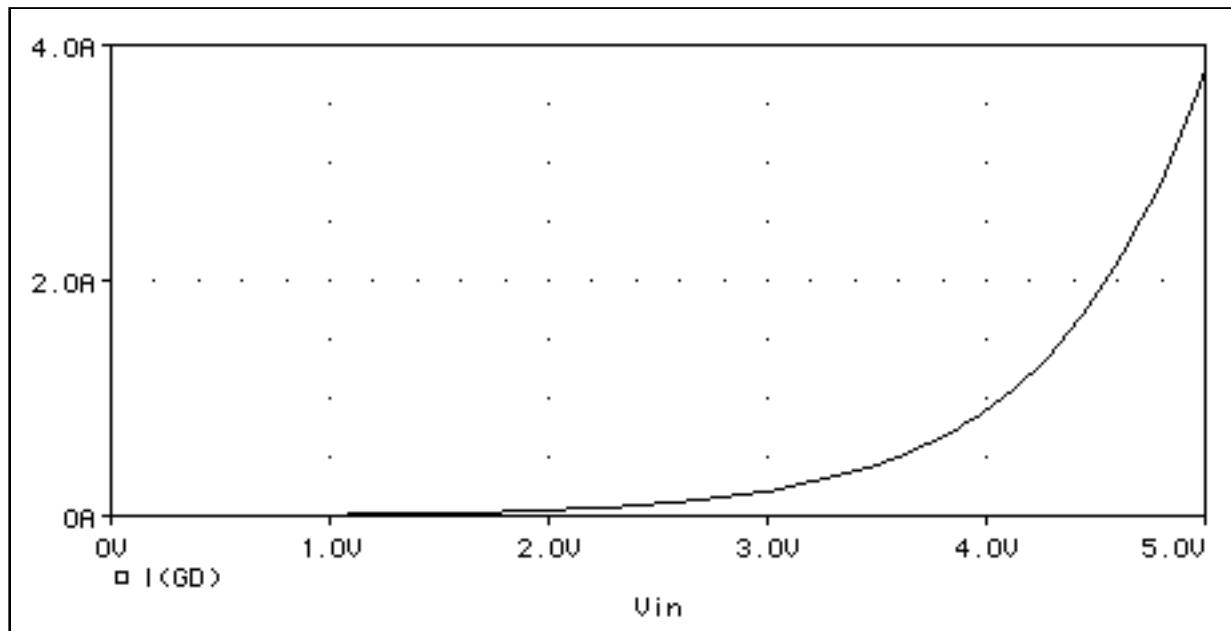


Figura 5.7: caratteristica diretta di un diodo simulata da un generatore dipendente.

In seguito paragoneremo questo risultato con il vero modello del diodo a semiconduttore.

Infine, con l'opzione LAPLACE c'è la possibilità di includere addirittura generatori dipendenti la cui caratteristica è espressa nel dominio trasformato di Laplace. Questo vuol dire che tali generatori possono rappresentare la funzione di trasferimento di un blocco funzionale anche complesso: basta semplicemente specificarne i poli e gli zeri.

Esempio 5.6

```
* Generatore dipendente di tipo LAPLACE
Vin 1 0 AC 1
Eout 2 0 LAPLACE {V(1)}={1/(1+1.591e-4*s)}
.AC DEC 10 10 10k
.PROBE
.END
```

L'esempio 5.6 illustra un generatore che realizza un filtro passa-basso del primo ordine (esso presenta un solo polo) con frequenza di taglio uguale a 1 kHz. Per controllarne il comportamento, utilizziamo un'analisi in frequenza, con 10 punti per decade, tra le frequenze di 10 Hz e 10 kHz.

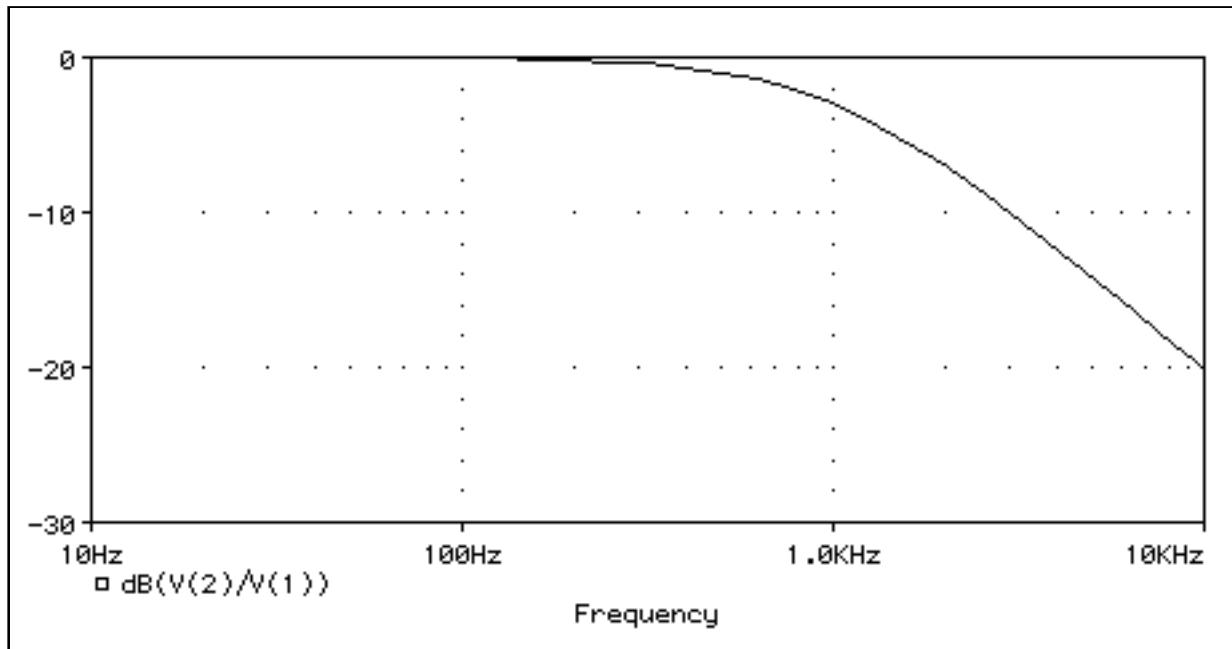


Figura 5.8: generatore dipendente di tipo LAPLACE.

In Figura 5.8 abbiamo riportato il grafico del rapporto tra la tensione del generatore dipendente e il generatore di controllo. La scala delle ordinate è logaritmica e si nota che alla frequenza di taglio il segnale è - 3 dB sotto il riferimento (0 dB), così come la teoria insegnava.

5.4 Sottocircuiti

Spice offre la possibilità di utilizzare i sottocircuiti per facilitare la descrizione di reti particolarmente complesse. Ciò significa che possiamo prendere una parte di circuito e descriverla separatamente in un sottocircuito e quest'ultimo sarà individuato da un proprio nome e ‘visibile’ all'esterno mediante dei nodi di interfacciamento. Il vantaggio è che tutto ciò può essere applicato a quelle parti del circuito che si ripetono identicamente in più punti e che possono essere definite una volta sola e poi ‘richiamate’ laddove occorrono.

Un esempio classico è l'amplificatore operazionale, un circuito abbastanza complesso e che contiene molte decine di componenti al suo interno, ma che si interfaccia all'esterno, per quanto riguarda i segnali, mediante tre soli morsetti.

Inserire due o più operazionali dello stesso tipo in una rete vorrebbe dire ripetere le stesse righe di programma più volte dovendo cambiare solo i nodi e questo aumenterebbe il rischio di commettere degli errori. Invece, con i sottocircuiti tutta la rete che rappresenta l'operazionale è racchiusa in una ‘scatola’ e quindi la definizione di un circuito che fa uso di più operazionali diventa estremamente semplice.

Il concetto è perfettamente analogo a quello di sottoprogramma nei linguaggi di programmazione in cui una funzione particolare può essere definita da alcune righe di programma speciali e poi può essere richiamata, ogni volta che occorre, nel programma principale.

Concretizziamo le cose dette in un esempio. Consideriamo il circuito in Figura 5.9 che rappresenta un filtro passa-basso del secondo ordine

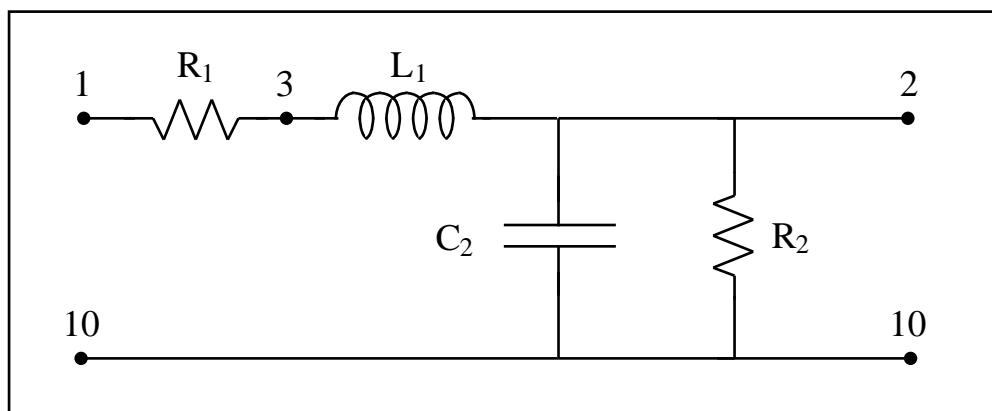


Figura 5.9: filtro passa-basso del secondo ordine.

Ecco le linee di programma che definiscono questo filtro.

```
.SUBCKT LPF2 1 2 10
R1 1 3 1
L1 3 2 1m
C2 2 10 1u
R2 2 10 100k
.ENDS LPF2
```

La prima linea individua l'inizio del sottocircuito (.SUBCKT), LPF2 è il nome che abbiamo scelto, poi seguono i tre nodi ‘esterni’ del sottocircuito; le restanti linee definiscono la topologia e gli elementi che lo compongono. L'ultima istruzione comunica che il sottocircuito di nome LPF2 è finito.

5.4.1 Sintassi dei sottocircuiti

La sintassi per definire un sottocircuito è

```
.SUBCKT NAME [NODI] ,
```

in cui NAME definisce il nome con cui sarà richiamato il sottocircuito. Esso deve cominciare necessariamente con una lettera e non con un numero. [NODI] rappresenta la sequenza di nodi con cui il sottocircuito si interfaccia con l'esterno. La sequenza di nodi è opzionale: possono esistere sottocircuiti senza nodi di interfaccia.

Poi seguono tutte le istruzioni che definiscono la struttura del sottocircuito che si conclude con una riga

```
.ENDS NAME .
```

La ripetizione del nome nella riga ‘.ENDS’ non è necessaria ma è raccomandabile per una maggiore leggibilità del listato, specialmente nel caso in cui ci siano più sottocircuiti.

Notate bene che la numerazione dei nodi del sottocircuito è locale e non è ‘vista’ dalla rete che fa uso del sottocircuito. L'unica eccezione è il nodo 0 che rappresenta il riferimento per i potenziali: esso resta comunque un nodo globale. In un sottocircuito è possibile definire dei dispositivi, tipo diodi o transistori, mediante istruzioni ‘.MODEL’ (che, più avanti, studieremo). Queste definizioni, tuttavia, avranno un valore locale e non potranno essere usate all'esterno del sottocircuito, a meno che, ovviamente, non siano ripetute.

Infine, le definizioni di sottocircuiti non possono essere innestate: all'interno di un sottocircuito non possiamo definirne un altro mediante la sequenza ‘.SUBCKT’ e ‘.ENDS’.

5.4.2 Sintassi della chiamata ai sottocircuiti

Dopo che un sottocircuito è stato definito mediante le regole e le modalità spiegate nel precedente paragrafo, esso può essere richiamato in un circuito facendo uso della seguente riga di programma

```
XNAME [NODI] NAME ,
```

dove XNAME è il nome dell'istanza di sottocircuito in cui la lettera X è obbligatoria e poi seguono caratteri e numeri fino ad un massimo di 8 (così come per qualsiasi altro nome Spice). Ovviamente, il numero di nodi della sequenza [NODI] deve essere uguale al numero di nodi che definiscono il sottocircuito individuato dal nome NAME.

Ad esempio, ecco una possibile riga che chiama il filtro del secondo ordine che abbiamo introdotto nei paragrafi precedenti

```
X12    10    20    0    LPF2
```

in cui il nome del filtro LPF2 nella rete chiamante è X12 ed è connesso ai nodi 10, 20 e 0. Prestate la massima attenzione alla sequenza dei nodi, andando sempre a riguardare il numero e il significato dei nodi del sottocircuito che volete richiamare. Un eventuale errore nell'ordine dei nodi non produrrebbe alcun errore sintattico (Spice fornirebbe comunque dei risultati in uscita), ma produrrebbe un sicuro errore semantico nei risultati ottenuti!

Le chiamate ai sottocircuiti possono essere innestate. Possiamo richiamare un sottocircuito A che all'interno ha una chiamata a un sottocircuito B e così via. Però le chiamate non possono essere ‘circolari’, ovvero B non può avere al suo interno una ulteriore chiamata al sottocircuito A!

Infine, nell'analisi dei risultati con Probe, i sottocircuiti possono essere espansi per analizzare le grandezze elettriche interne. Ad esempio, la corrente che scorre nel resistore R1 del filtro LPF2 che abbiamo introdotto prima è I(X12.R1), ovvero X12. è il ‘prefisso’ per individuare gli elementi del sottocircuito LPF2 di nome X12. Se avessimo introdotto un altro filtro LPF2 con la seguente riga

```
XA1    11    25    12    35    LPF2 ,
```

allora il resistore R1 di questa realizzazione del sottocircuito LPF2 sarebbe stato individuato da XA1.R1.

5.4.3 Sottocircuiti con parametri

Spice offre un ulteriore strumento nell'utilizzo dei sottocircuiti: la possibilità di definire sottocircuiti con parametri mediante l'opzione PARAMS (attenzione a non confonderla con l'istruzione .PARAM!). Questo significa che alcuni valori che definiscono i componenti possono essere passati come argomenti nella riga .SUBCKT che definisce il sottocircuito, per poi essere utilizzati all'interno di

espressioni presenti nella righe di programma interne. Consideriamo subito la seguente definizione di sottocircuito

```
* Filtro passa-basso parametrizzato
*
.SUBCKT LPF 1 2 10 PARAMS: ft={4000}
*
* ft frequenza di taglio del filtro
*
R1 1 2 10
C1 2 10 {1/(2*3.14159*10*ft)}
.ENDS LPF
```

Esso definisce un filtro passa-basso del primo ordine in cui passiamo la frequenza di taglio del filtro (ricordiamo che è la frequenza per cui l'attenuazione dell'uscita è già 3 dB sotto il riferimento 0 dB); poi il valore della resistenza è fisso mentre quello del condensatore viene ricavato rispetto alla frequenza di taglio mediante la formula

$$t = 2 \quad f_t = \frac{1}{R_1 C_1} .$$

Inoltre, notiamo la disposizione dei commenti che permette di ricavare agevolmente la funzione dei nodi e il significato dei parametri.

Un possibile circuito che richiama il filtro passa-basso appena introdotto potrebbe essere

```
Esempio 5.7
* Circuito con filtro passa-basso
.INC 'LPF.CIR'
V1 1 0 AC 1
X1 1 2 0 LPF PARAMS: ft=8000
R1 2 0 100
.AC DEC 10 100 10k
.PROBE
.END
```

Innanzitutto, notiamo l'istruzione .INC che permette di includere il file esterno LPF.CIR nel listato attuale. Il nome del file segue poi il '.INC' che va tra

virgolette; le modalità di scrittura del percorso (path) per raggiungere il file dipendono dal sistema operativo su cui gira Spice. Ovviamente in questo caso avremo avuto l'accortezza di salvare il sottocircuito precedente in un file di nome LPF.CIR.

Tornando al nostro esempio, in esso riconosciamo un generatore di tensione in ingresso e una resistenza di carico in uscita. Notiamo la modalità con cui richiamiamo il filtro: dopo il nome LPF segue la lista dei parametri da passare al sottocircuito; in questo caso abbiamo utilizzato una frequenza di taglio di 8 kHz. Infine, ricordiamo che è comunque necessario stabilire un valore dei parametri nella riga .SUBCKT che li definisce, anche se poi la chiamata al sottocircuito li sostituirà con altri valori.

5.4.4 Semplice modulatore AM

Come ulteriore esempio di utilizzo dei sottocircuiti con parametri e dei generatori dipendenti non lineari forniti nei precedenti paragrafi, di seguito riportiamo una possibile realizzazione di un modulatore AM (Amplitude Modulation) dal funzionamento ideale. Ricordiamo che un segnale modulato AM è per definizione

$$V_{out}(t) = [1 + S_a V_m(t)] \sin(2\pi f_0 t),$$

dove $V_{out}(t)$ è il segnale modulato, S_a è una costante ed è detta indice di modulazione, $V_m(t)$ è il segnale modulante e infine f_0 è il segnale della portante. Ecco un sottocircuito che realizza tale funzione

```
.SUBCKT MODAM 1 2 3 10  PARAMS: A=1 fo=1 Sa=1
*
R1 1 10 1
V2 2 10 SIN(0 {A} {fo} 0 0)
R2 2 10 1
ED 3 10 VALUE={V(2)*(1+(Sa*V(1)))}
.ENDS MODAM
```

Il sottocircuito MODAM ha quattro nodi di interfacciamento; il nodo 1 è l'ingresso del segnale modulante, il nodo 2 è l'uscita del segnale portante puro e il nodo 3 è l'uscita del segnale modulato e, infine, il nodo 10 è il riferimento comune. I parametri a disposizione sono l'ampiezza A del segnale portante, la frequenza fo del segnale portante e Sa indice di modulazione. Notiamo il generatore di tensione, controllato in tensione, che realizza direttamente la

funzione di modulazione AM. Ecco di seguito il semplice circuito che utilizza il sottocircuito MODAM.

Esempio 5.8

* Circuito modulatore AM

```
.PARAM val_Sa=0.3
Vin 1 0 PULSE(0 1 0 1ns 1ns 10us 20us)
XMOD 1 2 3 0 MODAM PARAMS: A=1 fo=666k Sa={val_Sa}
.STEP PARAM val_Sa LIST 0.3, 0.7
.TRAN 100n 30u
.PROBE
.END
```

Il generatore di ingresso Vin definisce un impulso periodico che agisce da segnale modulante. Abbiamo fissato l'ampiezza della portante a 1 V, la frequenza della portante a 666 kHz (tipico valore di un segnale AM per uso radiofonico) e, poi, effettuiamo un'analisi in transitorio per due valori dell'indice di modulazione.

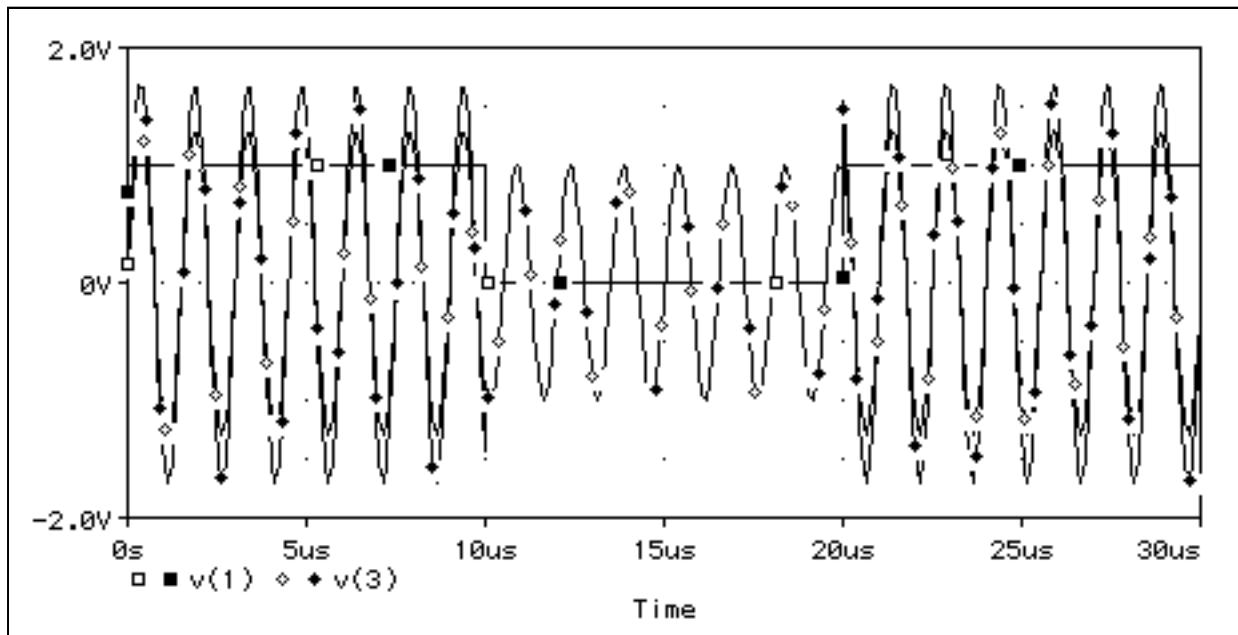


Figura 5.10: ingressi e uscite del modulatore AM.

Quest'ultimo punto merita qualche commento ulteriore: dopo aver definito il parametro val_Sa mediante l'istruzione '.PARAM', abbiamo 'passato' al sottocircuito MODAM proprio il valore val_Sa; successivamente, con l'aiuto dell'istruzione .STEP, facciamo variare un parametro (PARAM), in particolare val_Sa, fornendo una lista (LIST) dei valori che assumerà. Il risultato è che

PROBE ci fornirà entrambe le simulazioni e noi saremo in grado sovrapporre i grafici per apprezzarne le differenze, come mostrato in Figura 5.10.

5.5 L'istruzione '.MODEL'

L'istruzione '.MODEL' permette di specificare le caratteristiche di un dispositivo in modo da ottenere un modello ben definito e individuato da un nome che può essere richiamato poi nei circuiti. Questo modello può essere locale al circuito oppure può essere incluso in una libreria di modelli (.LIB) per essere utilizzato da tutti i circuiti che includeranno (.INCLUDE) quella libreria.

La sintassi generica è

```
.MODEL      NAME      MTYPE <P1=VAL1, P2=VAL2, ... > .
```

NAME è il nome particolare che si assegna al modello ed è buona norma utilizzare come prima lettera quella del tipo di dispositivo (Dxxx per i diodi, Rxxx per i resistori e così via) anche se non è obbligatorio.

MTYPE individua il tipo di dispositivo di cui si vogliono fissare i parametri. Nella seguente tabella riportiamo i tipi più comuni e che saranno studiati in questo libro.

Tipo del modello	MTYPE
Condensatore	CAP
Induttore	IND
Nucleo ferromagnetico	CORE
Diodo	D
FET a canale n	NJF
MOSFET a canale n	NMOS
Transistor bipolare tipo npn	NPN
FET a canale p	PJF
MOSFET a canale p	PMOS
Transistor bipolare tipo pnp	PNP
Resistore	RES

Poi nella riga del modello seguono i parametri di cui vogliamo fissare i valori; anche se non si fissa il valore di alcun parametro, il simulatore considererà, per i parametri non specificati, i valori di 'default'. La lista dei nomi dei parametri per i dispositivi a semiconduttore, il loro significato e i valori di 'default' saranno spiegati in un prossimo capitolo.

Qui di seguito riportiamo una descrizione più dettagliata del modello che riguarda i bipoli più comuni: resistori, condensatori e induttori.

5.5.1 Modello di resistore

Il modello di un resistore in Spice include gli effetti lineari e quadratici, sul valore della resistenza, provocati da variazioni della temperatura. La linea di programma ‘.MODEL’ per un resistore assume la forma

```
.MODEL      RNAME    RES (R=VAL1, TC1=VAL2, TC2=VAL3) ,
```

in cui RNAME è il particolare nome assegnato al modello di resistenza. I parametri fra parentesi sono spiegati nella seguente tabella.

Parametro	Significato	Unità di misura	Default
R	Coefficiente moltiplicativo	adimensionale	1
TC1	Coefficiente lineare di temperatura	1/°C	0
TC2	Coefficiente quadratico di temperatura	1/°C ²	0

e la formula che esprime il valore vero e proprio della resistenza è

$$<\text{VALUE}> \quad R \left[1 + T_{C1}(T - TNOM) + T_{C2}(T - TNOM)^2 \right] ,$$

dove <VALUE> è il valore di resistenza nella riga che chiama il modello, TNOM è la temperatura nominale per cui sono definiti i valori (essa vale 27 °C e può essere cambiata mediante l’istruzione .OPTIONS che spiegheremo nell’ultimo capitolo) e T è la temperatura attuale. Ad esempio, le istruzioni

```
.MODEL      RLIN    RES (R=1, TC1=0.002)
R1      1      2      RLIN      10
```

definiscono un resistore di nome R1 che utilizza il modello RLIN e ha un valore nominale di 10 , però il valore nominale cambia di 0.02 per ogni variazione di un grado della temperatura.

È anche possibile definire un modello di variazione esponenziale con la temperatura in cui si fissa il parametro R e il parametro TCE (il cui valore di default è zero) e in tal caso la resistenza varia con la seguente legge

<VALUE> R 1.01^{TCE (T-TNOM)}.

Consideriamo come esempio di tale modello il seguente listato Spice

Esempio 5.9

```
* Modello di resistore lineare in temperatura
.OPTIONS TNOM=25
.MODEL      RLIN    RES(R=1, TC1=0.002)
Iin     0     1     DC     1
RT      1     0     RLIN   10
.TEMP    25     30     50     100
.END
```

L'istruzione '.OPTIONS' permette di variare una serie di parametri interni del simulatore tra cui vi è anche il parametro TNOM che fissa la temperatura nominale a cui devono essere riferiti tutti i valori forniti nei modelli ed è anche la temperatura alla quale vengono effettuate le analisi. Poi c'è il modello lineare per la resistenza e un generatore di corrente unitario. Segue la resistenza che utilizza il modello RLIN e che ha valore 10 . Per la prima volta incontriamo l'istruzione '.TEMP' che permette di ripetere le analisi presenti nel circuito alle temperature specificate sulla stessa linea e riportate in gradi centigradi. Quando è presente questa istruzione, nel file '.OUT' ritroviamo l'aggiornamento dei valori dei componenti alle nuove temperature. Infatti nel nostro caso per il primo valore non ritroviamo nulla perché è il valore nominale, poi, ad esempio, alla temperatura di 30°C

```
**** TEMPERATURE-ADJUSTED VALUES TEMPERATURE = 30.000 DEG C
*****
**** RESISTORS
NAME      VALUE
RT       1.010E+01
```

che ci ricorda che l'unico resistore RT, alla temperatura di 30 °C, ha un valore di 10.1 .

5.5.2 Modello di condensatore

Nel modello per il condensatore, oltre agli effetti della temperatura, è inclusa anche una dipendenza lineare e quadratica dalla tensione applicata ai capi del

condensatore. La linea di programma ‘.MODEL’ per un condensatore assume la forma

```
.MODEL CNAME CAP(C, VC1, VC2, TC1, TC2),
```

dove CNAME è il particolare nome assegnato al modello di capacità. I parametri fra parentesi sono spiegati nella seguente tabella.

Parametro	Significato	Unità di misura	Default
C	Coefficiente moltiplicativo	adimensionale	1
VC1	Coefficiente lineare di tensione	1/V	0
VC2	Coefficiente quadratico di tensione	1/V ²	0
TC1	Coefficiente lineare di temperatura	1/°C	0
TC2	Coefficiente quadratico di temperatura	1/°C ²	0

La formula che esprime il valore vero e proprio della capacità è

$$<\text{VALUE}> C [1 + V_{C1} V + V_{C2} V^2] [1 + T_{C1} (T - TNOM) + T_{C2} (T - TNOM)^2],$$

dove <VALUE> è il valore di capacità nella riga che chiama il modello, TNOM è la temperatura nominale per cui sono definiti i valori (essa vale 27 °C e può essere cambiata mediante l’istruzione ‘.OPTIONS’) e T è la temperatura attuale e infine V è la tensione applicata ai capi del condensatore. Ad esempio, le istruzioni

```
.MODEL CQUAD CAP(C=1,VC2=0.002)
C1    1    2    CQUAD    10u
```

definiscono un condensatore di nome C1 che utilizza il modello CQUAD e ha un valore nominale di 10 μF, però il valore nominale varia con la seguente legge rispetto alla tensione applicata

$$10 (1 + 0.002 V^2) \mu\text{F}.$$

5.5.3 Modello di induttore

La linea di programma ‘.MODEL’ che definisce il modello di un induttore assume la forma

```
.MODEL      LNAME   IND(L, IL1, IL2, TC1, TC2)
```

dove LNAME è il particolare nome assegnato al modello di induttanza. I parametri fra parentesi sono spiegati nella seguente tabella e sono analoghi a quelli già visti per il condensatore

Parametro	Significato	Unità di misura	Default
C	Coefficiente moltiplicativo	adimensionale	1
IL1	Coefficiente lineare di corrente	1/A	0
IL2	Coefficiente quadratico di corrente	1/A ²	0
TC1	Coefficiente lineare di temperatura	1/°C	0
TC2	Coefficiente quadratico di temperatura	1/°C ²	0

La formula che esprime il valore dell'induttanza è

$$<\text{VALUE}> L \left(1 + I_{L1} I + I_{L2} I^2 \right) \left[1 + T_{C1} (T - TNOM) + T_{C2} (T - TNOM)^2 \right],$$

dove <VALUE> è il valore di induttanza nella riga che chiama il modello e I è la corrente che scorre attraverso l'induttore. Notiamo che per gli induttori è anche possibile includere un modello non lineare di induttore avvolto su nucleo ferromagnetico (CORE) di cui parleremo proprio nel prossimo capitolo.

Capitolo 6

Il trasformatore

6.1 Introduzione

6.2 Il trasformatore ideale

6.3 Accoppiamento mutuo

6.3.1 Accoppiamento perfetto

6.4 Trasformatore reale

6.5 Trasformatore su nucleo ferromagnetico

6.5.1 Il ciclo di isteresi

6.5.2 L'istruzione '.MODEL' per i nuclei ferromagnetici

6.5.3 Comportamento non lineare e analisi armonica

6.5.4 Modello di trasformatore su nucleo ferromagnetico

6.6 I traferri in aria

6.7 Come ricavare i parametri per il modello Jiles - Atherton

Sommario

In questo capitolo studieremo in che modo si possa simulare con Spice un trasformatore, utilizzando sia il modello linearizzato equivalente, sia un possibile modello non lineare.

6.1 Introduzione

In questo capitolo introdurremo i modelli che permettono la simulazione di un trasformatore con Spice. Per una trattazione teorica del trasformatore, rimandiamo agli altri testi di questa collana o, comunque, ad altri volumi che trattano l'argomento.

In questo capitolo, partendo dal modello di trasformatore ideale, considereremo circuiti equivalenti sempre più realistici. Introdurremo i materiali ferromagnetici e mostreremo come si possano simulare con Spice le loro caratteristiche, giungendo così anche ad un modello non lineare del trasformatore descritto per mezzo di un ciclo di isteresi, adoperando il quale è facile mettere in evidenza alcuni fenomeni caratteristici quali, ad esempio, quello della generazione della terza armonica. Ogni modello sarà rappresentato da un sottocircuito che potrà essere facilmente incluso in reti più complesse.

6.2 Il trasformatore ideale

Il trasformatore ideale (Figura 6.1) è un doppio bipolo caratterizzato dalle seguenti relazioni tra le grandezze di ingresso e quelle di uscita

$$\frac{v_1(t)}{v_2(t)} = a \quad \text{e} \quad \frac{i_1(t)}{i_2(t)} = -\frac{1}{a}.$$

Quindi la tensione di ingresso è ‘a’ volte la tensione di uscita (a è detto rapporto di trasformazione), mentre la corrente di uscita è ‘a’ volte la corrente di ingresso cambiata di segno.

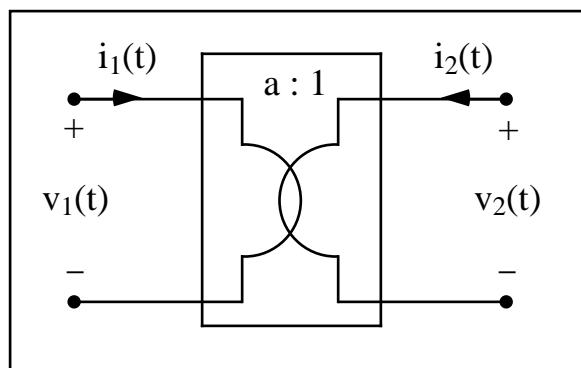


Figura 6.1: simbolo e convenzioni alle due porte di un trasformatore ideale

Ciò vuol dire che il trasformatore ideale è ‘trasparente alle potenze’, cioè

$$v_1(t) i_1(t) + v_2(t) i_2(t) = 0 ,$$

e la potenza istantanea assorbita all'ingresso coincide con quella erogata in uscita. L'altra proprietà di cui gode il trasformatore è la trasformazione di impedenze: se la porta secondaria è chiusa su una impedenza \bar{Z} , immaginando di essere in regime sinusoidale, si ha che

$$\bar{V}_1 = a \bar{V}_2 = - a \dot{Z} \bar{I}_2 = - a \dot{Z} (- a \bar{I}_1) = a^2 \dot{Z} \bar{I}_1 ,$$

per cui la porta primaria vedrà un'impedenza che è a^2 volte quella della secondaria.

Le equazioni che definiscono un trasformatore ideale, in una condizione di funzionamento qualsiasi, possono anche essere interpretate secondo il circuito equivalente di Figura 6.2, in cui abbiamo posto $v_S(t) = a v_2(t)$ e $j_s(t) = a i_1(t)$.

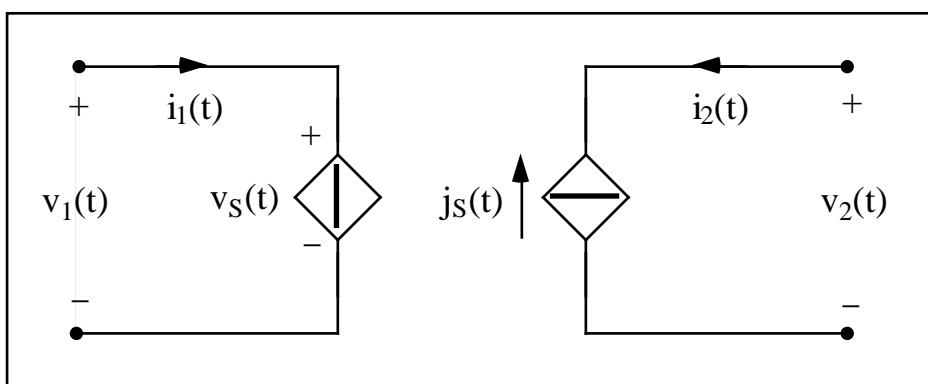


Figura 6.2: circuito equivalente di un trasformatore ideale

Il circuito di Figura 6.2 si può simulare secondo il seguente sottocircuito, in cui abbiamo scelto ' $a = 8$ '.

```
.SUBCKT TRAIDE 1 10 2 20
*a=8
V1 1 5 DC 0
ES 5 10 2 20 8
FS 20 2 V1 8
.ENDS TRAIDE
```

Notate che abbiamo dovuto inserire un generatore indipendente di tensione fittizio usato come amperometro per misurare la corrente al primario.

Vale la pena notare esplicitamente che il modello di trasformatore ideale e, di conseguenza, anche la sua realizzazione con Spice, soddisfa la proprietà di trasformazione delle impedenze e la trasparenza alle potenze per ogni regime di funzionamento, compresi i transitori e persino il regime continuo. Sappiamo bene che ciò non è vero per i trasformatori reali e per i suoi modelli più avanzati e, come vedremo in seguito, anche i modelli più realistici di Spice non verificheranno queste proprietà ‘ideali’.

Concludiamo il paragrafo con un esempio di utilizzo del sottocircuito TRAIDE, considerando il circuito mostrato in Figura 6.3.

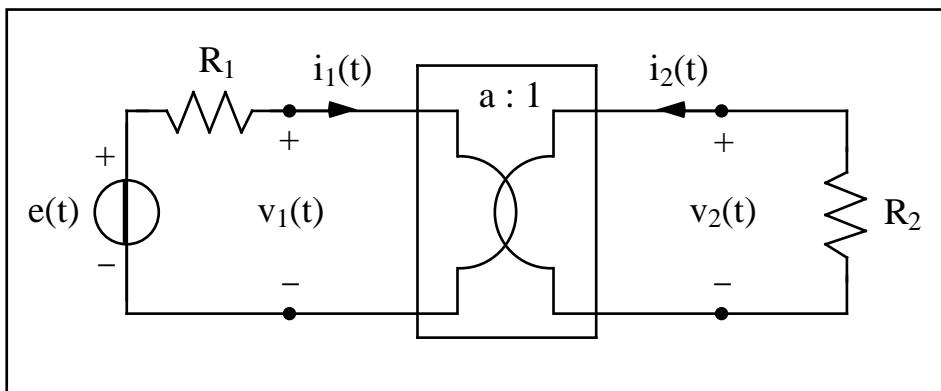


Figura 6.3: semplice circuito con un trasformatore ideale.

Esempio 6.1

```
* Il trasformatore ideale (a = 8)
.SUBCKT TRAIDE 1 10 2 20
V1 1 5 DC 0
ES 5 10 2 20 8
FS 20 2 V1 8
.ENDS TRAIDE
*
VE 1 0 PWL(0 0 1m 10 6m 10 7m 0)
R1 1 2 1u
XT1 2 0 3 0 TRAIDE
R2 3 0 5
.TRAN 0.2m 10m
.PROBE
.END
```

Il file Spice comincia con la definizione del sottocircuito, poi seguono le istruzioni per il generatore di ingresso variabile nel tempo e la sua resistenza interna; ancora una volta notiamo come l'utilizzo del sottocircuito rende molto più leggibile il file

dato che il trasformatore TRAIDE è rappresentato da una ‘scatola nera’ di nome XT1 che si interfaccia con gli altri elementi del circuito mediante i nodi 2 (ingresso), 3 (uscita) e 0 (riferimento comune). La resistenza R1 ha un valore bassissimo rispetto alle altre grandezze del circuito e non comporta una caduta di tensione apprezzabile. Effettuiamo un’analisi in transitorio proprio per mostrare che le tensioni di ingresso e di uscita del trasformatore ideale sono identiche nella forma e proporzionali tramite il fattore ‘a’. In Figura 6.4 notiamo che la tensione di uscita $v(3)$ è sempre proporzionale alla tensione $v(2)$; infatti, nel tratto costante la tensione di uscita è $v(3) = 1.25 \text{ V} = v(2)/8$ dato che nel nostro esempio abbiamo scelto $a = 8$.

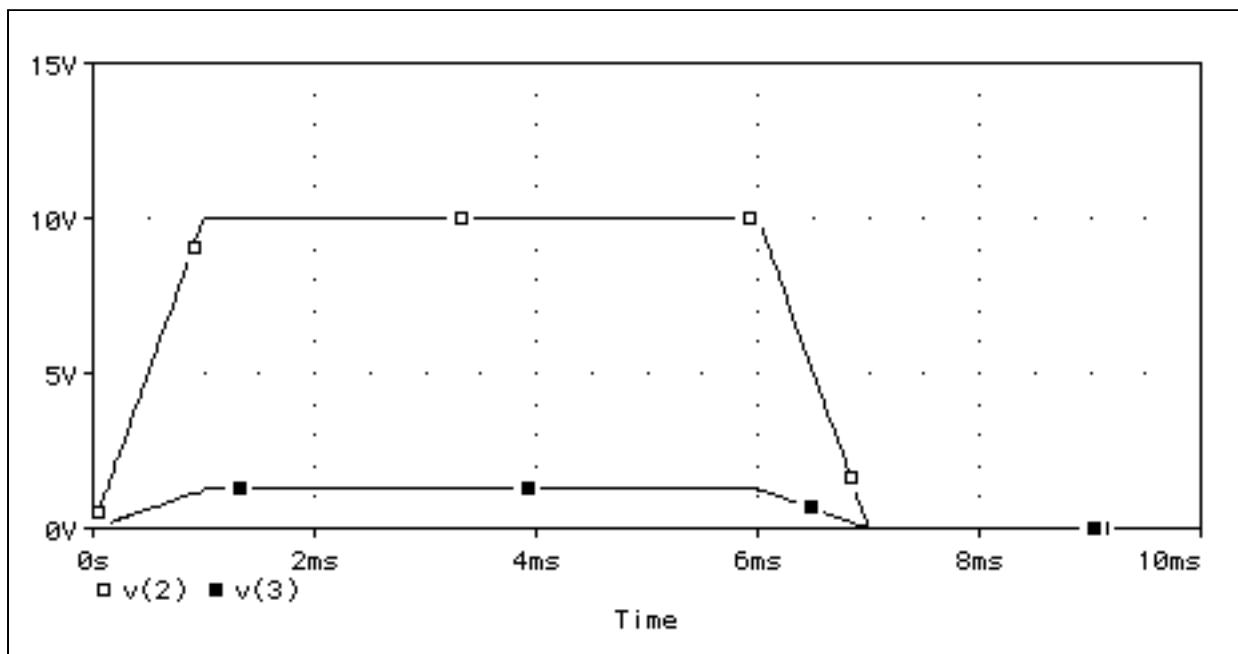


Figura 6.4: dinamiche delle tensioni di ingresso e uscita ($a = 8$).

Provate a fare le stesse considerazioni sulle correnti della porta primaria e della porta secondaria.

6.3 Accoppiamento mutuo

Il passo successivo verso la simulazione ‘realistica’ del trasformatore è considerare, in Spice, la presenza di induttori lineari accoppiati (Figura 6.5). La sintassi per la descrizione degli induttori non cambia, infatti due induttori di nome L1 e L2 connessi tra i nodi 3 e 2, e 5 e 2, rispettivamente, saranno descritti da

L1	3	2	10m
L2	5	2	7m

e la novità, ovvero l'accoppiamento mutuo, sarà descritta da una linea di programma del tipo

```
K12      L1      L2      0.75 .
```

Il nome dell'istruzione che definisce l'accoppiamento tra due induttori deve cominciare con la lettera K, poi seguono i nomi dei due induttori e infine il coefficiente di accoppiamento k che ricordiamo essere definito dalla relazione

$$-1 \quad k = \frac{M}{\sqrt{L_1 L_2}} \quad 1 ,$$

dove M è il coefficiente di mutua induzione (che può essere sia positivo che negativo). Nel caso in cui $|k| = 1$ diciamo che i due induttori sono in condizioni di accoppiamento perfetto. Ad esempio, due induttori accoppiati con gli avvolgimenti che realizzano una mutua induttanza negativa sono rappresentati da un coefficiente k negativo, ovvero

```
K12      L1      L2      -0.75 .
```

Nell'utilizzo di induttori accoppiati, ricordate sempre che ogni nodo in un circuito Spice deve presentare un percorso in continua verso il nodo di riferimento 0. Le soluzioni possibili per collegare elettricamente tra loro il primario e il secondario del trasformatore e per ovviare a tale problema sono due e sono illustrate in Figura 6.5.

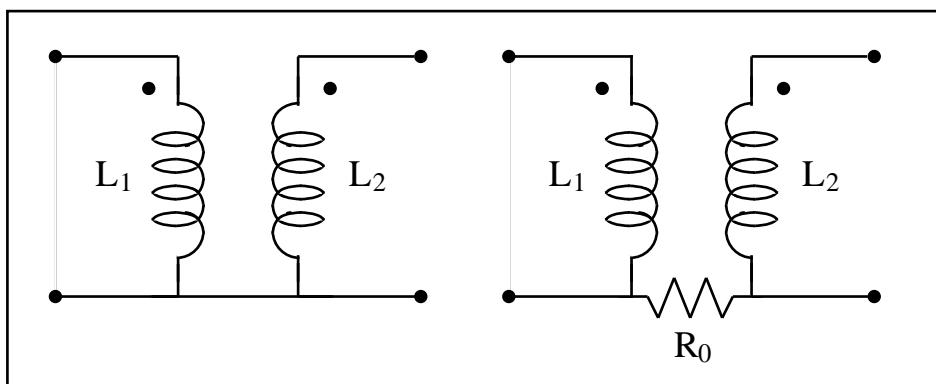


Figura 6.5: soluzioni per il percorso in continua negli induttori accoppiati

Senza questi accorgimenti potrebbe capitare, ad esempio, che il nodo 0 appartenga alla parte di circuito cui è connesso il primario e perciò il secondario e la restante

parte di rete ad esso connessa sarebbero isolati dal nodo 0 e ciò provocherebbe un errore.

Potete collegare i due induttori mediante un cortocircuito oppure mediante un resistore di valore molto elevato rispetto a quelli impiegati nel circuito.

Entrambe le scelte, nel limite in cui R_0 assume un valore molto grande, non alterano l'equilibrio di tensioni e correnti nel circuito per cui sono adatte a risolvere il problema.

In realtà, siccome R_0 avrà sempre un valore non rigorosamente infinito, è consigliabile usare la soluzione con il cortocircuito.

Tornando all'istruzione per la simulazione di induttori accoppiati, essa è molto generale e permette anche di definire anche accoppiamenti multipli. Ad esempio, considerando il caso di uno stesso induttore che si accoppia magneticamente con più induttori, è possibile scrivere le istruzioni che seguono

```
L1    3    2    10m  
L2    4    2    7m  
L3    5    2    5m  
K12    L1    L2    0.75  
K13    L1    L3    0.9
```

e questa situazione potrebbe rappresentare un trasformatore che ha un avvolgimento primario e due avvolgimenti secondari.

In realtà, è possibile anche rappresentare accoppiamenti multipli con una sola riga di programma

```
K123    L1    L2    L3    0.9
```

ma, in tal caso, il valore del coefficiente di accoppiamento sarà lo stesso per tutti gli induttori.

6.3.1 Accoppiamento perfetto

Consideriamo ora un esempio di circuito in cui utilizziamo induttori in condizione di accoppiamento perfetto, operanti in regime sinusoidale.

Esempio 6.2

* Accoppiamento perfetto

```
V1    1    0    AC    10    0
R1    1    2    1u
L1    2    0    64m
L2    3    0    1m
K12   L1   L2   1
R2    3    0    5
.AC   LIN   1    159.15    159.15
.PRINT  AC    VM(2)      VP(2)      VM(3)      VP(3)
.PRINT  AC    IM(R1)     IP(R1)     IM(R2)     IP(R2)
.END
```

Effettuiamo una analisi ad una sola frequenza e notiamo che il rapporto di trasformazione $a = 8$, dato dalla formula

$$a = \frac{L_1}{M} = \frac{L_1}{k \sqrt{L_1 L_2}} ,$$

è rispettato per le tensioni, ma non per le correnti, come dimostrano le due uscite che seguono.

FREQ	VM(2)	VP(2)	VM(3)	VP(3)
1.592E+02	1.000E+01	8.953E-07	1.250E+00	8.953E-07
FREQ	IM(R1)	IP(R1)	IM(R2)	IP(R2)
1.592E+02	1.593E-01	-7.869E+01	2.500E-01	8.953E-07

Questo risultato non ci meraviglia in quanto sappiamo che l'accoppiamento mutuo tra due induttori L_1 e L_2 equivale a un trasformatore ideale con in parallelo al primario un'induttanza uguale a L_1 , e da ciò discende che il rapporto di trasformazione è soddisfatto solo per le tensioni e non per le correnti, in quanto l'induttanza L_1 , in parallelo alla porta primaria, deriverà una parte della corrente di ingresso.

Per convincervi fino in fondo di quanto detto, consideriamo il seguente esempio con il circuito equivalente TRAIDE più un induttore L_1 in parallelo all'ingresso (per simulare un accoppiamento in condizioni di accoppiamento perfetto).

Esempio 6.3

* Accoppiamento mutuo perfetto: circuito equivalente

.SUBCKT TRAIDE 1 10 2 20

* a=8

V1 1 5 DC 0
ES 5 10 2 20 8
FS 20 2 V1 8

.ENDS TRAIDE

*

V1 1 0 AC 10 0
R1 1 2 1u
L1 2 0 64m
RS 2 20 1u
XT1 20 0 3 0 TRAIDE
R2 3 0 5
.AC LIN 1 159.15 159.15
.PRINT AC VM(2) VP(2) VM(3) VP(3)
.PRINT AC IM(R1) IP(R1) IM(R2) IP(R2)

.END

Vale la pena notare che la resistenza serie RS, peraltro di valore molto piccola, è necessaria ad evitare che si formi una maglia costituita dall'induttore L1 e il generatore fittizio V1, cosa che procurerebbe un errore segnalato dal compilatore. In uscita esso fornisce lo stesso risultato del caso precedente.

FREQ	VM(2)	VP(2)	VM(3)	VP(3)
1.592E+02	1.000E+01	8.953E-07	1.250E+00	8.953E-07
FREQ	IM(R1)	IP(R1)	IM(R2)	IP(R2)
1.592E+02	1.593E-01	-7.869E+01	2.500E-01	8.953E-07

Per concludere e riassumere possiamo dire che in Spice il doppio bipolo accoppiamento mutuo funzionante in condizioni di accoppiamento perfetto può essere rappresentato anche da un circuito costituito da un trasformatore ideale con l'induttanza L1 in parallelo alla porta primaria.

6.4 Trasformatore reale

Mediante un opportuno circuito equivalente comprendente induttori, resistori e un trasformatore ideale, sappiamo che possiamo rappresentare tutti i fenomeni che

avvengono nel funzionamento di un trasformatore reale avvolto su nucleo ferromagnetico, almeno nei limiti in cui questo doppio bipolo possa ritenersi funzionare in condizioni di linearità.

Partendo dall'ipotesi di trasformatore su ferro ideale, non è difficile arrivare al circuito equivalente mostrato in Figura 6.6 dove sono indicati anche i nomi per i fasori che rappresentano le diverse grandezze in gioco.

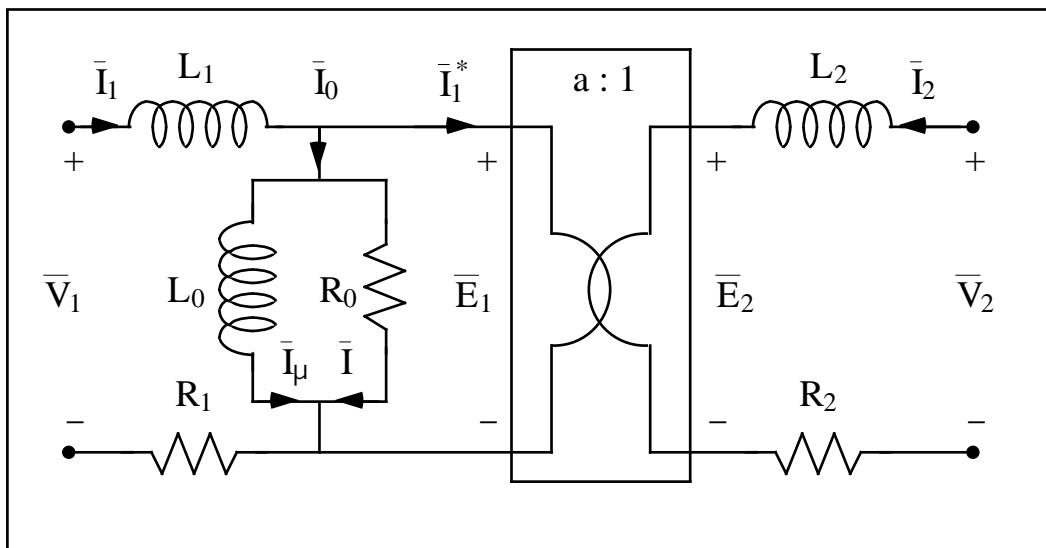


Figura 6.6: circuito equivalente completo per il trasformatore reale.

Le induttanze L_1 e L_2 sono dovute ai flussi dispersi per cui sono frazione delle induttanze primarie e secondarie, l'induttanza L_0 è dovuta alla riluttanza finita del circuito magnetico e, indicando con N_1 il numero di spire dell'avvolgimento al primario, vale:

$$L_0 = \frac{N_1^2}{\mu_0 \mu_r S},$$

in cui si è posto

$$= \frac{1}{\mu_0 \mu_r} \frac{L}{S}.$$

Le perdite nel ferro per isteresi e per correnti parassite, sintetizzate nel resistore R_0 , dipendono dal particolare materiale ferromagnetico. I resistori R_1 e R_2 rappresentano, invece, le perdite nel rame degli avvolgimenti e il loro valore è dato dalle relazioni

$$R_{1/2} = 58.6 N_{1/2} \frac{L_{avvolgimento}}{S_{avvolgimento}} () ,$$

in cui la lunghezza del percorso dell'avvolgimento è misurata in metri e la sezione del filo in millimetri quadri. Infine, il trasformatore ideale è caratterizzato dal solo rapporto di trasformazione $a = N_1/N_2$.

Di seguito riportiamo il sottocircuito equivalente che codifica il circuito equivalente appena discusso.

```
.SUBCKT TRAREA 1 10 2 20 PARAMS:  
+ N1=100, N2= 1000, R0=10k, R1=0.4, R2=0.8, RIL=0.995m  
+ P1=0.05, N12=0.1, L1D=0.5m , L2D=50m , L0D=10m  
L1 1 3 {L1D}  
L2 2 4 {L2D}  
R0 3 8 10u  
L0 8 5 {L0D}  
Rp0 3 5 {R0}  
Rp1 5 10 {R1}  
Rp2 6 20 {R2}  
V1 3 7 DC 0  
ES 7 5 4 6 0.1  
FS 6 4 V1 0.1  
.ENDS TRAREA
```

Notate l'uso dell'opzione 'PARAMS' per specificare tutti i parametri che definiscono il particolare modello di trasformatore reale che abbiamo considerato. Il rapporto di trasformazione è 10:1 e abbiamo supposto un flusso disperso del 5%. La resistenza R0, posta in serie all'induttanza che rappresenta la riluttanza finita del circuito magnetico, serve ad evitare la maglia chiusa tra questa induttanza e il generatore controllato incluso nel circuito che rappresenta il trasformatore ideale.

Questo sottocircuito può essere studiato mediante un'analisi '.AC' ad una sola frequenza in quanto gli effetti schematizzati mediante resistenze e induttanze dipendono, in maniera più o meno marcata, dalla frequenza; per questo motivo, i parametri forniti sono validi ad una sola frequenza e vanno aggiornati quando si usano altri valori di frequenza.

Esempio 6.4

```
* Trasformatore reale
.INC    'TRAREA.CIR'
V1      1      0      AC      1
R1      1      2      1m
XT1    2      0      3      0      TRAREA
R2      3      0      5
.AC     LIN      1      159.15      159.15
.PRINT   AC      VM(2)      VP(2)      VM(3)      VP(3)
.PRINT   AC      IM(R1)      IP(R1)      IM(R2)      IP(R2)
.END
```

Se il file TRAREA.CIR contiene il sottocircuito TRAREA, allora questo segmento di programma fornisce in uscita i valori di seguito riportati.

FREQ	VM(2)	VP(2)	VM(3)	VP(3)
1.592E+02	9.996E-01	4.828E-02	4.423E-01	-6.477E+01
<hr/>				
FREQ	IM(R1)	IP(R1)	IM(R2)	IP(R2)
1.592E+02	9.288E-01	-6.508E+01	8.845E-02	-6.477E+01

6.5 Trasformatore su nucleo ferromagnetico

Finalmente possiamo parlare del modello di trasformatore più complesso simulabile con Spice: quello comprendente un nucleo ferromagnetico e, quindi, caratterizzato da effetti di saturazione, perdite nel ferro per isteresi e fenomeni non lineari.

Cominceremo introducendo i nuclei ferromagnetici su cui è possibile ‘avvolgere’ anche induttori e poi ne illustreremo i parametri del modello e, infine, daremo il circuito equivalente del trasformatore su nucleo ferromagnetico.

6.5.1 Il ciclo di isteresi

L’aspetto più evidente del comportamento non lineare dei materiali ferromagnetici è che essi presentano il fenomeno dell’isteresi. Sottoposti all’azione di un campo di induzione magnetica essi si magnetizzano e, anche quando il campo esterno si annulla, conservano una magnetizzazione residua, conservando, in qualche maniera, ‘memoria’ di quel che è accaduto. Inoltre, se il campo di induzione magnetica viene ridotto a zero, cambiato di segno e poi riportato a zero e poi di

nuovo al valore precedente, la magnetizzazione del materiale non procede sugli stessi valori ma su valori differenti. Rappresentando questo fenomeno su di un grafico B - H, otteniamo un ciclo di isteresi.

La simulazione di queste caratteristiche è realizzata in Spice mediante il modello di Jiles - Atherton per i materiali ferromagnetici di cui daremo qualche cenno nell'ultimo paragrafo di questo capitolo.

Cominciamo subito con la visualizzazione del ciclo di isteresi introducendo un esempio tratto dalla libreria di componenti ‘EVAL.LIB’ fornita con le versioni ‘evaluation’ di Spice. Lo scopo è quello di mostrare la caratteristica B - H e il ciclo di isteresi corrispondente di un particolare nucleo ferromagnetico su cui si avvolge un induttore.

Esempio 6.5

```
* Nucleo ferromagnetico
.MODEL K528T500_3C8 CORE(LEVEL=2 ALPHA=0 MS=415.2k
+ A=44.82 C=.4112 K=25.74 AREA=1.17 PATH=8.49)
RL    1    0    1
L1    1    0    20
K1    L1    0.9999    K528T500_3C8
IGEN    0    1    SIN(0 0.8 0.25Hz 0)
.TRAN    0.1    5
.OPTIONS    itl5=0
.PROBE
.END
```

La riga di programma ‘.MODEL’ definisce le caratteristiche del nucleo ferromagnetico di nome ‘K528T500_3C8’, che nel prossimo paragrafo spiegheremo.

Una delle principali novità è che quando si fa uso di un nucleo ferromagnetico il valore nella riga dell’induttore, avvolto su tale nucleo, non specifica più il valore dell’induttanza in henry, ma piuttosto il numero di spire che costituiscono l’avvolgimento sul nucleo ferromagnetico. Nell’esempio precedente l’induttore L1 è costituito da 20 avvolgimenti.

Inoltre, come si nota dall’esempio 6.5, nella riga per gli induttori accoppiati è possibile specificare anche un solo induttore che sarà inteso avvolto sul nucleo ferromagnetico.

L’induttore è alimentato da un generatore di corrente sinusoidale capace di percorrere un intero ciclo di isteresi come si può notare dalla Figura 6.7.

Con Probe è possibile visualizzare il ciclo di isteresi, mostrando $B(K1)$ come traccia sull'asse delle ordinate (l'unità di misura sarà il gauss) e disponendo $H(K1)$ come variabile sull'asse delle ascisse (l'unità di misura sarà l'oersted). Notate che queste informazioni sono disponibili solo mediante Probe e non mediante le istruzioni '.PRINT' e '.PLOT'.

Infine, abbiamo utilizzato l'istruzione '.OPTIONS' per regolare il valore di un'opzione in transitorio; nell'ultimo capitolo troverete un paragrafo che spiega questa ed altre opzioni.

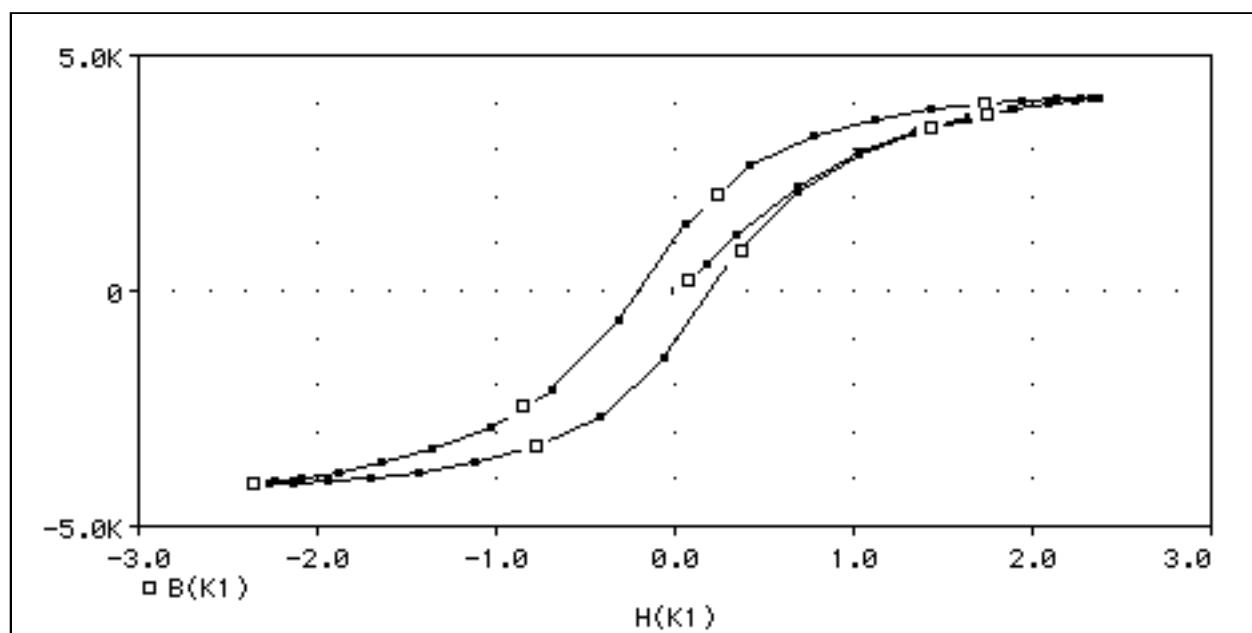


Figura 6.7: esempio di ciclo di isteresi.

Quindi come si può osservare dalla figura, a seguito dell'andamento sinusoidale della corrente, il campo magnetico H cambia segno e il campo B raggiunge un massimo sulla cosiddetta ‘curva di prima magnetizzazione’ e poi segue la curva chiusa caratteristica, detta ciclo di isteresi.

6.5.2 L'istruzione '.MODEL' per i nuclei ferromagnetici

Passiamo ora a spiegare i parametri che intervengono nell'istruzione '.MODEL' per un nucleo ferromagnetico. Innanzitutto, la riga è del tipo

```
.MODEL      NAME      CORE  [(PARAMETRI)]
```

e dopo '.MODEL' segue il nome (che deve cominciare con una lettera) che vogliamo assegnare a questo particolare modello, poi CORE (nucleo) specifica che si tratta di un nucleo ferromagnetico, infine abbiamo la lista dei parametri. Se la

lista dei parametri è vuota vengono considerati i valori di ‘default’, elencati nella tabella che segue.

Parametro	Descrizione	Unità di misura	Default
LEVEL	Tipo di modello		1
A	Parametro di energia termica	A/m	1000
ALPHA	Coefficiente di accoppiamento tra domini (per LEVEL = 1)		0.001
AREA	Sezione trasversa media per il campo magnetico	cm ²	0.1
C	Flessibilità dei domini	adimensionale	0.2
GAMMA	Parametro di attenuazione dei domini (solo per LEVEL=1)	s ⁻¹	Infinito
GAP	Lunghezza totale dei traferri	cm	0
K	Coefficiente di anisotropia dei domini	A/m	500
MS	Magnetizzazione di saturazione	A/m	1.000.000
PACK	Coefficiente di stipamento	adimensionale	1.0
PATH	Lunghezza media del percorso per le linee di flusso magnetico	cm	1.0

Esaminiamoli in qualche dettaglio.

LEVEL stabilisce il tipo di modello da adottare (1 oppure 2): alcuni parametri non sono disponibile nel modello 2.

AREA è un parametro che rappresenta la media della sezione trasversa del nucleo ferromagnetico. Un toroide a sezione costante avrà come AREA proprio la misura dell’area trasversa.

GAP è la lunghezza totale degli eventuali traferri in aria necessari a volte per linearizzare il comportamento dell’intero circuito magnetico.

K è il coefficiente di anisotropia dei domini e in pratica stabilisce il livello di isteresi: più grande è K, più largo è il ciclo di isteresi.

MS è la magnetizzazione di saturazione e, come ricorda il nome stesso, stabilisce il valore di magnetizzazione oltre il quale il nucleo ferromagnetico satura, assumendo un comportamento simile al vuoto.

PACK è il coefficiente di stipamento e tiene conto del fatto che i lamierini sono separati l’un l’altro da uno strato di carta o di vernice che aumenta lo spessore complessivo.

Ovviamente, il modello di Jiles - Atherton non è in grado di simulare tutti gli effetti di un trasformatore reale, ma sicuramente è in grado di simulare, l'effetto di prima magnetizzazione, la saturazione, la coercitività, l'effetto di magnetizzazione residua e le perdite per isteresi.

Infine, tornando alla riga che specifica l'accoppiamento magnetico, se in essa è presente un modello tipo CORE, dobbiamo tener conto che:

- l'accoppiamento mutuo tra induttori diventa non lineare;
- gli induttori elencati in tale riga diventano ‘avvolgimenti’ e al valore dell’induttanza nella riga che li definisce deve essere sostituito un numero intero che specifica il numero di giri dell'avvolgimento;
- nella riga che specifica l'accoppiamento può esserci anche solo un induttore.

6.5.3 Comportamento non lineare e analisi armonica

Nel seguito, consideriamo un esempio che mostra l'effetto non lineare che subisce la corrente di un induttore se lo si avvolge su un nucleo ferromagnetico.

Esempio 6.6

* Effetto non lineare

```
.MODEL K528T500_3C8 CORE(LEVEL=2 ALPHA=0 MS=415.2k
+ A=44.82 C=.4112 K=25.74 AREA=1.17 PATH=8.49)
*
V1    1    0    SIN(0 1m 50 0)
RL    1    2    1
L1    2    0    20
K1    L1    0.9999    K528T500_3C8
.TRAN    2m    60m
.PROBE
.END
```

Il listato Spice precedente simula un semplice circuito RL, in questo caso il generatore sinusoidale, con frequenza 50 Hz, ha ampiezza massima uguale a 1 mV e produce la corrente visibile in Figura 6.8.

Essendo l'ampiezza del segnale contenuta, non sono presenti grossi effetti non lineari e la corrente conserva l'andamento sinusoidale del generatore di ingresso. Invece considerando un generatore con ampiezza massima 10 V.

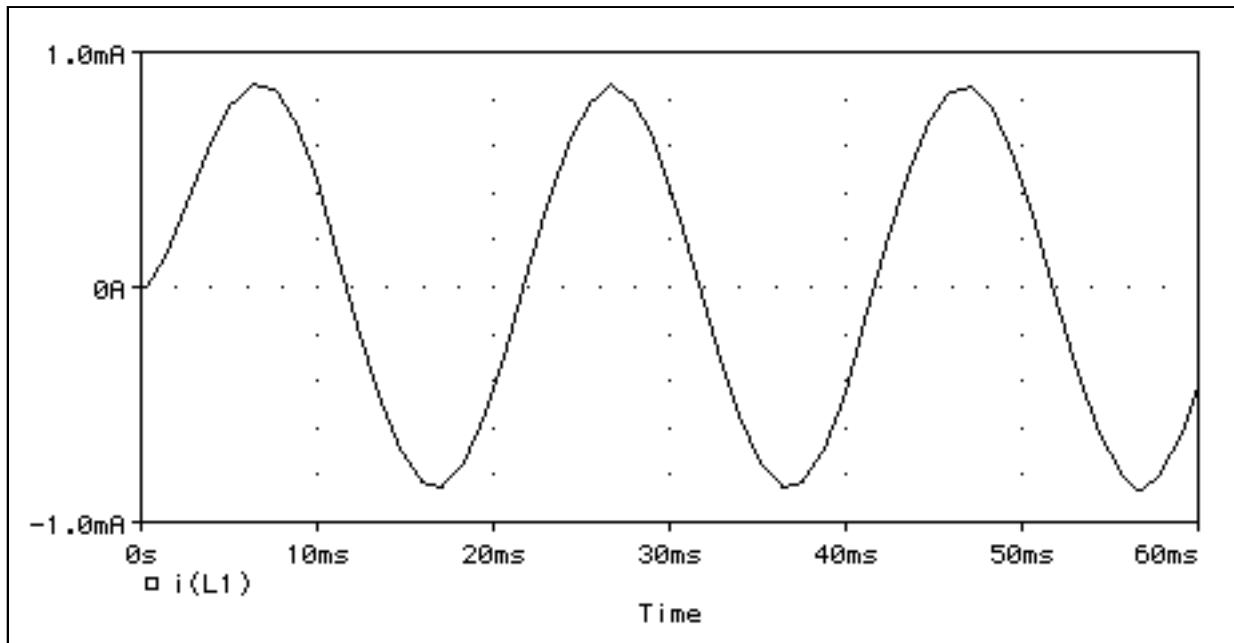


Figura 6.8: corrente in un induttore dal comportamento lineare.

Come mostrato in Figura 6.9, l'effetto non lineare dovuto alla saturazione del nucleo è ben presente e consiste nella cosiddetta ‘generazione della terza armonica’. Questo effetto non lineare può essere studiato anche mediante l’ausilio dell’istruzione ‘.FOUR’, dove FOUR sta per Fourier: il nostro simulatore effettua persino un’analisi armonica!

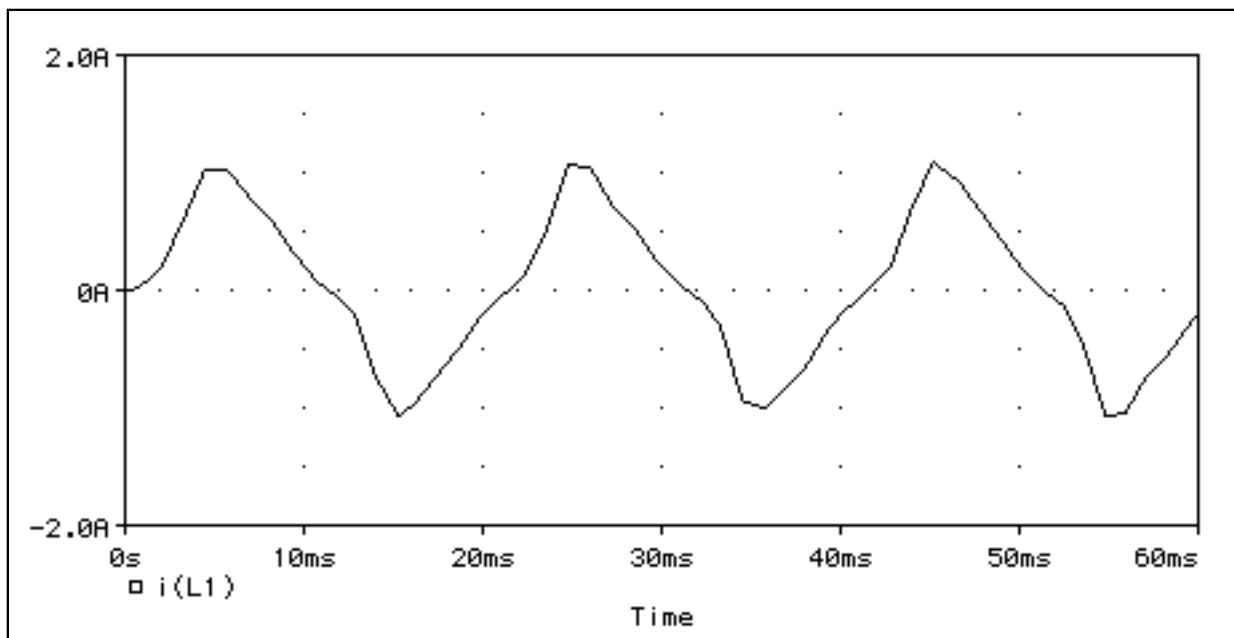


Figura 6.9: generazione della terza armonica in un induttore non lineare.

A partire da un'analisi nel dominio del tempo, Spice può calcolare le prime nove armoniche spettrali di uno o più segnali del circuito se è presente nel listato una istruzione del tipo

```
.FOUR      FREQ    VAR1    VAR2    ... ,
```

in cui FREQ è la frequenza fondamentale e VAR1 e VAR2 sono le grandezze del circuito di cui si vogliono calcolare le prime nove armoniche. Come già abbiamo detto, questa istruzione deve essere sempre associata all'analisi nel tempo '.TRAN'. Poi, riguardo la precisione del calcolo, bisogna aggiungere che Spice effettua il calcolo sull'ultimo periodo a disposizione nell'analisi in transitorio ovvero nell'intervallo

$$\left[\frac{\text{TSTOP} - 1}{\text{FREQ}}, \text{TSTOP} \right]$$

per cui è necessario scegliere per l'analisi un tempo TSTOP sufficientemente lungo da far esaurire il 'transitorio' in modo da poter esaminare una situazione di regime. Evidentemente l'idea è di eccitare il circuito sotto esame con un generatore sinusoidale fissandone la frequenza e gli altri parametri; poi nella riga '.FOUR' la frequenza fondamentale sarà la stessa del generatore e sulle variabili richieste saranno calcolate le armoniche generate dalle non linearità presenti nel circuito.

Tornando al nostro esempio di induttore non lineare con nucleo ferromagnetico, inseriamo la linea di programma per l'analisi secondo Fourier come di seguito riportato.

Esempio 6.7

```
* Effetto non lineare: analisi di Fourier
.MODEL K528T500_3C8 CORE(LEVEL=2 MS=415.2k A=44.82
+ C=.4112 K=25.74 AREA=1.17 PATH=8.49)
V1    1    0    SIN(0 10 50 0)
RL    1    2    1
L1    2    0    20
K1    L1    0.9999    K528T500_3C8
.TRAN    1m    120m
.FOUR    50    I(L1)
.PROBE
.END
```

Noteate che abbiamo raddoppiato il tempo di analisi (TSTOP = 120 ms) e che abbiamo richiesto l'analisi di Fourier della corrente che scorre nell'induttore. Il risultato dell'analisi Spice è mostrato nel riquadro che segue.

FOURIER COMPONENTS OF TRANSIENT RESPONSE I(L1)					
DC COMPONENT = -6.154045E-03					
NO	(HZ)	HARMONIC FREQUENCY FOURIER NORMALIZED PHASE	NORMALIZED COMPONENT	COMPONENT (DEG)	PHASE (DEG)
1	5.000E+01	9.729E+00	1.000E+00	-1.758E+00	0.000E+00
2	1.000E+02	1.619E-01	1.664E-02	1.743E+02	1.760E+02
3	1.500E+02	5.238E-01	5.384E-02	-1.195E+02	-1.177E+02
4	2.000E+02	3.478E-01	3.575E-02	9.139E+01	9.315E+01
5	2.500E+02	3.386E-01	3.480E-02	-1.645E+02	-1.627E+02
6	3.000E+02	3.299E-01	3.390E-02	1.392E+01	1.567E+01
7	3.500E+02	1.509E-01	1.551E-02	-1.691E+02	-1.673E+02
8	4.000E+02	2.294E-01	2.358E-02	-6.810E+01	-6.634E+01
9	4.500E+02	1.951E-01	2.005E-02	8.558E+01	8.733E+01
TOTAL HARMONIC DISTORTION = 8.951200E+00 PERCENT					

Esaminiamo cosa abbiamo ottenuto nel file ‘.OUT’ di Spice.

La prima colonna stabilisce il numero dell’armonica (1 rappresenta la ‘fondamentale’), la seconda colonna dà la frequenza dell’armonica, la terza e la quinta danno l’ampiezza e la fase della generica armonica, la quarta e la sesta colonna contengono informazioni sull’ampiezza e la fase relative al valore della fondamentale.

Dalla quarta colonna notiamo che la terza armonica ha un’ampiezza maggiore di tutte le altre armoniche compresa la seconda.

6.5.4 Modello di trasformatore su nucleo ferromagnetico

Eccoci finalmente giunti al modello più completo di trasformatore che possiamo utilizzare con Spice.

Nel riquadro seguente possiamo vedere il sottocircuito TRAFER che presenta due induttori avvolti sul nucleo ferromagnetico 528T500_3C8, le due resistenze Rp1 e Rp2 rappresentano le perdite nel rame mentre la resistenza Rp0 tiene conto delle perdite per correnti parassite.

```

.SUBCKT TRAFER      1   10    2   20      PARAMS:
+ N1=100, N2= 1000, R1=0.4, R2=0.8, R0=10k
*
.MODEL K528T500_3C8 CORE(LEVEL=2 ALPHA=0 MS=415.2k A=44.82
+ C=.4112 K=25.74 AREA=1.17 PATH=8.49)
*
Rp1  1   3   {R1}
Rp2  2   4   {R2}
Rp0  3   10  {R0}
L1   3   10  {N1}
L2   4   20  {N2}
K12  L1  L2  0.95      K528T500_3C8
.ENDS  TRAFER

```

Per mezzo di un'analisi '.AC' a decadi a partire da 1 Hz fino a 10 kHz è possibile studiare il comportamento in frequenza del modello TRAFER. Ma in che senso si può fare un'analisi in frequenza di un dispositivo non lineare?

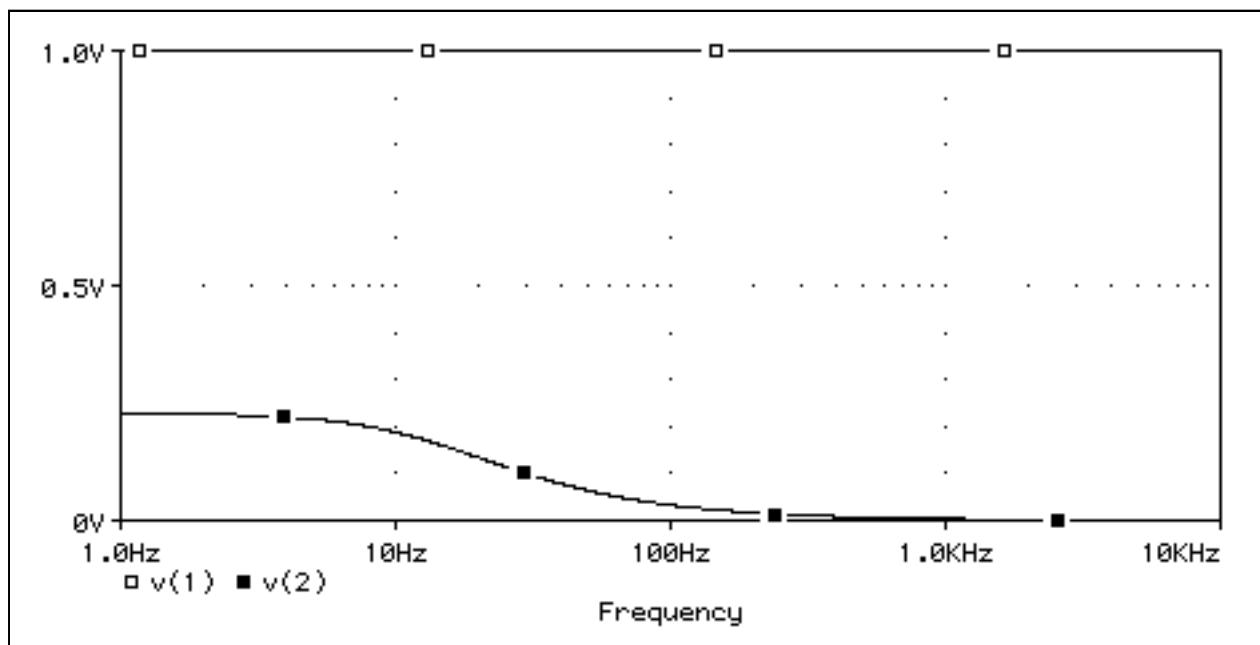


Figura 6.10: tensioni in ingresso e in uscita al modello TRAFER.

Quando abbiamo a che fare con un dispositivo che si comporta in maniera non lineare, a rigore, non potremmo effettuare un'analisi in frequenza. Tuttavia è pratica comune realizzare quella che si chiama analisi per piccoli segnali: si tratta di fissare un punto di funzionamento e di far variare, in maniera sinusoidale, un segnale di piccola ampiezza attorno a questo punto. Se la perturbazione è piccola,

la caratteristica non lineare, a meno di piccoli errori, si potrà approssimare con una retta e, in tal senso, si può effettuare un'analisi in frequenza. Spice effettua l'analisi per piccoli segnali per dispositivi non lineari, come, ad esempio, il trasformatore con isteresi che stiamo considerando. In tal modo, usando il sottocircuito precedente si possono facilmente ottenere le due figure che seguono che rappresentano, rispettivamente, gli andamenti in frequenza delle tensioni di ingresso e di uscita ai capi del trasformatore e il modulo della funzione di trasferimento, intesa come rapporto tra i valori efficaci delle due precedenti tensioni.

Nell'esempio abbiamo supposto di alimentare la porta primaria con un generatore di valore massimo di 1 V e frequenza variabile e di chiudere la porta secondaria su un resistore da 1 .

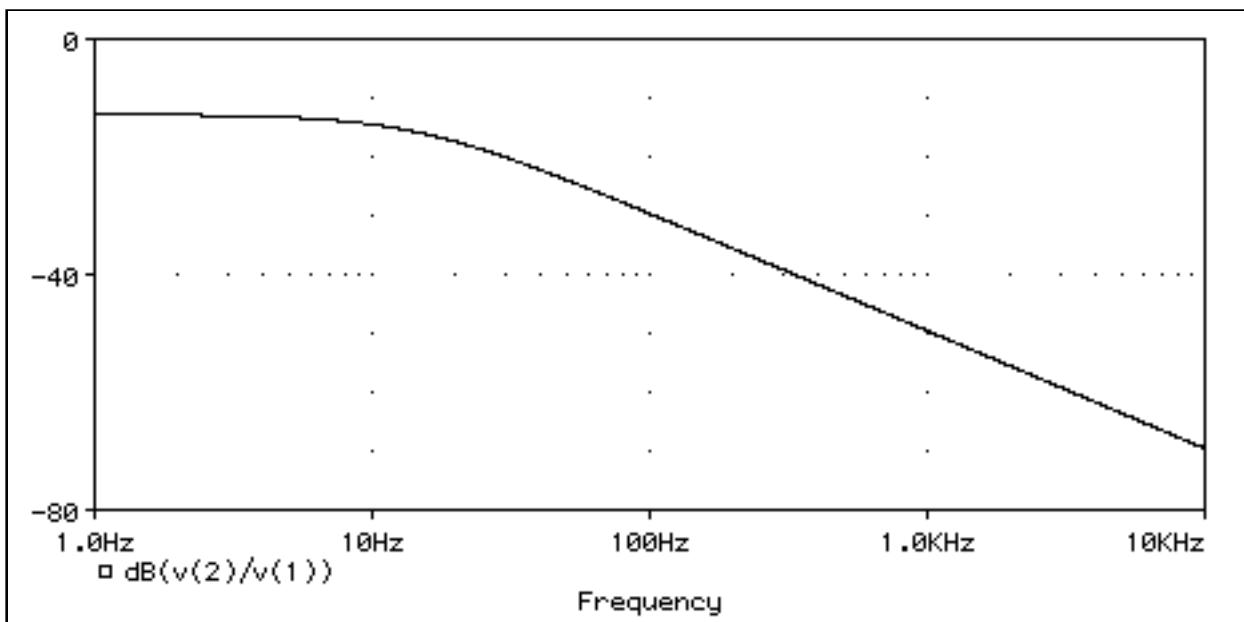


Figura 6.11: analisi in frequenza di un trasformatore su nucleo ferromagnetico.

6.6 I traferri in aria

Nella presentazione dei parametri del modello di nucleo ferromagnetico è possibile includere la lunghezza totale dei traferri in aria presenti nel nucleo (parametro ‘GAP’). Per determinare la soluzione dei campi magnetici con il traferro, Spice realizza una tecnica che parte dai risultati ottenuti senza traferro.

Nel seguito spieghiamo brevemente come Spice realizzi questa procedura per ‘aggiustare’ i valori ottenuti senza traferro.

Se lo spessore del traferro è piccolo rispetto alle dimensioni caratteristiche del nucleo ferromagnetico, possiamo assumere che tutte le linee di flusso magnetico passino indisturbate attraverso il traferro e che i cosiddetti effetti di bordo siano trascurabili, o al massimo, se ne possa tener conto mediante un piccolo aumento della dimensione effettiva del traferro in aria. Integrando il campo magnetico lungo un percorso medio che corre attraverso il nucleo, grazie alla legge di Ampère - Maxwell, possiamo scrivere che

$$H_{\text{CORE}} L_{\text{CORE}} + H_{\text{GAP}} L_{\text{GAP}} = N I ,$$

ovvero scomponiamo l'integrale in due tratti in cui il campo H è praticamente costante e può essere moltiplicato per la lunghezza del percorso. Sappiamo che l'integrale è uguale alla somma delle correnti concatenate dal percorso d'integrazione ovvero è uguale a NI , prodotto tra il numero di spire dell'avvolgimento intorno al nucleo e la corrente che lo attraversa. Le grandezze L_{CORE} e L_{GAP} sono rispettivamente la lunghezza del percorso nel nucleo e lo spessore del traferro; H_{CORE} e H_{GAP} sono, rispettivamente, i campi magnetici nel nucleo e nel traferro. Inoltre sappiamo che nel traferro $B_{\text{GAP}} = \mu_0 H_{\text{GAP}}$ e, siccome le linee di forza dell'induzione magnetica sono normali al traferro, sarà anche $B_{\text{GAP}} = B_{\text{CORE}}$, ed allora

$$H_{\text{CORE}} L_{\text{CORE}} + \mu_0 B_{\text{GAP}} L_{\text{GAP}} = N I .$$

Questa è una equazione abbastanza complessa da risolvere per il modello Jiles - Atherton che, come discuteremo nel prossimo paragrafo, è il modello che rappresenta i nuclei ferromagnetici in Spice. Infatti quest'ultimo è un modello di equazione di stato piuttosto che una funzione esplicita. Per questo si preferisce usare una tecnica grafica secondo cui, assegnata la forza magneto-motrice NI , ricaviamo B_{CORE} e H_{CORE} . Essa consiste nei seguenti passi:

- considerare la curva $B - H$ senza traferro;
- tracciare una retta che parte dal valore NI con una pendenza di $-L_{\text{CORE}}/L_{\text{GAP}}$;
- trovare l'intersezione della retta con la curva $B - H$.

Il punto di intersezione fornisce i valori B_{CORE} e H_{CORE} relativi a NI per il nucleo comprendente il traferro. Il valore NI è, quindi, il valore apparente o esterno di H_{CORE} : in realtà il valore di H_{CORE} è più piccolo.

Spice realizza l'equivalente numerico del procedimento grafico appena descritto.

6.7 Come ricavare i parametri per il modello Jiles - Atherton

In questo paragrafo mostreremo come ottenere i parametri da inserire nel modello CORE a partire dai dati (data sheets) e dal ciclo di isteresi forniti con il materiale ferromagnetico che vogliamo studiare e facendo uso principalmente del grafico del ciclo di isteresi mediante Probe.

Cominciamo tracciando la curva che non presenta isteresi per $K = 0$. Questa curva è centrata nel ciclo $B - H$ come ‘la spina dorsale’ di un corpo umano. La pendenza per $H = 0$ di tale curva è all’incirca la stessa di quella di quando la curva passa il punto in cui $B = 0$ (asse delle ascisse). Così possiamo fissare la magnetizzazione di saturazione

$$MS = \frac{B_{max}}{0.01257},$$

e in tal caso risulta pure

$$\text{pendenza} = \frac{1}{3A/MS - \text{ALPHA}}.$$

Possiamo cominciare con $\text{ALPHA} = 0$ e poi ‘giocare’ con i valori per ottenere il ciclo che desideriamo. Ovviamente alla fine MS dovrà essere uguale al valore di saturazione che desideriamo.

Poi, poniamo K diverso da zero per considerare l’isteresi. In effetti K ha effetto sull’apertura del ciclo. Dopo aver stabilito la larghezza e forma del ciclo, possiamo muovere il ciclo fino ad ottenere i valori desiderati per l’induzione residua B_R e il campo coercitivo H_C .

Poi, bisogna stabilire C per fissare la permeabilità iniziale. In effetti, C è il rapporto della suscettibilità della curva iniziale e quella anisteretica. Dal momento che la permeabilità è dB/dH , Probe calcolerà questa formula (fate attenzione al fatto che in realtà Spice non calcola la derivata ma il limite del rapporto incrementale!). Quindi notiamo che il valore iniziale della permeabilità è uguale proprio a dB/dH .

Poi, possiamo stabilire $GAMMA$ per dare la giusta dipendenza dalla catena. Si può provare a partire con la corner frequency. Aumentare $GAMMA$ significa ridurre le perdite su quella frequenza. I restanti valori del modello sono geometrici e possono essere ricavati direttamente dalle caratteristiche geometriche del materiale ferromagnetico.

Vale la pena, alla fine di questo paragrafo, segnalare che, per una breve introduzione al modello Jiles - Atherton su cui si basa la simulazione dei nuclei ferromagnetici in Spice, rimandiamo agli articoli presenti nella letteratura scientifica tra cui segnaliamo sicuramente l'articolo degli autori D. C. Jiles e D. L. Atherton dal titolo ‘Theory of ferromagnetic hysteresis’ e pubblicato sulla rivista scientifica ‘Journal of Magnetism and Magnetic Materials’ nei numeri 61 e 48 dell’anno 1986.

Capitolo 7

Linee di trasmissione

7.1 Introduzione

7.2 Richiami sulle linee di trasmissione

7.3 Linee di trasmissione ideali

7.4 Linee di trasmissione con perdite

7.5 L'effetto pelle

7.6 Linee accoppiate

7.7 I parametri delle linee

Sommario

In questo capitolo esamineremo la sintassi per simulare le linee di trasmissione con e senza perdite. Un cenno sarà fatto anche alle linee con effetto pelle e alle linee accoppiate.

7.1 Introduzione

In questo capitolo introdurremo le linee di trasmissione che sono elementi molto importanti per modellare le reti elettriche ed elettroniche in cui le dimensioni caratteristiche dei circuiti sono confrontabili con le lunghezze d'onda dei segnali elettrici che le attraversano, ovvero per descrivere compiutamente tutti quei casi in cui è fondamentale considerare la ‘propagazione’ dei segnali nei circuiti. Ciò vuol dire che, sia che si tratti di una pista in rame sulla piastra madre di un moderno calcolatore, sia che si consideri una linea elettrica ad alta tensione che unisce due città distanti centinaia di chilometri tra loro, la simulazione di questi sistemi non può prescindere dalla velocità finita di propagazione dei segnali. Quindi essa deve essere modellata nel simulatore Spice mediante le linee di trasmissione.

In questo capitolo, illustreremo l'utilizzo dei vari modelli di linee di trasmissione disponibili in Spice, attraverso esempi particolarmente interessanti nelle applicazioni.

7.2 Richiami sulle linee di trasmissione

La linea di trasmissione è il più semplice modello utilizzato per rappresentare i segnali elettrici ed elettromagnetici che si propagano su un conduttore lungo una direzione prefissata. Nelle sue forme più semplici, essa può semplicemente essere costituita da due conduttori metallici che corrono paralleli in aria o incapsulati in un mezzo dielettrico che ha anche una funzione di sostegno. Oppure può essere realizzata nella forma del cavo coassiale, in cui un conduttore centrale è completamente chiuso in un altro conduttore cavo e concentrico. Anche in quest'ultimo caso l'intercapedine tra i due conduttori può essere riempita di dielettrico, che funge anche da separatore e distanziatore tra i conduttori, oppure può essere lasciata in aria, nel qual caso sono necessari dei distanziatori posti a opportuna distanza tra loro.

In realtà esistono molti altri tipi di strutture rappresentabili mediante linee di trasmissione, ricordiamo, ad esempio, le microstrisce e la propagazione nelle guide d'onda. Quanto detto implica che lo studio delle linee con il simulatore Spice diventa fondamentale per molte applicazioni.

In Figura 7.1 abbiamo rappresentato schematicamente la sezione trasversa di alcune tra le più semplici configurazioni di linea di trasmissione; in realtà i primi due tipi schematizzano la disposizione dei cavi che portano l'energia elettrica, il terzo rappresenta la classica struttura del cavo coassiale.

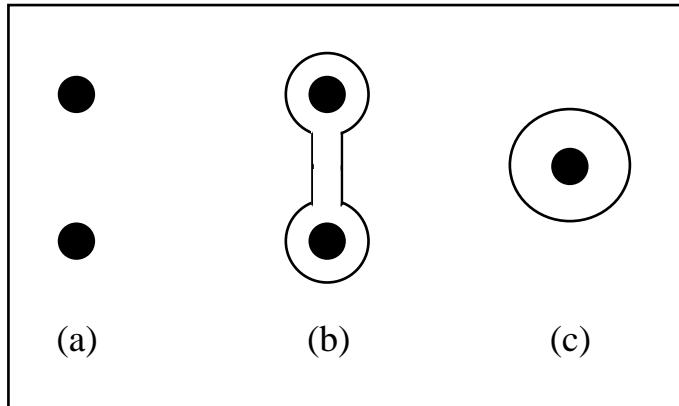


Figura 7.1: sezioni di (a) linea bifilare, (b) piattina e (c) cavo coassiale

Data una linea di trasmissione si possono definire la tensione $v(z,t)$ e la corrente $i(z,t)$, funzioni della coordinata longitudinale e del tempo, in ogni sezione trasversale. Queste grandezze devono soddisfare il sistema di equazioni alle derivate

$$\begin{cases} \frac{v}{z} = -L \frac{i}{t}, \\ \frac{i}{z} = -C \frac{v}{t}, \end{cases}$$

dove L e C sono rispettivamente l'induttanza e la capacità della linea per unità di lunghezza e sono dette parametri primari della linea. Queste due equazioni sono anche conosciute come *equazioni delle linee*. I parametri L e C , in generale, possono dipendere anche dalla coordinata z nel caso in cui la linea non sia spazialmente uniforme (si pensi ai conduttori della linea bifilare che si allontanano tra loro o a un cavo coassiale riempito di un dielettrico le cui proprietà dipendono dalla coordinata z).

Possiamo, eliminare le derivate rispetto al tempo, trasformando il sistema nel dominio dei fasori. In tal caso risulta

$$\begin{cases} \frac{d\bar{V}}{dz} = -j L \bar{I}, \\ \frac{d\bar{I}}{dz} = -j C \bar{V}, \end{cases}$$

in cui, come avrete certamente notato, le derivate spaziali sono diventate derivate totali. In questa seconda forma, le due equazioni vengono dette *equazioni dei telegrafisti*.

A partire dai parametri primari è possibile definire i cosiddetti parametri secondari

$$Z_0 = \sqrt{\frac{L}{C}}, \quad k = \sqrt{LC},$$

dove Z_0 viene detta impedenza caratteristica della linea, i cui valori tipici sono 50, 75 e 220, e k viene detta costante di propagazione della linea. Nel caso in cui la linea non sia nel vuoto ma immersa in un mezzo dielettrico (omogeneo) di permittività ϵ_r , allora la velocità di fase con cui si propaga il fronte d'onda di un segnale lungo la linea sarà inferiore alla velocità della luce e pari a

$$v = \frac{1}{\sqrt{LC}} = \frac{c}{\sqrt{\epsilon_r}},$$

avendo indicato con ‘c’ la velocità della luce nel vuoto. L’onda, di tensione o di corrente che sia, avrà un periodo temporale T e un periodo spaziale λ , legati tra loro dalle ben note relazioni

$$T = \frac{\lambda}{v}, \quad \lambda = \frac{2\pi}{k} \quad \text{e} \quad \frac{\lambda}{T} = v.$$

Invece, nel caso in cui la linea di trasmissione presenti piccole perdite, le equazioni dei telegrafisti diventano

$$\begin{cases} -\frac{d\bar{V}}{dz} = j \quad L \bar{I} + R \bar{I} = j \quad \left(L + \frac{R}{j}\right) \bar{I} = j \quad L_{EQ} \bar{I}, \\ -\frac{d\bar{I}}{dz} = j \quad C \bar{V} + G \bar{V} = j \quad \left(C + \frac{G}{j}\right) \bar{V} = j \quad C_{EQ} \bar{V}, \end{cases}$$

dove R e G sono rispettivamente la resistenza e la conduttanza di linea, sempre considerate per unità di lunghezza. La resistenza R è legata principalmente alle perdite ohmiche nei conduttori della linea mentre la conduttanza G è legata alle perdite nel dielettrico. I parametri L_{EQ} e C_{EQ} sono detti, rispettivamente, induttanza e capacità ‘equivalenti’ e sono grandezze complesse. I parametri secondari della linea si modificano come

$$k = \sqrt{L_{EQ} C_{EQ}} = \sqrt{\left(L + \frac{R}{j}\right) \left(C + \frac{G}{j}\right)},$$

$$Z_0 = \sqrt{\frac{L_{EQ}}{C_{EQ}}} = \sqrt{\frac{R + j \frac{L}{C}}{G + j \frac{C}{L}}},$$

ed entrambi diventano quantità complesse. La linea si dice ‘con piccole perdite’ se

$$R \ll L, \quad G \ll C,$$

in tal caso, sviluppando in serie la radice quadrata, giungiamo facilmente alle due formule approssimate

$$k = \sqrt{LC} - j \frac{1}{2} \left(\frac{R}{Z_0} + G Z_0 \right),$$

$$Z_0 = \sqrt{\frac{L}{C}} - j \frac{1}{2k} \left(R - G \frac{L^2}{C^2} \right),$$

in cui la parte immaginaria della costante di propagazione k è detta ‘costante di attenuazione’ (Neper/m) e, anche se molto piccola, non può essere trascurata, specialmente se la linea copre distanze molto lunghe rispetto alla lunghezza d’onda del segnale. Notiamo che l’impedenza caratteristica è complessa e comporta quindi uno sfasamento tra l’onda di tensione e quella di corrente lungo la linea. Vi è, però, un caso in cui, pur essendo presenti delle perdite, essa è comunque reale: ciò accade se è verificata la condizione di Heaviside:

$$\frac{R}{L} = \frac{G}{C},$$

come potete facilmente verificare nelle formule precedenti.

7.3 Linee di trasmissione ideali

Consideriamo la seguente istruzione Spice

```
T1    1    0    2    0    Z0=220    F=400    NL=1.5
```

essa specifica una linea di trasmissione di nome T1 (d’altra parte tutti i nomi delle linee devono cominciare con la lettera T) la cui porta d’ingresso è individuata dai nodi 1 e 0 e la porta di uscita dai nodi 2 e 0, l’impedenza caratteristica è 220

(nel simbolo Z0 si può usare sia lo ‘zero’ che la lettera ‘O’) e la linea è lunga 1.5 volte (NL=1.5) la lunghezza d’onda caratteristica corrispondente alla frequenza di 400 Hz (F=400).

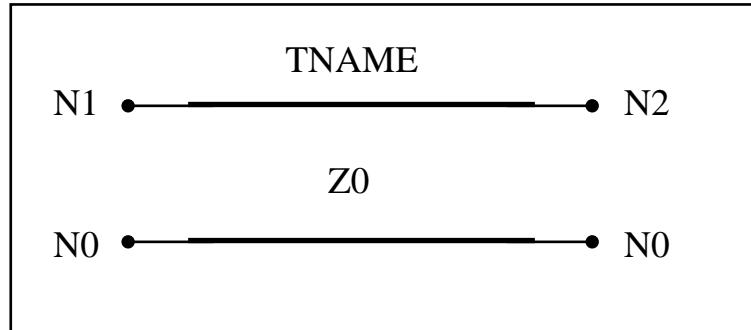


Figura 7.2: linea di trasmissione ideale

La forma generale per l’istruzione che specifica una linea di trasmissione ideale in Spice è

TNAME N1 N0 N2 N0 Z0=VAL TD=VAL

oppure

TNAME N1 N0 N2 N0 Z0=VAL F=VAL <NL=VAL>

in cui il nome deve obbligatoriamente iniziare con la lettera T, N1 e N0 sono i nodi della porta d’ingresso, N2 e N0 quelli della porta d’uscita, Z0 rappresenta l’impedenza caratteristica della linea ed è un’informazione necessaria per il simulatore. Poi bisogna specificare il tempo di ritardo, oppure la lunghezza caratteristica della linea a una certa frequenza. Se viene indicata solo la frequenza, la linea è considerata lunga un quarto della lunghezza d’onda (NL=0.25).

Queste informazioni sono perfettamente equivalenti tra loro: ad esempio, le tre righe di programma seguenti specificano tutte la stessa linea di trasmissione

T1	1	2	3	2	Z0=220	TD=115N ,
T2	1	2	3	2	Z0=220	F=2.25MEG ,
T3	1	2	3	2	Z0=220	F=4.5MEG NL=0.5 .

I tre parametri introdotti precedentemente sono legati tra loro dalle ben note relazioni

$$TD = \frac{NL}{F}, \quad NL = L, \quad = \frac{v}{F},$$

dove L è la lunghezza fisica della linea, λ è la lunghezza d'onda del segnale alla frequenza F e, infine, v è la velocità di fase sulla linea.

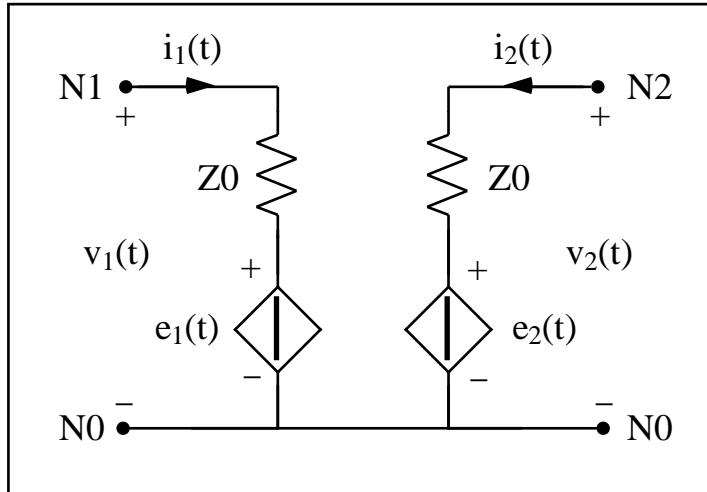


Figura 7.3: modello equivalente per una linea di trasmissione ideale

In Figura 7.3 è rappresentato il modello equivalente di Spice per una linea di trasmissione. Le equazioni che definiscono il valore istantaneo dei generatori $e_1(t)$ e $e_2(t)$ sono

$$e_1(t) = v_2(t - TD) + Z_0 i_2(t - TD),$$

$$e_2(t) = v_1(t - TD) + Z_0 i_1(t - TD),$$

per cui, la tensione a una porta al tempo t è legata alla tensione e alla corrente sull'altra porta al tempo $t - TD$. Consideriamo ora, come primo esempio, un semplice circuito costituito da una linea di trasmissione chiusa su un carico ‘adattato’. La parola adattato ci informa che il carico è tale da assorbire completamente l’onda di tensione in arrivo e non generare alcuna onda riflessa.

Nel listato Spice che segue abbiamo utilizzato in ingresso alla linea un generatore di tensione lineare a tratti con un valore massimo di 4V di durata 10 ms, i tempi di salita e discesa sono uguali a 2 ms. La linea è terminata su un carico di 50 . L’istruzione ‘.IC’ stabilisce le condizioni iniziali per i nodi del circuito: in questo caso ci siamo assicurati che la linea fosse ‘scarica’ all’istante iniziale.

Vale la pena notare che nella analisi in transitorio con le linee di trasmissione la costante di tempo interna per il calcolo è sempre minore o uguale della metà del più piccolo tempo di ritardo delle linee presenti nel circuito: questo vuol dire che linee di trasmissione molto piccole provocheranno un incremento notevole del tempo di calcolo. Inoltre, se vogliamo che siano rispettati i tempi di salita e di discesa dell'onda, bisogna che la costante di tempo per il calcolo del transitorio sia più piccola di essi. Nell'esempio precedente abbiamo risolto questo problema usando l'estensione TMAX che abbiamo introdotto nel Capitolo 4 quando discutemmo l'istruzione '.TRAN'.

Esempio 7.1

* Linea di trasmissione terminata su un carico adattato

```

V1      1      0      PWL(0 0 2m 4 8m 4 10m 0)
T1      1      0      2      0      Z0=50      TD=15m
.IC      V(2)=0
R1      2      0      50
.TRAN    0.1m    60m      0      0.1m   UIC
.PROBE
.END

```

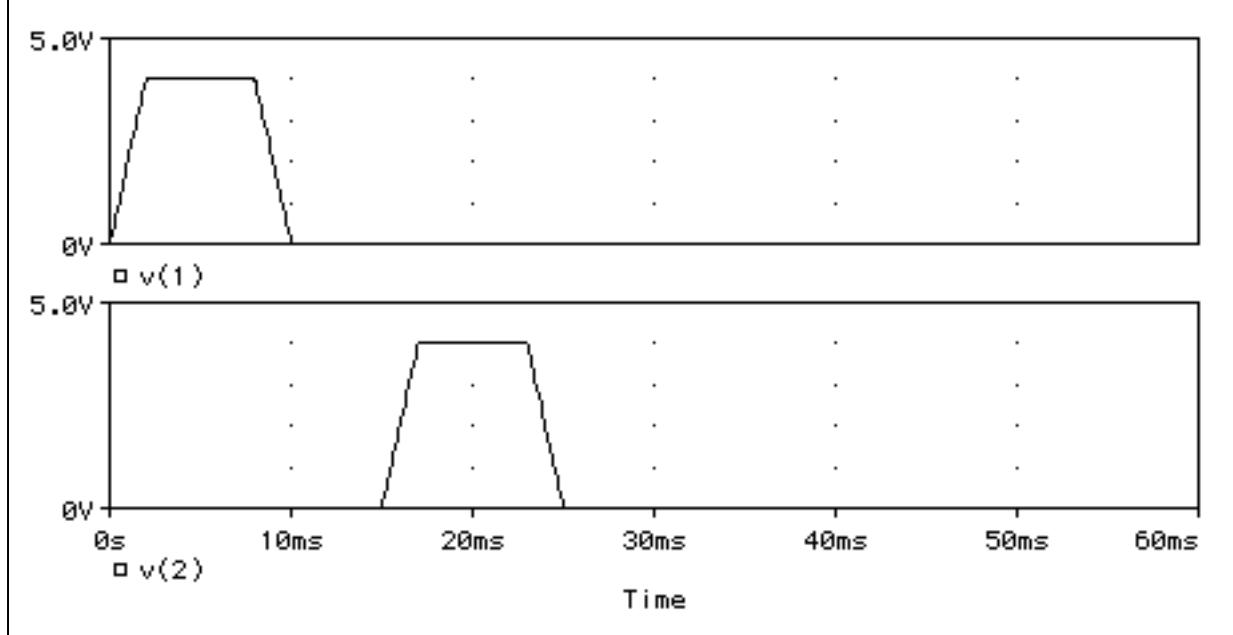


Figura 7.4: linea terminata su un carico adattato.

Dato che la linea è senza perdite e la sua resistenza caratteristica coincide proprio con la resistenza R1 su cui è terminata, essa è adattata: l'onda di tensione prodotta dal generatore impiega 15 ms per transitare lungo la linea e, una volta arrivata

all'altra estremità, non viene più riflessa. Questo è il significato della parola ‘adattamento’. Notate che per la prima volta, almeno in questo libro, state studiando un circuito a costanti distribuite, un circuito nel quale la propagazione dei segnali non è più istantanea.

Ora proviamo a terminare la linea su un carico disadattato.

Esempio 7.2

```
* Linea di trasmissione terminata su un carico disadattato
V1    1    0    PWL(0 0 2m 4 8m 4 10m 0)
T1    1    0    2    0    Z0=50      TD=15m
.IC    V(2)=0
R1    2    0    10
.TRAN    0.1m    60m    0    0.1m   UIC
.PROBE
.END
```

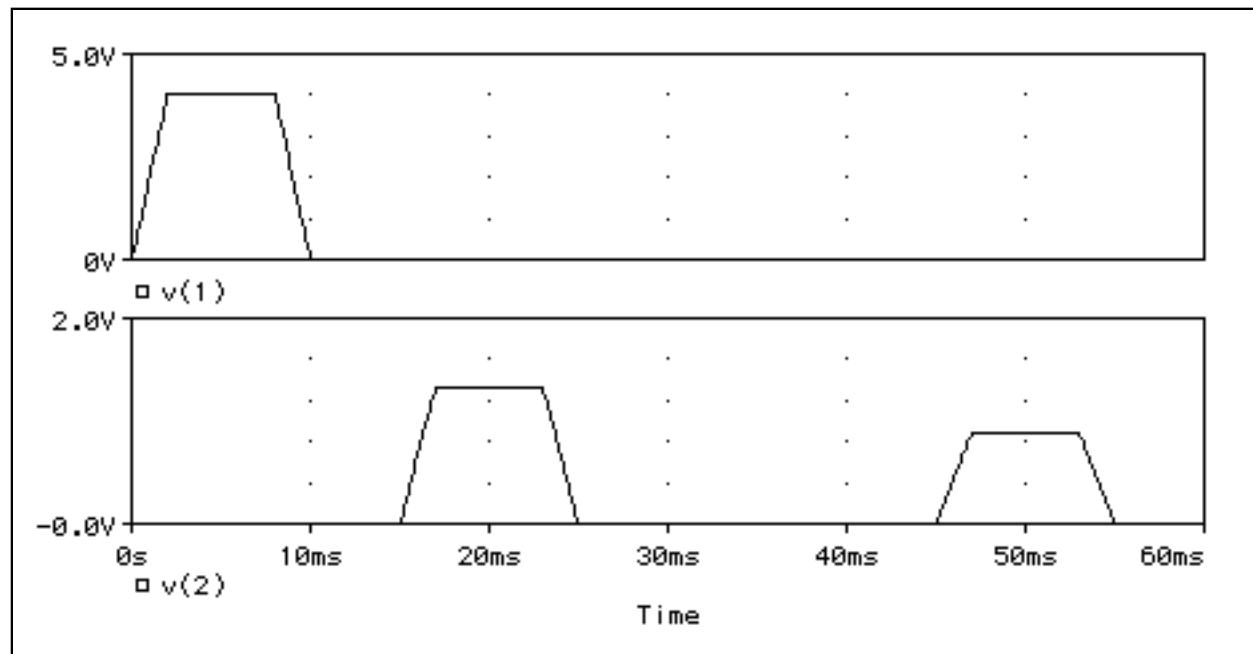


Figura 7.5: linea terminata su un carico disadattato.

Notate che la tensione sulla porta secondaria è non nulla ogni 30 ms, il tempo che impiega la perturbazione per andare e tornare dalla stessa terminazione. Tuttavia la seconda onda riflessa è attenuata rispetto alla prima a causa della potenza irreversibilmente dissipata ad ogni passaggio sul resistore R1.

Proviamo ora il caso in cui la linea è terminata in circuito aperto, ovvero su un resistore di valore molto elevato rispetto alla resistenza della linea

Esempio 7.3

```
* Linea di trasmissione terminata su un circuito aperto
V1      1      0      PWL(0 0 2m 4 8m 4 10m 0)
T1      1      0      2      0      Z0=50      TD=15m
.IC      V(2)=0
R1      2      0      1meg
.TRAN    0.1m    60m      0      0.1m    UIC
.PROBE
.END
```

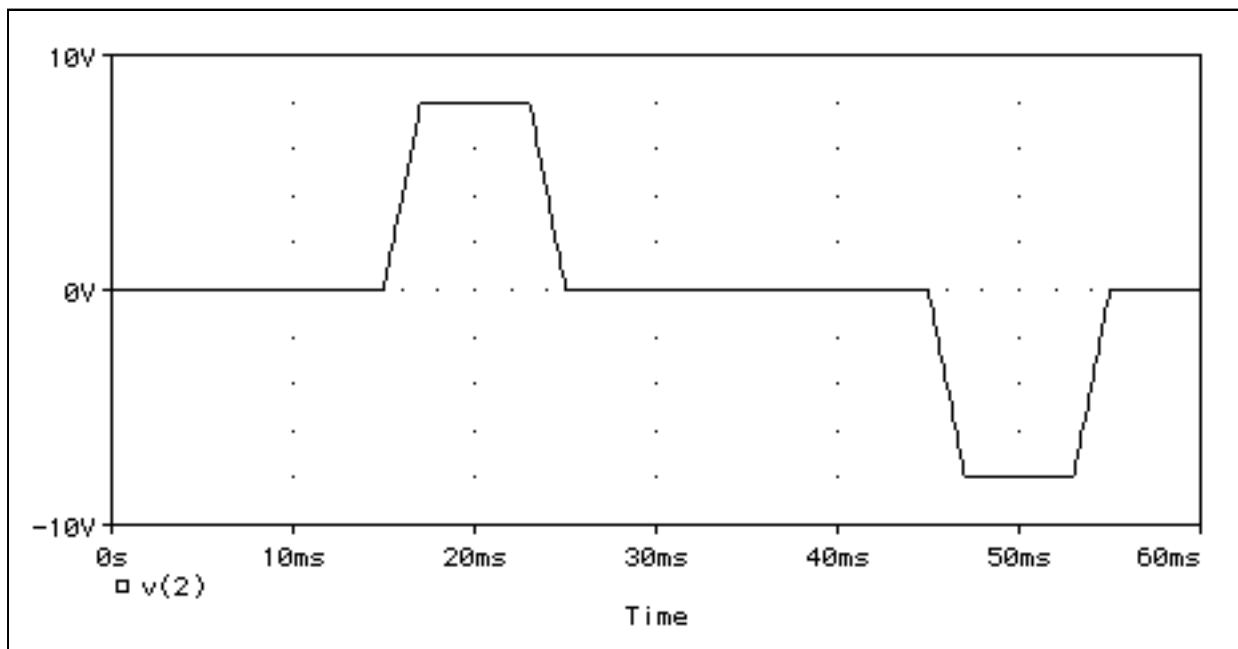


Figura 7.6: linea terminata su un circuito aperto.

Prestate attenzione alla figura: questa volta la tensione alla seconda porta, non solo va e torna a intervalli di 30 ms, ma ha raddoppiato il suo valore massimo che è passato dai 4 V del generatore in ingresso ai ben 8 V sulla terminazione.

Come è possibile ciò?

L'onda di ampiezza 8 V è il risultato della composizione dell'onda di 4V in arrivo dal generatore e dell'onda riflessa dal 'quasi' circuito aperto.

Inoltre le onde che viaggiano sulla linea, stavolta, non sono affatto attenuate: infatti la resistenza elevata che simula un circuito aperto dissipa ad ogni passaggio solo una frazione non apprezzabile dell'onda stessa.

Adesso proviamo il caso opposto in cui la linea è terminata in cortocircuito.

Esempio 7.4

* Linea di trasmissione terminata su un cortocircuito

```
V1 1 0 PWL(0 0 2m 4 8m 4 10m 0)
T1 1 0 2 0 Z0=50 TD=15m
.IC V(2)=0
R1 2 0 1e-6
.TRAN 0.1m 60m 0 0.1m UIC
.PROBE
.END
```

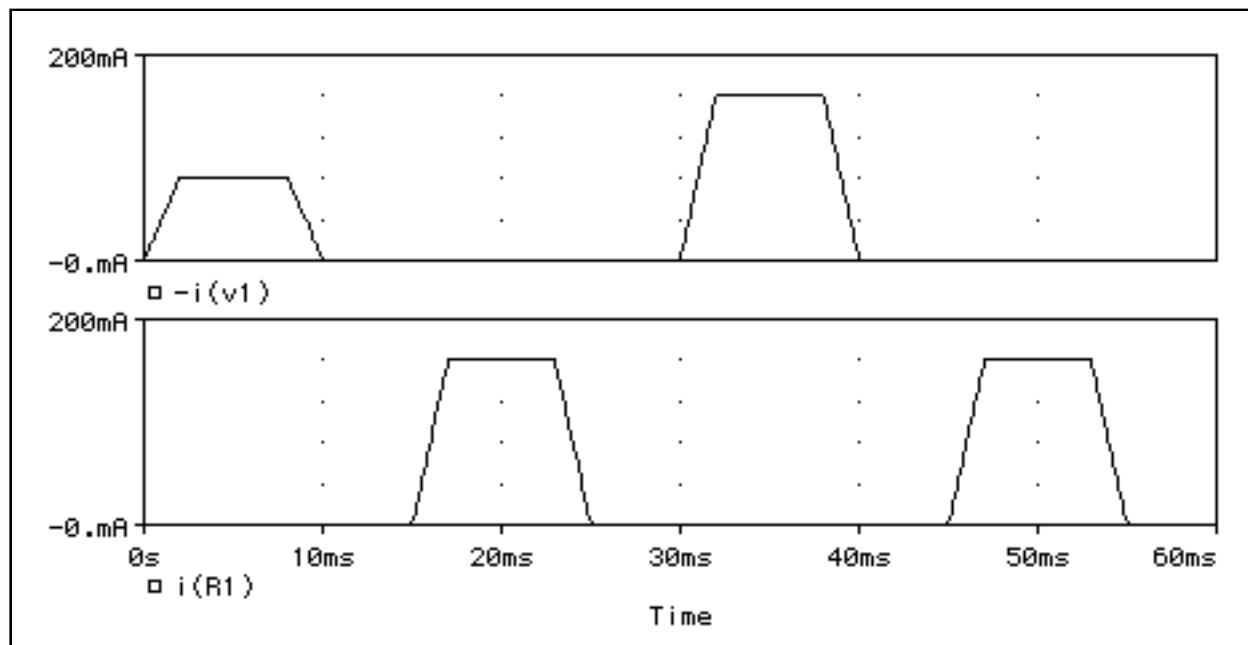


Figura 7.7: linea terminata su un cortocircuito.

Stavolta ci affidiamo alla corrente che scorre nel cortocircuito in quanto le tensioni sarebbero inesorabilmente nulle. L'impulso di corrente che parte dal generatore ha ampiezza massima 80 mA in quanto sappiamo che la linea equivale a un carico uguale alla sua resistenza caratteristica. Invece l'onda di corrente sul cortocircuito ha ampiezza doppia perché si sommano in fase i due contributi dell'onda incidente e dell'onda riflessa.

Cosa accade se la linea è terminata con dei bipoli a memoria?

La risposta completa non è semplice e esula dai limiti che ci siamo imposti. Tuttavia, per farvi intuire le dinamiche nuove che si possono innescare, vi proponiamo di chiudere la linea su un condensatore e su un induttore

Esempio 7.5

```
* Linea di trasmissione terminata su un condensatore
V1 1 0 PWL(0 0 2m 4 8m 4 10m 0)
T1 1 0 2 0 Z0=50 TD=15m
.IC V(2)=0
C1 2 0 10u IC=0
.TRAN 0.1m 150m 0 0.1m UIC
.PROBE
.END
```

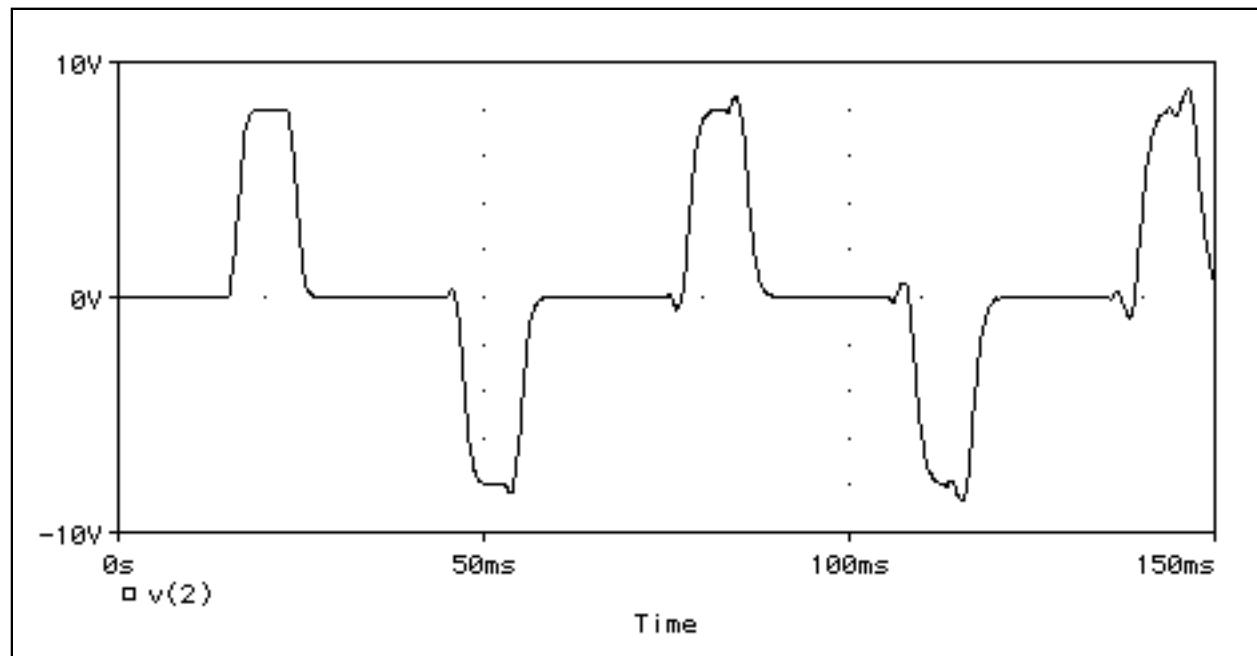


Figura 7.8: linea terminata su un condensatore.

Notate quanto è diventata complessa la dinamica della tensione sul condensatore: durante il primo passaggio riconoscete i fenomeni di carica e di scarica della capacità, poi le riflessioni multiple complicano ... maledettamente la dinamica.

Una cosa, però, resta vera: durante il transitorio la tensione ai capi del condensatore resta continua.

Per la corrente che attraversa un induttore che termina la linea si possono ripetere considerazioni analoghe per quelle appena fatte per la tensione ai capi di un condensatore.

Esempio 7.6

```
* Linea di trasmissione terminata su un induttore
V1    1    0    PWL(0 0 2m 4 8m 4 10m 0)
T1    1    0    2    0    Z0=50      TD=15m
.IC    V(2)=0
L1    2    0    50m    IC=0
.TRAN    0.1m    150m    0    0.1m    UIC
.PROBE
.END
```

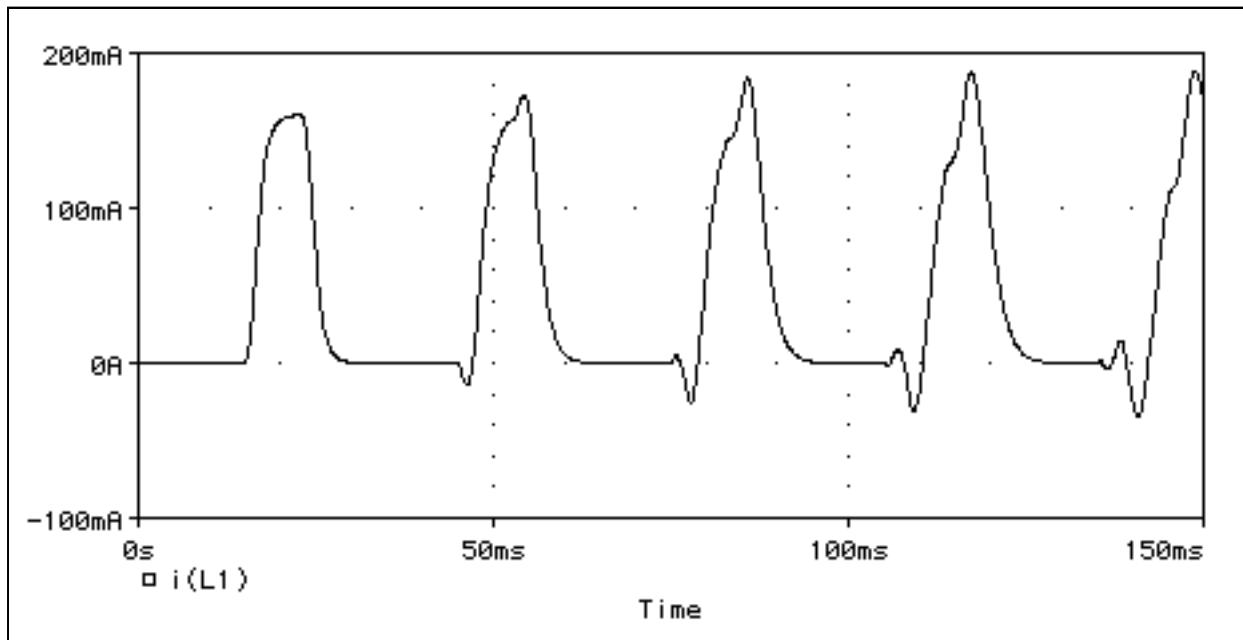


Figura 7.9: linea terminata su un induttore.

Riteniamo che questo possa bastare per simulare le linee di trasmissione senza perdite con Spice.

7.4 Linee di trasmissione con perdite

Spice consente la simulazione di circuiti elettrici contenenti linee di trasmissione con perdite. Le caratteristiche della linea vengono immesse mediante l'uso dei parametri primari R, L, G, C. Notiamo che nel caso in cui siano nulli i parametri

R e G ricadiamo nel modello di linea senza perdite, esaminato nel paragrafo precedente.

Il modello utilizzato da Spice per simulare la linea di trasmissione con perdite è simile a quello ideale ma le tensioni e le correnti, oltre a essere ritardate per effetto della propagazione, devono essere attenuate a causa delle perdite proprie della linea.

Come al solito, cominciamo con un esempio

```
T1 1 0 2 0 LEN=1 R=0.311 L=0.378u G=6.27u C=67.3p .
```

In tal caso abbiamo comunicato al simulatore che vogliamo simulare una linea lunga un metro (LEN=1), definita dai parametri R, L, G e C tipici del cavo coassiale di tipo commerciale RG-11/U.

La prima lettera che contraddistingue, come d'abitudine, il nome della linea è sempre la T, mentre ai parametri tocca il compito di indicare se la linea è con perdite o meno.

I parametri della linea possono essere specificati anche attraverso l'uso di espressioni elementari e ciò facilita l'utilizzo, in quanto, ad esempio, in tal modo è sempre possibile specificare come costanti (istruzione .PARAM) l'impedenza caratteristica e la costante dielettrica relativa del mezzo, per poi ricavare i parametri primari R, L, G e C.

Spice offre anche la grossa opportunità di poter esprimere i parametri R e G attraverso la loro trasformata di Laplace, in questo modo possiamo modellare ad esempio l'effetto pelle e le perdite nel dielettrico che separa i conduttori, come vedremo nel seguito. Per introdurre le linee di trasmissione con perdite cominciamo con un esempio nel dominio del tempo.

Esempio 7.7

```
* Una simulazione con il cavo RG-11/U
V1 1 0 PWL(0 0 0.5n 4 1n 0)
T1 1 0 2 0 LEN=1 R=0.311 L=0.378u G=6.27u C=67.3p
.IC V(2)=0
R1 2 0 50
.TRAN 0.1n 20n 0 0.01n UIC
.PROBE
.END
```

Come potete controllare nella Figura 7.10, il tempo di ritardo è circa

$$TD = \frac{L}{V} \sqrt{0.378} \quad 67.3 \text{ ns} \quad 5.04 \text{ ns} .$$

Inoltre notate che dopo il primo passaggio la durata del segnale comincia piano piano ad aumentare: la seconda onda triangolare, oltre a essere più piccola a causa delle perdite, ha una durata, stimabile a occhio, un po' più grande di 1 ms che è la durata dell'impulso prodotto dal generatore.

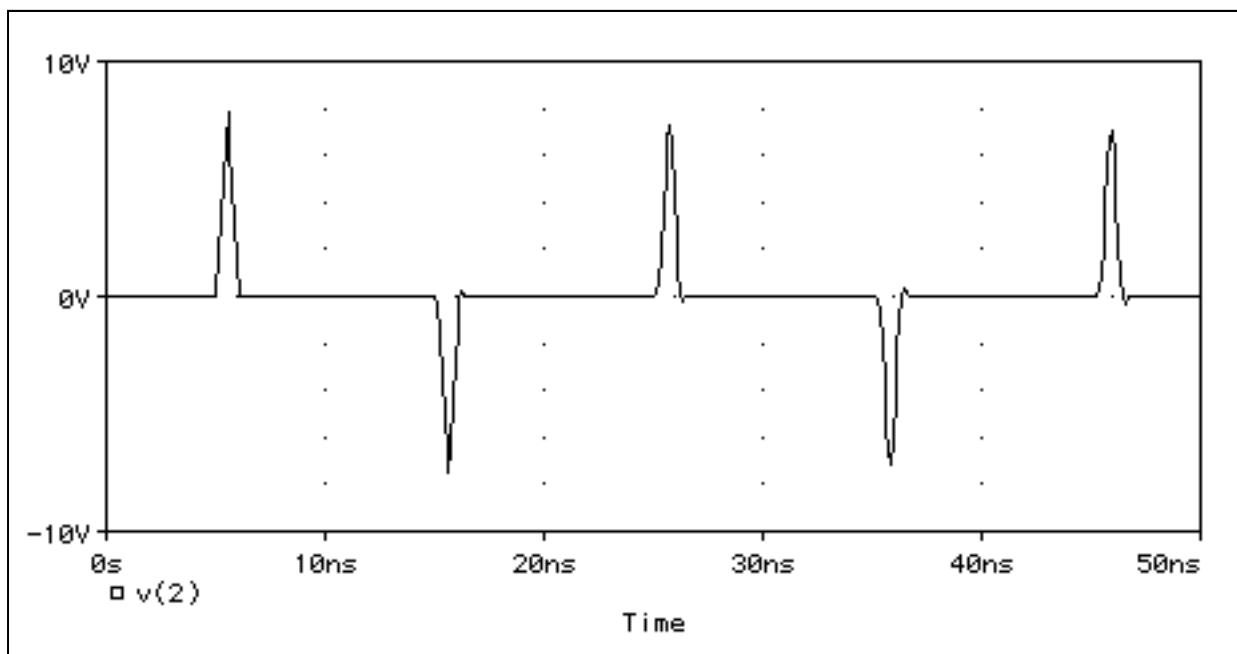


Figura 7.10: linea con perdite terminata su un carico resistivo.

L'allungamento della durata dell'impulso si può evitare solo usando linee per le quali vale la condizione di Heaviside:

$$\frac{R}{L} = \frac{G}{C} .$$

Esempio 7.8

* Condizione di Heaviside

```
V1 1 0 PWL(0 10 0.4 10 0.401 0)
T1 1 0 2 0 LEN=1 R=10m L=1 G=10m C=1
.IC V(2)=0
R1 2 0 5e6
.TRAN 10m 20 0 10m UIC
.PROBE
.END
```

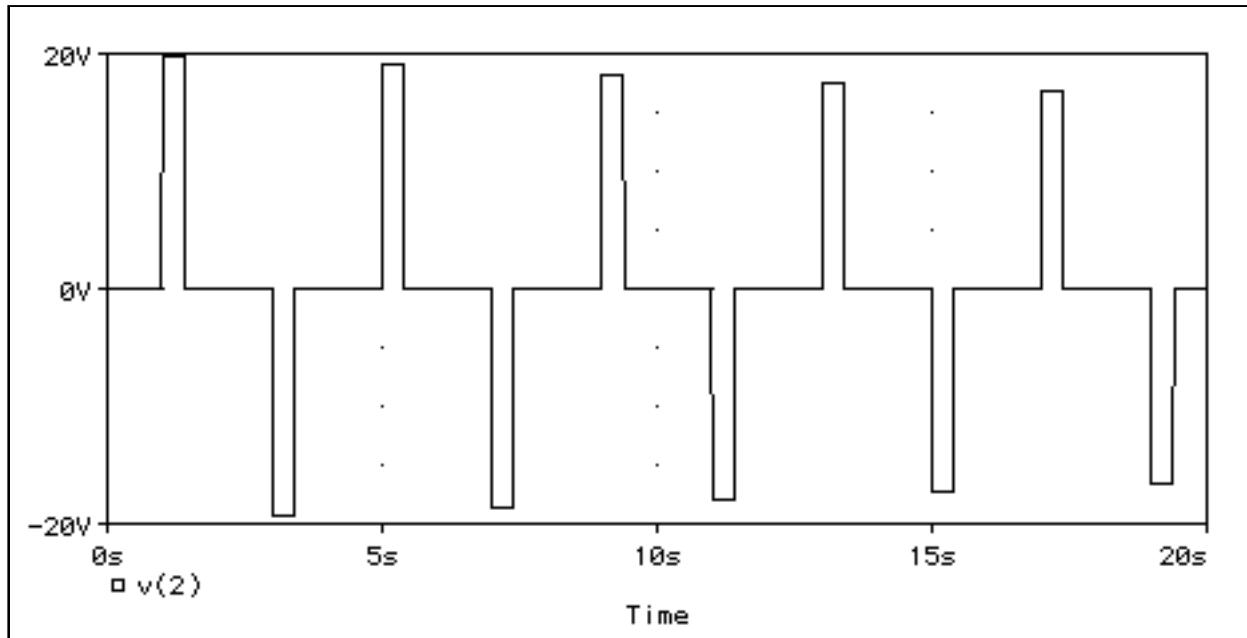


Figura 7.11: linea con perdite soddisfacenti la condizione di Heaviside.

L'esempio 7.8 mostra una tale linea nella quale un'onda quadra, terminata su un circuito praticamente aperto, dolcemente si attenua senza alterare la sua durata.

Passiamo ora ad un'analisi nel dominio della frequenza considerando l'esempio di seguito riportato.

Esempio 7.9

```
* Linee di trasmissione con perdite nel dominio della frequenza
*
.PARAM mu={4*3.14159*1e-7}, ep={8.856*1e-12}, cl={1/sqrt(mu*ep)}
.PARAM zeta0=75, er=9, trit=3m, vp={cl/sqrt(er)}
V1    1    0    AC    1
R1    1    2    75
T4    2    0    3    0
+ LEN={vp*trit} R =50u L={zeta0/vp} G=0.1n C={1/(zeta0*vp)}
R2    3    0    75
.AC    DEC    20      1    100meg
.PROBE
.END
```

In esso, i parametri L e C sono ottenuti a partire dall'impedenza caratteristica desiderata ($Z_0 = 75 \Omega$) e dalla velocità di fase dell'onda ottenuta a partire dalla

permittività relativa del dielettrico ($\epsilon_r = 9$). I parametri legati alle perdite R e G sono impostati direttamente. Anche la lunghezza della linea è impostata come prodotto tra la velocità di fase e il tempo di ritardo desiderato. Effettuiamo un'analisi in frequenza a decadi da 1 Hz fino a 100 MHz.

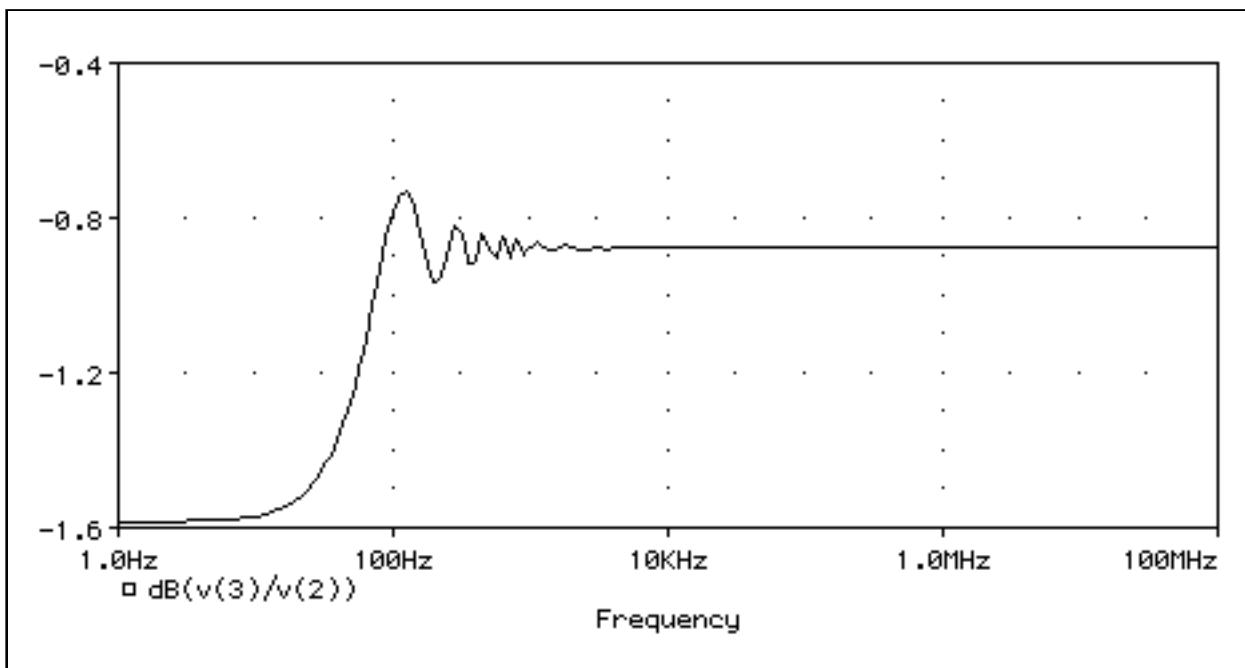


Figura 7.12: risposta in frequenza di una linea con perdite.

Questa figura mostra chiaramente come il rapporto riportato si assesti ad un valore costante per alte frequenze.

7.5 L'effetto pelle

Consideriamo una linea di trasmissione bifilare costituita da due conduttori metallici di sezione cilindrica di raggio a e di conducibilità elettrica σ . Per un tale conduttore si definisce lo spessore di penetrazione λ del campo elettromagnetico mediante la relazione:

$$\lambda(\omega) = \sqrt{\frac{2}{\mu_0 \sigma}} \cdot \omega^{-1}.$$

Esso ha le dimensioni di una lunghezza ed è funzione della frequenza del segnale elettromagnetico. Se lo spessore λ è più piccolo del raggio a del conduttore allora il campo elettromagnetico, ovvero la corrente che scorre nel conduttore, sarà confinato intorno alla superficie esterna, in uno spessore di pochi λ . È questo il

cosiddetto effetto pelle; la sua schematizzazione in Spice comporta che il parametro R sia rappresentato nel dominio di Laplace dall'espressione

$$R(s) = R_{DC} + K \sqrt{s}$$

dove R_{DC} è la resistenza ohmica del conduttore e, nel caso di conduttori cilindrici a sezione circolare, la costante K vale

$$K = \frac{1}{\pi} \sqrt{\frac{\mu}{\rho}},$$

quindi, essa risulta inversamente proporzionale al raggio e alla radice quadrata della conducibilità elettrica del conduttore. Notiamo, inoltre, che il parametro R aumenta con la frequenza.

In realtà, è importante notare che l'espressione per $R(s)$ è valida sia a bassa frequenza dove il termine preponderante è la R_{DC} , sia alle alte frequenze. In altri casi, spesso, entrano in gioco altri fenomeni che complicano lo studio e non permettono di ricavare una espressione chiusa per $R(s)$. Grazie alle possibilità offerte da Spice è possibile modellare tale comportamento nei parametri di una linea di trasmissione con perdite. Consideriamo il seguente esempio.

Esempio 7.10

```
* Effetto pelle nelle linee di trasmissione
*
.PARAM mu={4*3.14159*1e-7}, ep={8.856*1e-12}, cl={1/sqrt(mu*ep)}
.PARAM zeta0=75, er=9, trit=3m, vp={cl/sqrt(er)}*
V1    1    0    AC    1
R1    1    2    75
T4    2    0    3    0    LEN={vp*trit}
+ R={50U+9.37*1e-9*SQRT(s)}
+ L={zeta0/vp} G=0.1n C={1/(zeta0*vp)}
R2    3    0    75
.AC    DEC    20    1    100meg
.PROBE
.END
```

Si può notare dalla Figura 7.10 che la risposta in alta frequenza, a differenza di quanto mostrato nel paragrafo precedente, subisce un'attenuazione che aumenta con la frequenza stessa.

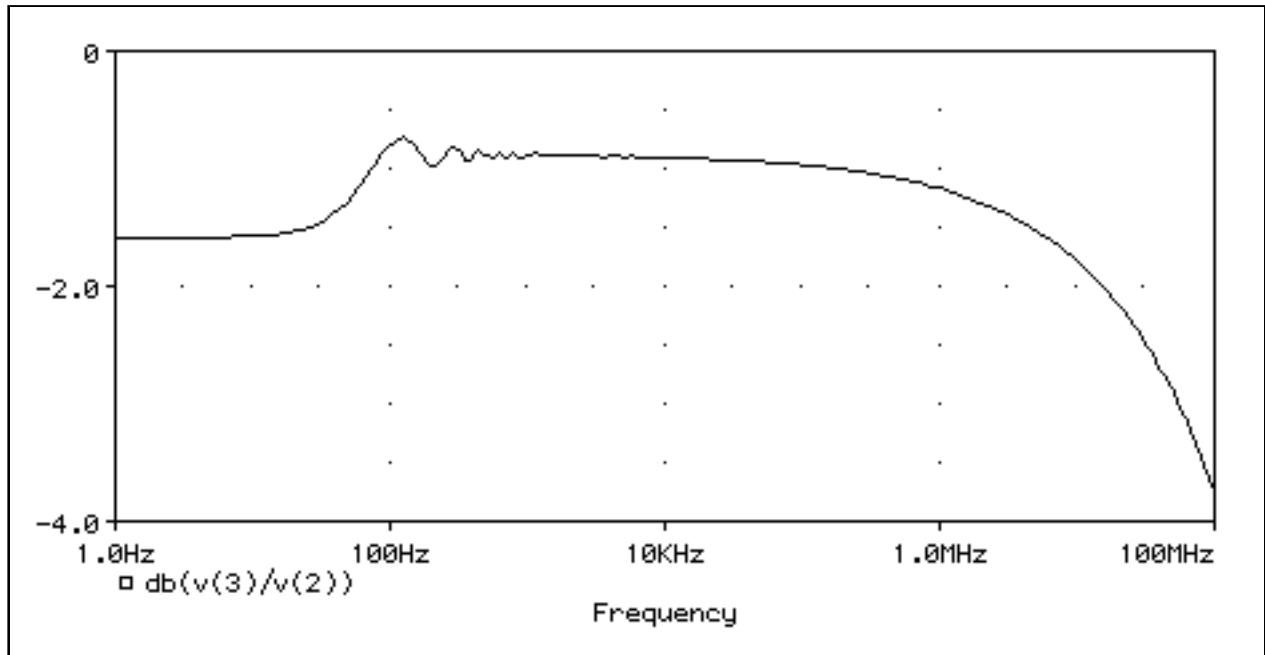


Figura 7.13: risposta in frequenza di una linea con perdite e effetto pelle.

7.6 Linee accoppiate

Spice offre anche la possibilità di includere nei circuiti le linee di trasmissione accoppiate. Consideriamo la generica riga di programma

```
KNAME    TNAME1      TNAME2      CM=<xx>      LM=<xx>
```

in cui il nome, che deve cominciare con la lettera 'k' (analogamente alla linea che definisce l'accoppiamento tra induttori), è seguito da quelli delle due linee di trasmissione da accoppiare e infine seguono i coefficienti mutui CM e LM di capacità ed induttanza.

La Figura 7.14 mostra una possibile realizzazione concreta di due linee accoppiate; notate che abbiamo due lunghi conduttori, denominati come linea 1 e 2, mostrati in sezione che poggiano su un substrato dielettrico che a sua volta è posto su un piano di massa. Nella figura non sono mostrate le sezioni di alimentazione e di carico.

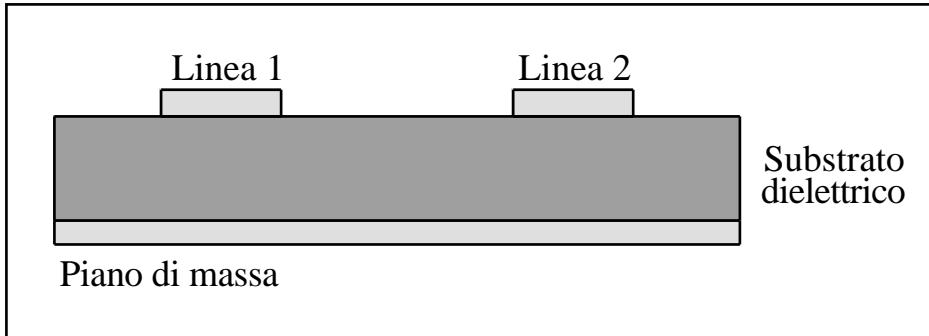


Figura 7.14: un esempio di linee accoppiate.

È importante notare che il modello da usare, per le linee che devono essere accoppiate tra loro, è quello con le perdite e inoltre che le due linee devono avere la stessa lunghezza. Consideriamo l'esempio di seguito riportato.

Esempio 7.12

```

* Linee di trasmissione accoppiate
*
.PARAM mu={4*3.14159*1e-7}, ep={8.856*1e-12}, cl={1/sqrt(mu*ep)}
.PARAM zeta0=75, er=9, trit=3m, vp={cl/sqrt(er)}
*
V1      1      0      PWL(0 0 1m 0 1.01m 10 2m 10 2.01m 0)
R1      1      2      75
*
T1      2      0      3      0      LEN={vp*trit}
+ R=0.1m  L={zeta0/vp}  G=0.1n  C={1/(zeta0*vp)}
T2      20     0      30     0      LEN={vp*trit}
+ R=0.1m  L={zeta0/vp}  G=0.1n  C={1/(zeta0*vp)}
K12    T1    T2    CM=0.13p  LM=0.75n
*
R2      3      0      75
R10     20     0      100meg
R20     30     0      100meg
.TRAN   5u      10m
.PROBE
.END

```

Abbiamo due linee accoppiate tra loro: la linea T1 è alimentata e chiusa su un carico adattato mentre la seconda linea T2 è praticamente lasciata in circuito

aperto da entrambe le porte, dato che è chiusa su due resistori di valore molto elevato.

Dal grafico riportato in Figura 7.15 si nota come l'impulso in ingresso alla linea T1 venga indotto anche sulla linea T2.

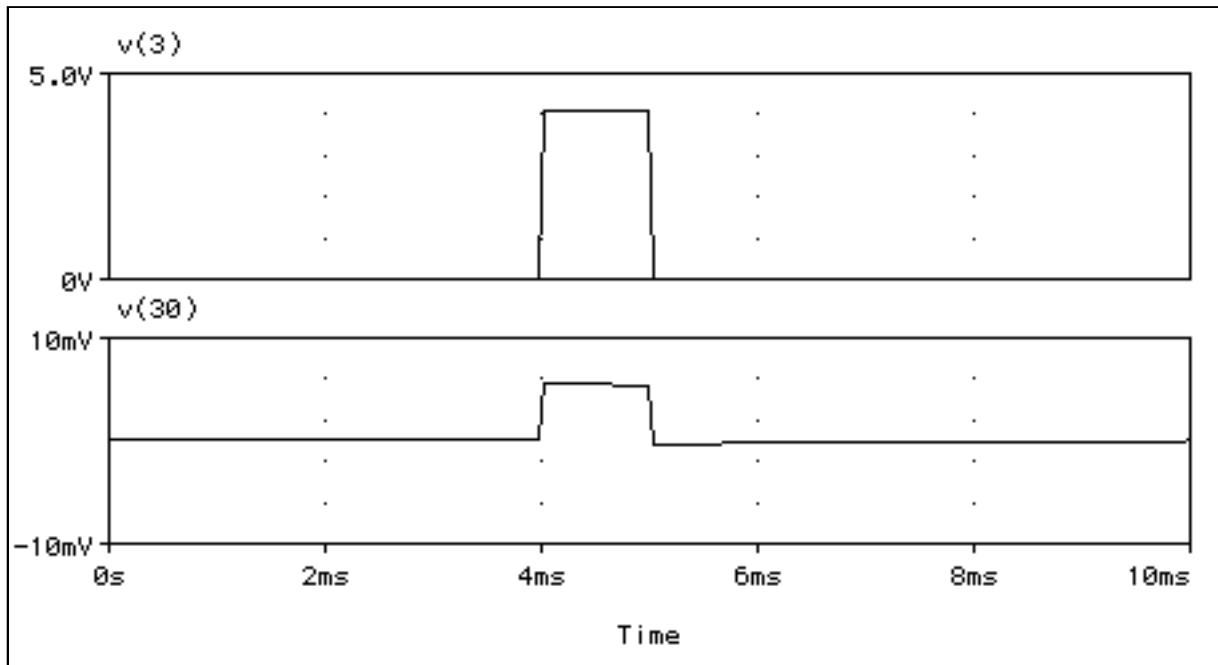


Figura 7.15: tensioni alle terminazioni di due linee accoppiate.

Notiamo infine che l'ultima versione di PSpice per PC dà la possibilità di simulare anche tre linee accoppiate su un riferimento comune, possibilità che apre la strada alla simulazione dei sistemi trifase di trasmissione dell'energia.

7.7 I parametri delle linee

In questo ultimo paragrafo riportiamo le formule che esprimono i parametri R, L, G e C rispetto alle caratteristiche geometriche dei tipi più diffusi di linee di trasmissione: la linea bifilare ed il cavo coassiale.

Consideriamo, innanzitutto, una coppia di cavi in aria di raggio ‘a’ e distanza mutua ‘d’, di conducibilità elettrica . In tal caso i parametri primari valgono

$$L = \frac{\mu_0}{2a} \operatorname{arccosh} \frac{d}{2a}, \quad C = \frac{0}{\operatorname{arccosh} \frac{d}{2a}},$$

$$R = \begin{cases} \frac{-2}{a^2}, & d > a, \\ \frac{-1}{a}, & d < a, \end{cases} \quad G = 0 \text{ S},$$

essendo d lo spessore di penetrazione nel materiale considerato alla frequenza di lavoro.

Invece, un cavo coassiale con il conduttore interno di raggio ‘a’ e il conduttore esterno di raggio ‘b’, riempito di dielettrico con tangente di perdita $\tan \delta$, è descrivibile per mezzo dei seguenti parametri primari:

$$L = \frac{\mu_0}{2} \log \frac{b}{a}, \quad C = \frac{2 \cdot 0_r}{\log \frac{b}{a}},$$

$$R = \begin{cases} \frac{1}{a^2} \left(\frac{1}{a^2} + \frac{1}{b^2} \right), & b > a, \\ \frac{1}{2a} \left(\frac{1}{a} + \frac{1}{b} \right), & b < a, \end{cases} \quad G = C \tan \delta.$$

Capitolo 8

Dispositivi e modelli a semiconduttore

8.1 Introduzione

8.2 Il diodo

8.2.1 Sintassi del modello per il diodo

8.2.2 Un esempio: il ponte di Graetz

8.3 Il transistore

8.3.1 Sintassi del modello per il transistore

8.3.2 Un esempio: alimentatore stabilizzato

8.4 I transistor a effetto di campo: il jfet e il mosfet

8.4.1 Un esempio: amplificatore CMOS

8.5 Come ottenere i parametri Spice di un dispositivo

8.6 L'amplificatore operazionale

8.6.1 L'amplificatore operazionale ideale

8.6.2 L'amplificatore operazionale reale

8.7 L'oscillatore a ponte di Wien

8.8 Modello completo di amplificatore operazionale

Sommario

Forse il pregio maggiore di Spice è la capacità di risolvere circuiti elettrici in cui siano presenti elementi non lineari. In questo capitolo esamineremo la sintassi utile a simulare i principali componenti elettronici a semiconduttore. Particolare attenzione è dedicata, alla fine del capitolo, allo studio dell'amplificatore operazionale.

8.1 Introduzione

Uno dei pregi maggiori di Spice è la capacità di risolvere le reti elettriche in cui siano presenti elementi non lineari per i quali l'analisi ‘a mano’ comporterebbe, laddove fosse possibile, un notevole dispendio di tempo e l'introduzione di numerose approssimazioni. Spice contiene i modelli dei più diffusi componenti a semiconduttore: basta specificare solo alcuni parametri per ottenere il componente che vogliamo simulare.

Ma quali sono i componenti a semiconduttore simulabili con Spice?

Praticamente tutti quelli presenti sul mercato: diodi, transistori, jfet, mosfet, scr e così via. Ogni nuovo componente riceve, prima o poi, il suo circuito equivalente per Spice, che permette di simularlo al meglio.

In questo capitolo cominceremo dagli elementi più noti: il diodo e il transistore, dando anche qualche cenno alle caratteristiche di tali dispositivi. Proseguiremo con i transistori a effetto di campo: il jfet e il mosfet. Poi introdurremo l'amplificatore operazionale, dando qualche breve cenno alle caratteristiche ed alla sua utilizzazione per mezzo di alcuni esempi.

Gli ultimi paragrafi del capitolo saranno dedicati a esempi complessi ed istruttivi, nei quali parleremo degli oscillatori e di un circuito equivalente completo dell'amplificatore operazionale.

8.2 Il diodo

Il diodo è un dispositivo (bipolo) a semiconduttore progettato per permettere il passaggio della corrente in una direzione e impedirlo nell'altra. Il suo simbolo circuitale è illustrato in Figura 8.1.

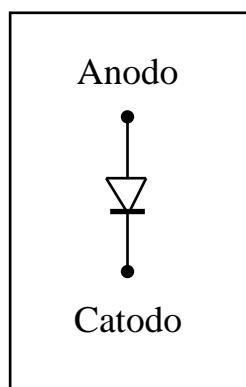


Figura 8.1: rappresentazione circuitale del diodo.

Il comportamento di un diodo (ideale) si può sintetizzare dicendo che esso consente il passaggio di corrente dall'anodo al catodo, ma non nella direzione opposta. In Figura 8.2 riportiamo la caratteristica statica di un diodo generico.

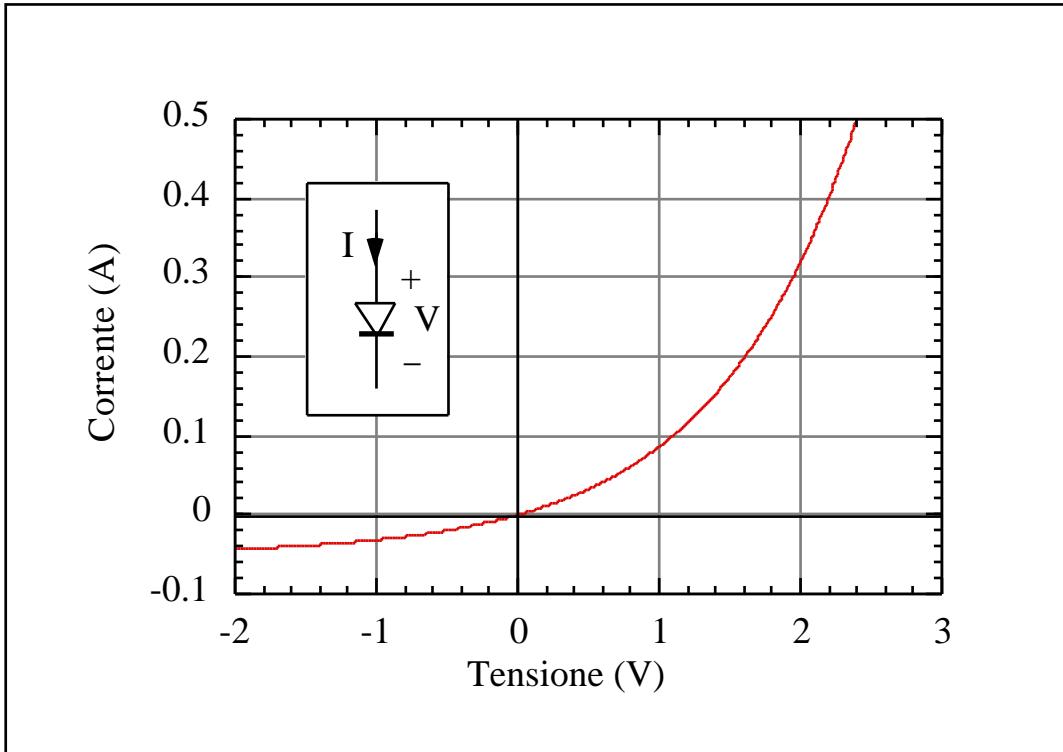


Figura 8.2: caratteristica statica di un diodo generico.

Idealmente, durante l'interdizione, il diodo dovrebbe comportarsi come un vero e proprio dispositivo di blocco per la corrente e essere invece un cortocircuito per tensioni maggiori di zero. In realtà, se la tensione diretta è maggiore di una piccola tensione, detta di soglia, il diodo fa passare una forte corrente, se invece si applica una tensione inversa il diodo conduce solo una piccolissima frazione della corrente che è in grado di sopportare in conduzione diretta (dell'ordine di un milione di volte più piccola) ma comunque diversa da zero, detta corrente di saturazione inversa.

Ovviamente, per un diodo reale, esiste un limite fisico alla funzione di blocco: se la tensione inversa applicata al diodo supera in valore assoluto un valore, detto di breakdown, si ha conduzione significativa anche di corrente inversa; questo però, nella maggior parte dei casi, corrisponde con la rottura del dispositivo stesso.

Come abbiamo già detto, in Spice esistono anche delle librerie che contengono già i modelli di molti dispositivi presenti sul mercato, tra cui anche i diodi. L'utente non deve fare altro che dichiarare l'uso della libreria all'inizio del file Spice e poi

usare il nome convenzionale assegnato al componente di interesse nella riga che lo introduce nel circuito.

Veniamo, finalmente, alla riga che definisce un semplice modello per un diodo

```
.MODEL      NOME      D      (IS=VALORE) .
```

Il comando **.MODEL** comunica a Spice che la riga è dedicata all'introduzione di un nuovo modello chiamato **NOME**; il terzo campo '**D**' indica che il modello riguarda un diodo e tra parentesi sono indicate i parametri caratteristici del modello. Ovviamente ogni parametro ha un suo valore di default, e noi in questo caso abbiamo forzato la corrente di saturazione inversa '**IS**'. La libreria di SPICE contiene decine e decine di '**.MODEL**', sempre più complicati e, quindi, più vicini alla realtà!

Ecco un semplice circuito che fa uso del diodo che abbiamo appena definito.

Esempio 8.1			
* Raddrizzatore a una semionda			
.MODEL DMOD1 D (IS=1e-9)			
D0	1	2	DMOD1
R0	2	0	50
C0	2	0	1m
V0	1	0	SIN(0 10 50 0 0)
.TRAN	1u	0.05	
.PROBE			
.END			

Tra i nodi 1 e 2 è connesso il diodo D0 che segue il modello indicato con il nome **DMOD1**. Il diodo D0 filtra la tensione sinusoidale del generatore V0 (si tratta di un generatore canonico di tipo sinusoidale con fattore di attenuazione nullo) e carica, durante la semionda positiva, il condensatore C0. Durante la semionda negativa il condensatore si scarica sulla resistenza di carico R0 in quanto il diodo, essendo un blocco, li isola dal generatore.

La Figura 8.3 mostra il grafico che PROBE fornisce per la tensione del generatore e per quella ai capi del condensatore.

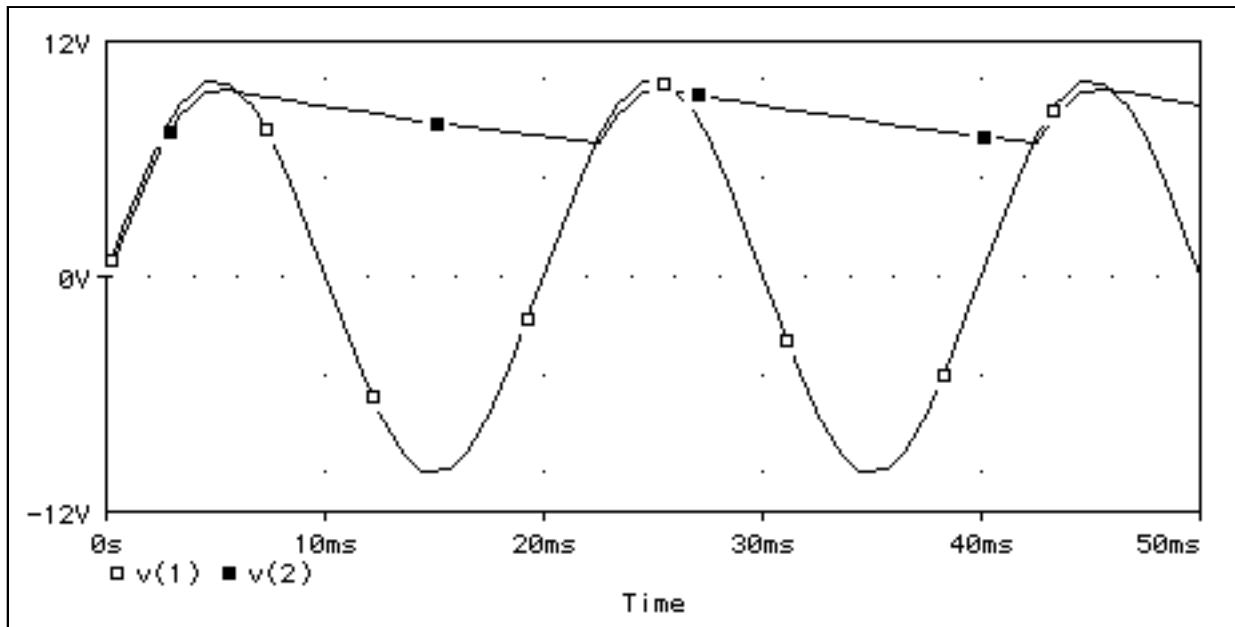


Figura 8.3: tensioni di uscita e ingresso per un raddrizzatore a una semionda.

Per illustrare l'importanza della tensione di breakdown, consideriamo l'esempio 8.2 in cui questo valore (Breakdown Voltage) è stato impostato a 7.5 V. Ovviamente il valore è del tutto irrealistico, nella realtà la tensione di breakdown è svariate centinaia di Volt!

Esempio 8.2

* Raddrizzatore a una semionda

```
.MODEL      DMOD1    D      (IS=1e-9, BV=7.5)
D0        1    2    DMOD1
R0        2    0    50
C0        2    0    1m
V0        1    0    SIN(0 10 50 0 0)
.TRAN      1u    0.05
.PROBE
.END
```

In questo caso ci aspettiamo che, quando la tensione inversa ai capi del diodo diventerà maggiore di 7.5 V, esso condurrà anche durante la semionda negativa, scaricandosi attraverso la capacità.

In Figura 8.4 possiamo notare il risultato di tale modifica: il diodo conduce di nuovo non appena la tensione di alimentazione diventa minore di - 7.5 V per cui il condensatore si scarica rapidamente grazie al generatore.

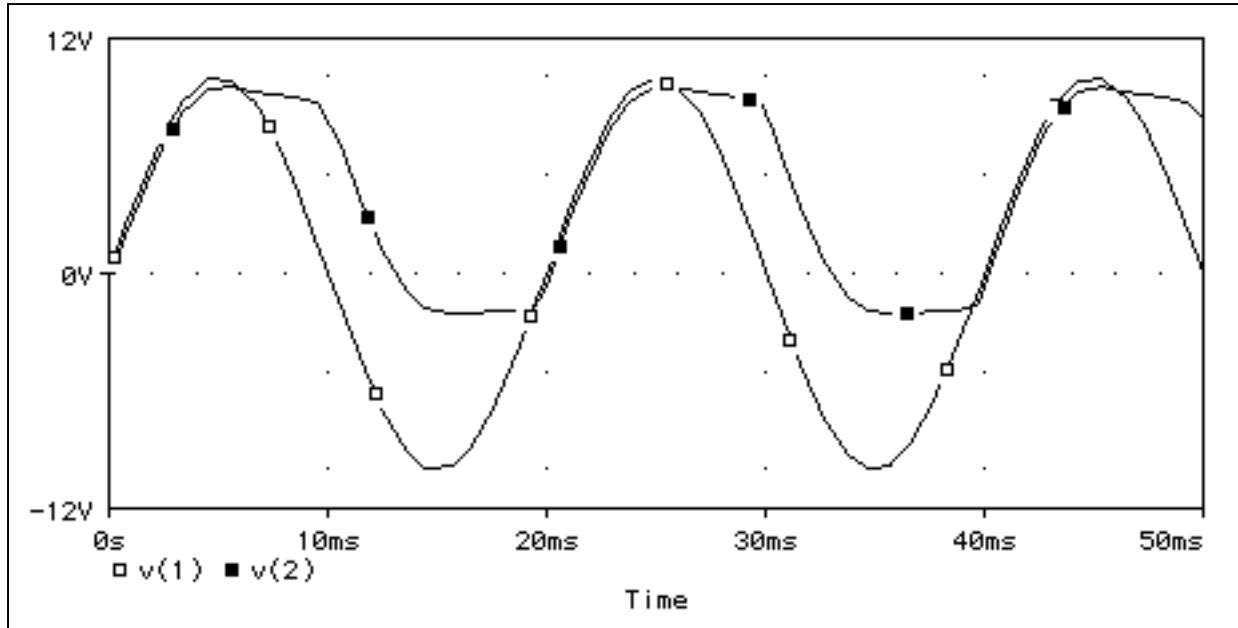


Figura 8.4: diodo con tensione di breakdown piccola.

8.2.1 Sintassi del modello per il diodo

La sintassi completa per il modello di un diodo è sintetizzata nella riga

```
.MODEL      DMOD      D      ([PARAMETRI]) ,
```

in cui DMOD rappresenta il particolare nome che vogliamo assegnare al modello di diodo (come al solito, è buona norma che cominci con la lettera D). Un diodo che utilizzi tale modello deve essere rappresentato dalla riga

```
DNAME    NA    NC    DMOD    AREA    OFF    IC=VALUE
```

dove il diodo di nome DNAME è connesso ai nodi NA e NC, rispettivamente anodo e catodo, ed è di tipo DMOD. La variabile opzionale AREA è il fattore di scala per la grandezza del diodo ed è interpretabile come il numero di diodi connessi in parallelo (per ‘default’ AREA=1); OFF è una voce opzionale che, se presente, inizializza il diodo con una polarizzazione inversa ai fini del calcolo della polarizzazione stessa, altrimenti il simulatore inizia il calcolo assegnando il valore di polarizzazione diretta di 0.6 V alla tensione tra anodo e catodo. Il suffisso opzionale IC ovviamente stabilisce la tensione da usare nel calcolo del transitorio se presente una riga ‘.TRAN’.

La seguente tabella riporta i principali parametri a disposizione per rappresentare un diodo e i valori assunti per ‘default’.

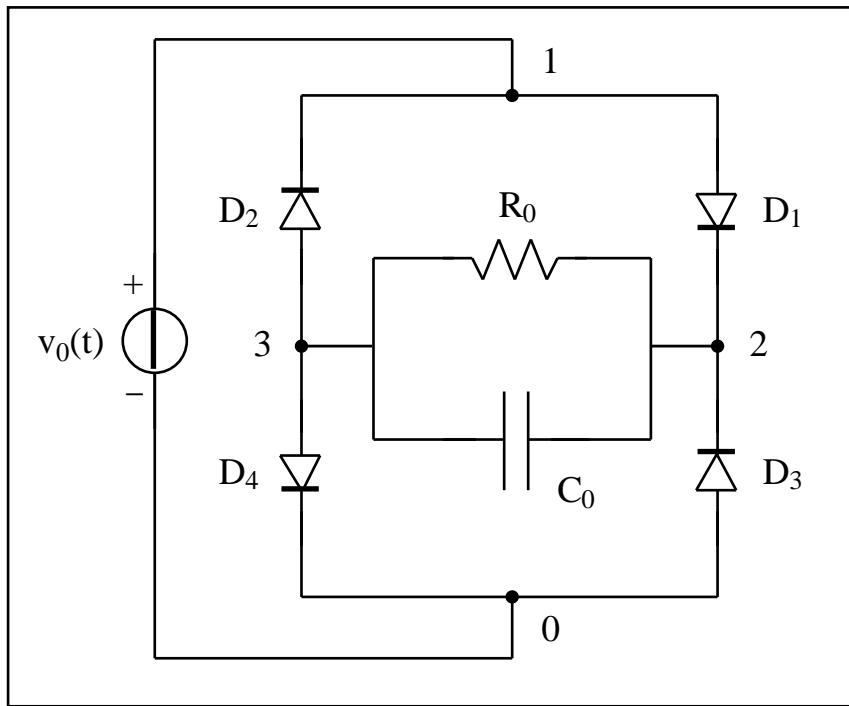
Nome	Parametro	Unità	Default	Fattore di scala
IS	Corrente di saturazione	A	10^{-14}	area
N	Coefficiente di emissione	-	1	-
RS	Resistenza serie		0	1/area
TT	Tempo di transito	s	0	-
CJ0	Capacità di giunzione per polarizzazione nulla	F	0	area
VJ	Potenziale di giunzione	V	1	-
M	Coefficiente di ‘grading’	-	0.5	-
EG	Energia di attivazione	eV	1.11	-
BV	Tensione di Breakdown	V		-
IBV	Corrente alla tensione di Breakdown	A	0.001	area

Come si può notare dalla tabella, alcuni parametri sono proporzionali al fattore AREA, altri inversamente proporzionali. Ad esempio, la capacità di giunzione è direttamente proporzionale all’area, quindi se $CJ0 = 2 \text{ pF}$ e $\text{AREA} = 3$ allora il simulatore, per CJ0, adotterà il valore $2 \text{ pF} \cdot 3 = 6 \text{ pF}$. EG rappresenta l’energia di attivazione della barriera dei due materiali che compongono il diodo, ad esempio per il silicio vale 1.11 eV, per il germanio 0.67 eV e per un diodo Schottky vale 0.69 eV.

8.2.2 Un esempio: il ponte di Graetz

Come esempio di utilizzo in Spice dei diodi, parleremo di una classica applicazione: il ponte di Graetz, tipicamente adoperato per raddrizzare la tensione sinusoidale e fornire una tensione continua in uscita.

In Figura 8.5 è riportato lo schema di tale circuito in cui C_0 è la capacità di livellamento e R_0 rappresenta il carico da alimentare in continua. Il listato Spice di questo circuito è qui di seguito riportato.

**Figura 8.5:** ponte di Graetz a diodi.**Esempio 8.3**

* Ponte di Graetz

```

.MODEL      DMOD1    D
D1   1   2   DMOD1
D2   3   1   DMOD1
D3   0   2   DMOD1
D4   3   0   DMOD1
C0   2   3   470u
R0   2   3   50
V0   1   0   SIN(0 10 50 0 0)
.TRAN     0.01m   0.6
.PROBE
.END

```

Notate la presenza del generatore sinusoidale il cui parametro di attenuazione, come spesso abbiamo fatto, è stato posto a zero. Una semplice analisi in frequenza (per piccoli segnali) non sarebbe stata molto utile per i nostri scopi dal momento che siamo interessati al transitorio iniziale più che agli sfasamenti relativi di regime tra le diverse grandezze della rete.

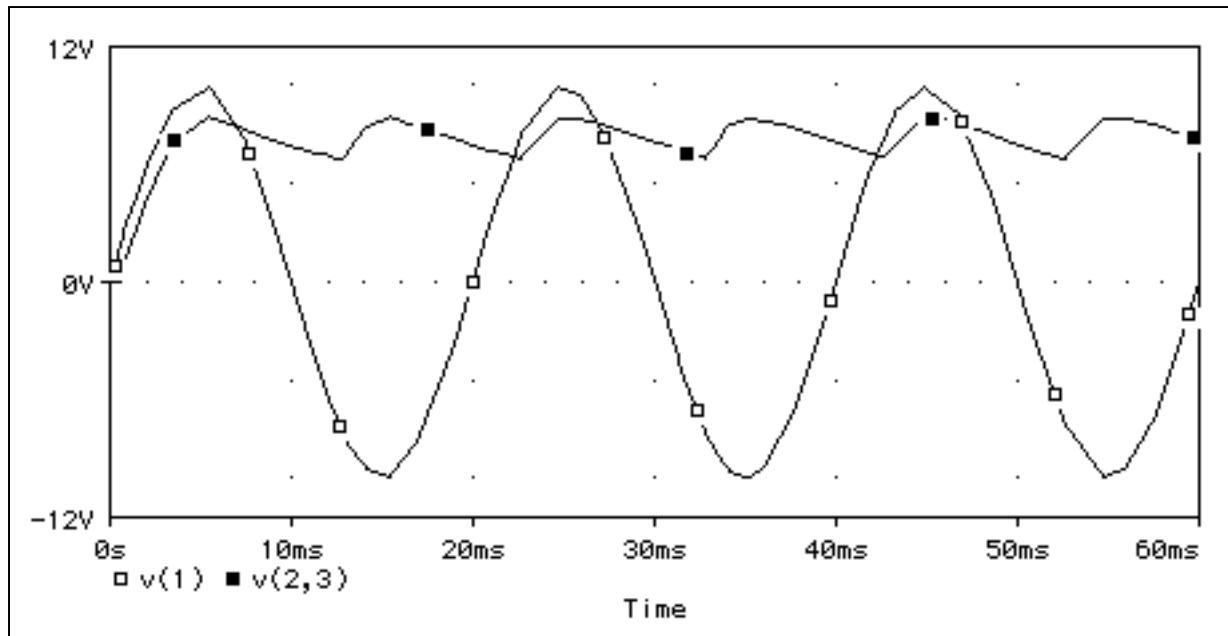


Figura 8.6: tensione raddrizzata in uscita al ponte.

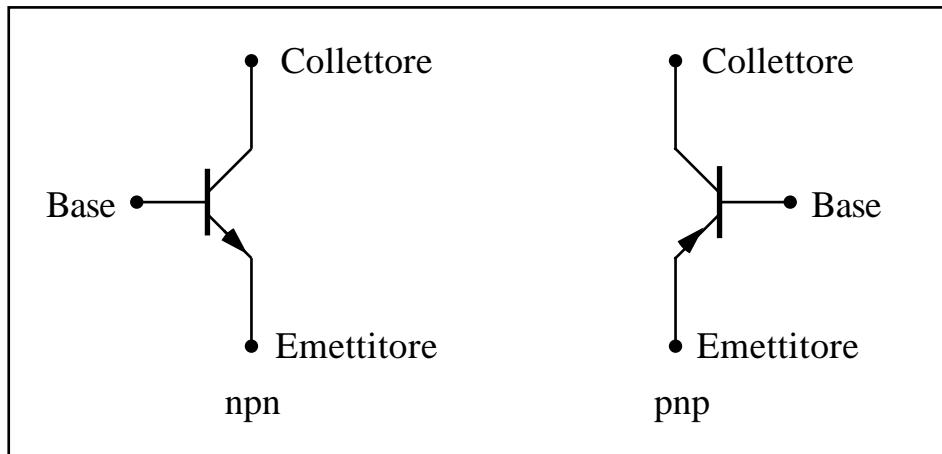
In Figura 8.6 possiamo notare la tensione di uscita $v(2,3)$ con la tipica ondulazione (ripple) dovuta al fatto che il condensatore si carica quando la semionda di alimentazione è, in valore assoluto, maggiore e si scarica sul resistore negli altri istanti. Questo discorso vale per entrambe le semionde grazie alla presenza dei quattro diodi.

8.3 Il transistore

Il transistore è un dispositivo a semiconduttore a tre morsetti. Il principio di funzionamento si basa sul fatto che la corrente che scorre tra due morsetti (detti base ed emettitore) controlla la corrente, molto più elevata, che fluisce nel terzo morsetto (detto collettore). Da ciò discende che, il transistore può essere visto come un amplificatore: la corrente nel collettore segue la forma della corrente che circola tra la base e l'emettitore, mentre in ampiezza risulta amplificata.

In maniera semplificata, può essere visto anche come un interruttore controllato: la corrente nel collettore passa dal valore ‘basso’ a un valore ‘alto’ a seconda della tensione di controllo. È questa una delle funzioni base dei circuiti digitali.

I simboli dei transistori del tipo ‘npn’ e ‘pnp’ sono riportati in Figura 8.7: la piccola differenza è costituita dalla freccia sull’emettitore.

**Figura 8.7:** simboli adoperati per i transistori.

Nel paragrafo 5.2.1 abbiamo simulato il circuito equivalente per piccoli segnali di un transistore utilizzando i generatori dipendenti; ecco, invece, un semplice elenco Spice che determina le caratteristiche di uscita di un transistore reale di cui specifichiamo il modello mediante un istruzione .MODEL.

Esempio 8.4

* Caratteristiche di uscita di un transistore

```
.MODEL QMOD1 NPN (IS=10f, BF=100, RC=0.7, RB=10)
Q1 1 2 0 QMOD1
R1 1 3 10
R2 2 4 100
IB 0 4 DC 0
VCE 3 0 DC 0
.DC VCE 0 10 0.5 IB 0 0.4m 0.1m
.PROBE
.END
```

Effettuiamo una doppia analisi DC in cui varia la tensione tra il collettore e l'emettitore e la corrente di base. In Figura 8.8 possiamo notare la corrente di collettore in funzione della tensione VCE. Ogni curva corrisponde a un valore della corrente di base: partendo dal basso, la prima curva, che coincide con l'asse delle ascisse, corrisponde a $IB = 0$ A, la seconda a $IB = 0.1$ mA, l'ultima a $IB = 0.4$ mA.

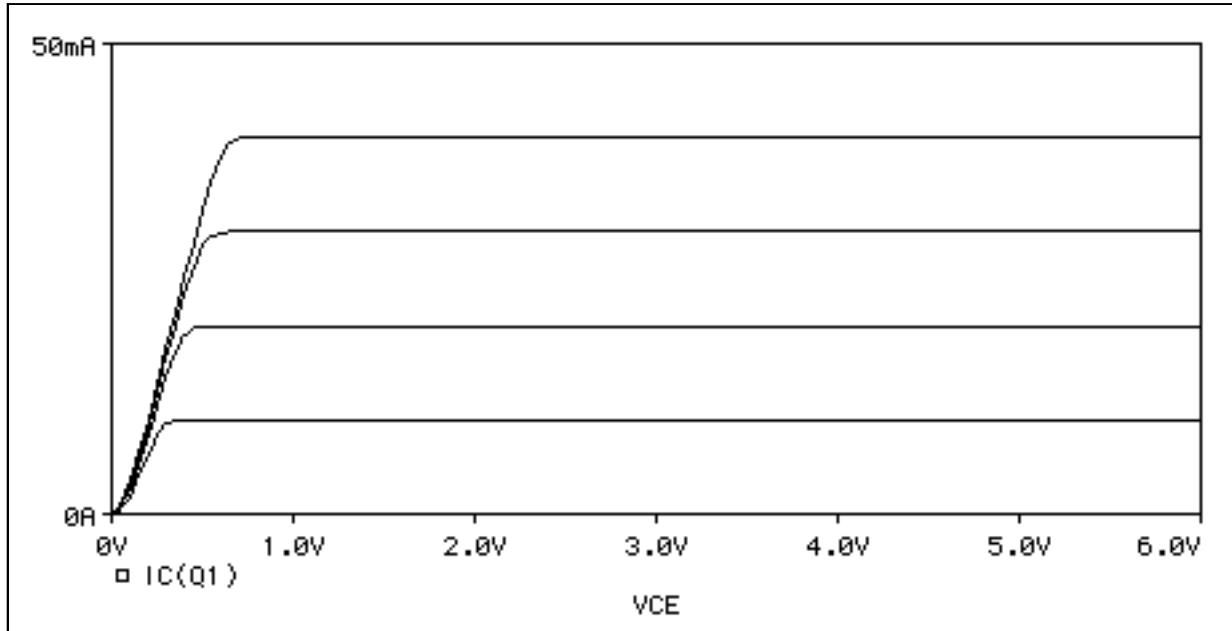


Figura 8.8: caratteristiche di uscita di un transistore tipo NPN.

8.3.1 Sintassi del modello per il transistore

La riga che introduce direttamente il modello più generale per un transistore è del tipo

```
.MODEL      MYNAME    NPN([PARAMETRI]).
```

Il comando .MODEL comunica a Spice che la riga è dedicata all'introduzione di un nuovo modello di nome MYNAME; il terzo campo 'NPN' indica che il modello riguarda un transistore di tipo NPN (un transistore di tipo PNP è individuato dal campo 'PNP') e tra parentesi sono indicate i parametri caratteristici del modello e, ovviamente, ogni parametro ha un suo valore di 'default'.

La seguente tabella riporta i principali parametri a disposizione per rappresentare un transistore e i valori assunti per 'default'.

Nome	Parametro	Unità	Default	Fattore di scala
IS	Corrente di saturazione	A	10^{-16}	area
BF	Guadagno di corrente diretta	-	100	-
BR	Guadagno di corrente inverso	-	1	-
NF	Coefficiente di emissione diretta	-	1	-
NR	Coefficiente di emissione inverso	-	1	-
VAF	Tensione di Early diretta	V		-
VAR	Tensione di Early inversa	V		-

RC	Resistenza di collettore		0	1/area
RE	Resistenza di emettitore		0	1/area
RB	Resistenza di base		0	1/area
TF	Tempo di transito diretto	s	0	-
TR	Tempo di transito inverso	s	0	-
CJE	Capacità base - emettitore a polarizzazione nulla	F	0	area
VJE	Potenziale di giunzione base emettitore	V	0.75	-
CJC	Capacità base - collettore a polarizzazione nulla	F	0	area
VJC	Potenziale di giunzione base collettore	V	0.75	-

Alcuni parametri sono proporzionali al fattore AREA, altri inversamente proporzionali. Tra gli altri, le resistenze di collettore, base e emettitore RC, RB e RE rispettivamente rappresentano le dissipazioni interne del dispositivo. Poi, il fatto che il transistore ha una resistenza d'uscita non nulla, per cui si comporta come un generatore non ideale, è rappresentato dai due parametri VAF e VAR, tensioni di Early diretta e inversa; queste tensioni rappresentano l'effetto precedentemente esposto e producono una pendenza finita e positiva per le caratteristiche di uscita nella zona di saturazione. Infine, i tempi di transito e le capacità di giunzione permettono di realizzare le limitazioni nella risposta in frequenza del dispositivo. Come esempio, tracciamo le caratteristiche di uscita del precedente transistore, nel cui modello abbiamo posto VAF = 100. In Figura 8.9 possiamo notare come le caratteristiche stesse presentino una pendenza finita nella zona di saturazione.

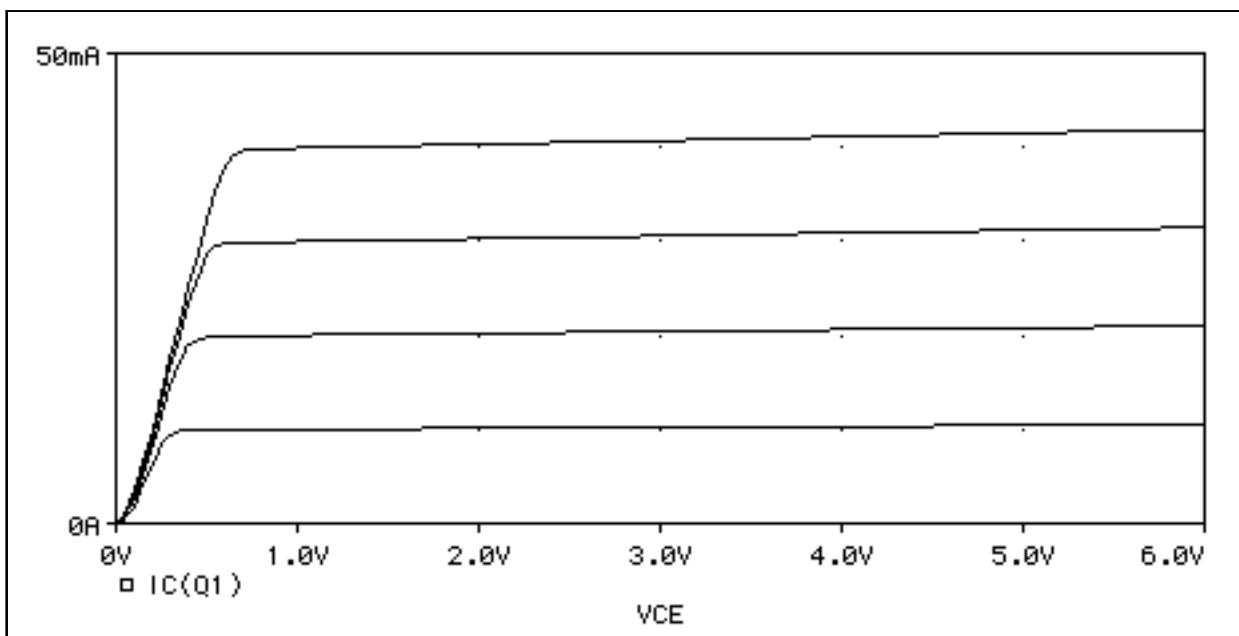


Figura 8.9: effetto Early nelle caratteristiche di uscita di un transistore NPN.

La forma generale per richiamare un transistore all'interno di un listato Spice è

QNAME NC NB NE NS QMOD AREA OFF IC=VBE0, VCE0 ,

in cui la prima lettera del nome QNAME per un BJT deve essere sempre la ‘Q’ e può essere seguita, al solito, da un massimo di sette caratteri; NC, NB, NE e NS identificano rispettivamente i nodi cui sono connessi il collettore, la base, l’emettitore e il morsetto di substrato (questo morsetto può essere omesso, oppure connesso al nodo, tra quelli di collettore, base e emettitore, posto a potenziale più basso. Esso è comunque ininfluente per il modello statico); QMOD è il nome del particolare modello, definito in precedenza mediante un istruzione .MODEL, che si vuole utilizzare; AREA rappresenta un fattore di scala e indica il numero di transistori in parallelo (il suo valore di ‘default’ è 1). Il termine opzionale OFF inizializza il transistore come se fosse polarizzato inversamente, altrimenti, ai fini del calcolo della polarizzazione, il simulatore considera $V_{BE} = 0.6$ V e $V_{BC} = -1.0$ V. Infine, IC definisce le tensioni di giunzione VBE0 e VCE0 all’istante iniziale da considerare nell’analisi in transitorio del dispositivo.

8.3.2 Un esempio: alimentatore stabilizzato

Sfruttando le caratteristiche di funzionamento di un transistore BJT è possibile realizzare un semplice alimentatore stabilizzato in grado di ‘rispondere’ a eventuali disturbi sull’alimentazione e/o sul carico. Allo scopo, consideriamo lo schema semplificato di Figura 8.10: in esso la tensione continua $V(2)$ viene stabilizzata mediante il semplice circuito costituito dal transistore la cui tensione di controllo è stabilizzata mediante un diodo zener. Lo schema è stabile in quanto mediante il transistore e il diodo si realizza una retroazione che bilancia i disturbi.

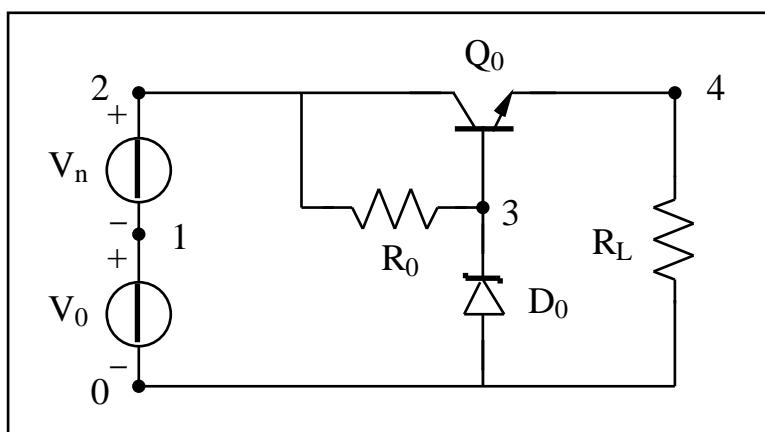


Figura 8.10: schema di alimentatore stabilizzato con transistore serie.

Il comportamento di tale schema può essere simulato in Spice dal seguente listato.

Esempio 8.5

* Alimentatore stabilizzato con transistore tipo 2N3904

```
.model Q2N3904 NPN(Is=6.734f Xti=3 Eg=1.11 Vaf=74.03 Bf=416.4 Ne=1.259
+ Ise=6.734f Ikf=66.78m Xtb=1.5 Br=.7371 Nc=2 Isc=0 Ikr=0 Rc=1
+ Cjc=3.638p Mjc=.3085 Vjc=.75 Fc=.5 Cje=4.493p Mje=.2593 Vje=.75
+ Tr=239.5n Tf=301.2p Itf=.4 Vtf=4 Xtf=2 Rb=10)
*
.model D1N750 D(Is=880.5e-18 Rs=.25 Ikf=0 N=1 Xti=3 Eg=1.11 Cjo=175p
+ M=.5516 Vj=.75 Fc=.5 Isr=1.859n Nr=2 Bv=4.7 Ibv=20.245m Nbv=1.6989
+ Ibvl=1.9556m Nbvl=14.976 Tbv1=-21.277u)

Vn    2    1    SIN(0 200m 100 0 0)
V0    1    0    PWL(0 0 0.001 9 0.05 9 0.051 7)
Q0    2    3    4    Q2N3904
D0    0    3    D1N750
R0    2    3    150
RL    4    0    10
.TRAN    0.1m    0.1    UIC
.PROBE
.END
```

All'inizio del listato sono presenti le descrizioni del transistore 2N3904 tipo NPN e del diodo zener 1N750 con $V_z = 4.7$ V. L'alimentazione in continua, che possiamo immaginare essere fornita dal ponte di Graetz dell'esempio precedente, ha un valore di 9 V e vede sovrapposta una tensione di ripple di 200 mV che rappresenta il residuo di alternata che non viene filtrato dal ponte di Graetz. Dopo 100 ms, un disturbo fa in modo che la tensione di alimentazione scenda a 7 V. Il carico è costituito da un resistore di 10 .

In Figura 8.11 si nota che l'uscita presenta già un ripple notevolmente ridotto rispetto all'ingresso in continua e il disturbo di ben 2 V provoca un effetto notevolmente minore sulla tensione del carico. Si noti che la tensione di uscita è minore di 4.7 V perché a questo valore bisogna sottrarre la caduta di tensione sulla giunzione base emettitore.

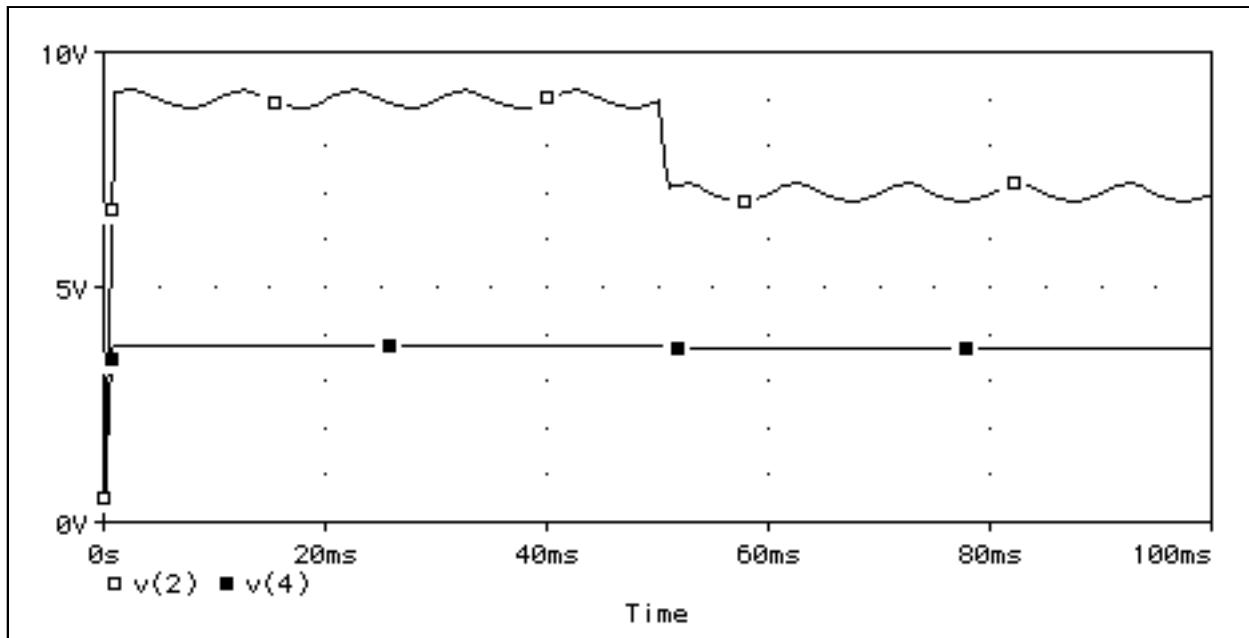


Figura 8.11: tensioni in ingresso e in uscita all'alimentatore.

Notiamo inoltre che quello mostrato è soltanto uno schema di principio in quanto lo schema reale dovrebbe presentare come minimo una capacità di livellamento in uscita.

Infine notiamo che i componenti usati sono presenti nella libreria standard ‘EVAL.LIB’ fornita con la versione evaluation di PSpice, per cui le definizioni dei componenti stessi potevano essere sostituite dall’istruzione

```
.LIB      EVAL.LIB
```

che rende ‘visibili’ al nostro file tutti i componenti presenti nella libreria. In generale, una libreria di nome MYLIB.LIB sarà inclusa, in maniera analoga, mediante l’istruzione

```
.LIB      MYLIB.LIB .
```

8.4 I transistor a effetto di campo: il jfet e il mosfet

Il transistore a giunzione ad effetto di campo, anche detto jfet (o transistore unipolare), è un dispositivo a tre morsetti concettualmente simile al transistore, in cui la tensione applicata a due morsetti controlla la corrente che scorre nel terzo morsetto. Per questo il jfet può essere usato come amplificatore e come interruttore in maniera analoga al transistore. Il transistore ad effetto di campo deriva il suo nome dal suo principio di funzionamento: il meccanismo di controllo

della corrente è basato sull'effetto che produce la tensione applicata al morsetto di controllo: essa produce una sorta di canale di portatori in cui può scorrere la corrente di uscita e in questo canale è presente un solo tipo di portatori, elettroni oppure 'lacune' che parteciperanno al processo di conduzione. Per questo motivo il jfet sarà detto a canale 'n' o 'p'. Inoltre, si fa notare che, per le caratteristiche stesse del dispositivo, il morsetto di controllo assorbe una corrente piccolissima dal generatore di tensione di ingresso.

Il vantaggio principale nell'uso dei jfet è che questi componenti possono essere realizzati con dimensioni molto piccole e che si prestano alla costruzione di circuiti integrati su larga scala.

I mosfet sono concettualmente simili ai jfet ma presentano una giunzione metallo-ossido-semiconduttore che esalta ancor di più le doti dei jfet. Essi trovano applicazione sia nei circuiti integrati come interruttori, sia come amplificatori di segnale.

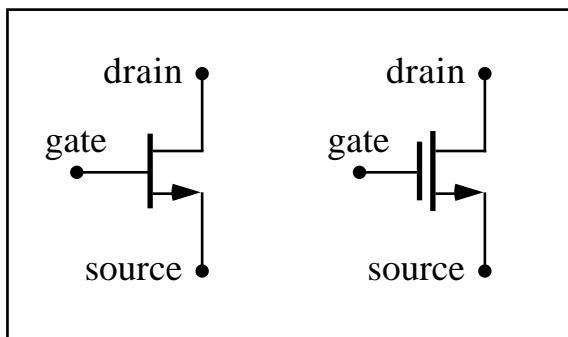


Figura 8.12: simboli del jfet e del mosfet a canale n.

In Figura 8.12 sono mostrati i simboli circuituali per un jfet e un mosfet a canale n. Il morsetto di controllo è chiamato 'gate' che vuol dire 'porta', mentre gli altri due morsetti sono chiamati 'drain' e 'source' che significano rispettivamente 'collettore' e 'sorgente'.

Passiamo ora alla sintassi che permette la simulazione di questi dispositivi in Spice. La riga che introduce direttamente il modello più generale per un jfet è del tipo

```
.MODEL      MYNAME    NJF    ([PARAMETRI]).
```

Il comando .MODEL comunica a Spice che la riga è dedicata all'introduzione di un nuovo modello di nome MYNAME; il terzo campo 'NJF' indica che il modello riguarda un transistore a effetto di campo a canale n, per uno a canale p bisogna usare 'PJF' e tra parentesi sono indicate i parametri caratteristici del modello.

La seguente tabella riporta i principali parametri a disposizione per rappresentare un jfet e i valori assunti per ‘default’.

Nome	Parametro	Unità	Default	Fattore di scala
IS	Corrente di saturazione (giunzione di gate)	A	10^{-14}	-
PB	Potenziale della giunzione di gate	V	1	-
VTO	Tensione di soglia (pinch-off)	V	- 2.0	-
BETA	Parametro di trans-conduttanza	AV ⁻²	10^{-4}	area
LAMBDA	Modulazione della lunghezza di canale	V ⁻¹	0	-
RD	Resistenza di drain		0	1/area
RS	Resistenza di source		0	1/area
CJS	Capacità gate source a polarizzazione nulla	F	0	area
CJD	Capacità gate drain a polarizzazione nulla	F	0	area

Come si può notare dalla tabella, alcuni parametri sono proporzionali al fattore AREA, altri inversamente proporzionali.

La forma generale per richiamare un jfet all’interno di un listato Spice è

```
JNAME ND NG NS JMOD   AREA      OFF      IC=VDS0, VGS0 ,
```

in cui la prima lettera del nome JNAME per un jfet deve essere sempre la ‘J’ e può essere seguita, al solito, da un massimo di sette caratteri; ND, NG e NS identificano rispettivamente i nodi cui sono connessi il drain, gate e il source; JMOD è il particolare modello che si vuole utilizzare, mentre AREA rappresenta un fattore di scala e indica il numero di transistori in parallelo (il suo valore di default è 1). Il termine opzionale OFF inizializza il transistore come se fosse polarizzato inversamente, altrimenti, ai fini del calcolo della polarizzazione, il simulatore considera VGS = VTO che è la tensione di soglia e VDS = 0 V. Il parametro IC definisce le tensioni di giunzione all’istante iniziale dell’analisi in transitorio ed agisce solo nell’istruzione ‘.TRAN’ quando è presente l’opzione UIC.

La riga che introduce direttamente il modello più generale per un *mosfet* è invece del tipo

```
.MODEL      MYNAME   NMOS    ([PARAMETRI]) .
```

Il comando .MODEL comunica a Spice che la riga è dedicata all'introduzione di un nuovo modello di nome MYNAME; il terzo campo 'NMOS' indica che il modello riguarda un transistore mosfet a canale n, per uno a canale p bisogna scrivere 'PMOS' e tra parentesi sono indicate i parametri caratteristici del modello.

Bisogna dire che il modello di mosfet di Spice, vista l'importanza del dispositivo, è piuttosto complesso e permette una simulazione accurata addirittura con quattro livelli (LEVEL) a disposizione di cui il primo (LEVEL=1) permette solo un'analisi in continua. Forniremo, pertanto, nella tabella seguente solo i principali parametri per rappresentare un mosfet e i relativi valori assunti per 'default', rimandando ai manuali forniti con le varie versioni del programma per una più approfondita descrizione.

Nome	Parametro	Unità	Default	Fattore di scala
IS	Corrente di saturazione (giunzione di bulk)	A	10^{-14}	-
PB	Potenziale della giunzione di bulk	V	1	-
VTO	Tensione di soglia	V	0	-
KP	Fattore di trans-conduttanza	AV^{-2}	$2.0 \cdot 10^{-5}$	-
LAMBDA	Modulazione della lunghezza di canale	V^{-1}	0	-
RD	Resistenza di drain		0	1/area
RS	Resistenza di source		0	1/area
CJS	Capacità gate source a polarizzazione nulla	F	0	area
CJD	Capacità gate drain a polarizzazione nulla	F	0	area

Come si può notare dalla tabella, alcuni parametri sono proporzionali al fattore AREA, altri inversamente proporzionali.

La forma generale per richiamare un mosfet all'interno di un listato Spice è:

MYNAME ND NG NS NB MMOD AREA OFF IC=VDS0, VGS0 ,

dove la prima lettera del nome MYNAME per un mosfet deve essere sempre la 'M' e può essere seguita, al solito, da un massimo di sette caratteri; ND, NG e NS identificano rispettivamente i nodi cui sono connessi il drain, gate e il source; NB specifica il nodo cui è connesso il substrato del dispositivo; MMOD è il particolare modello che si vuole utilizzare, mentre AREA rappresenta un fattore di scala e indica il numero di transistori in parallelo (il suo valore di default è 1). Il termine opzionale OFF inizializza il transistore come se fosse polarizzato inversamente,

altrimenti, ai fini del calcolo della polarizzazione, il simulatore considera $VGS = VTO$ che è la tensione di soglia e $VDS = 0$ V. Il parametro IC definisce le tensioni di giunzione all'istante iniziale da considerare nell'analisi in transitorio.

8.4.1 Un esempio: amplificatore CMOS

Per applicare i modelli di transistori ad effetto di campo, illustreremo un esempio di amplificatore CMOS che agisce da invertitore, l'elemento base dell'elettronica digitale.

Con il termine ‘tecnologia CMOS’ (Complementary MOS) si indica tutta la produzione di circuiti elettronici digitali, spesso integrati, che fa uso di transistor ad effetto di campo di entrambi i tipi ‘P’ e ‘N’, al contrario della tecnologia NMOS (PMOS) che fa uso solo di dispositivi N (P).

In Figura 8.13 è riportato lo schema dell'amplificatore CMOS. Il MOS a canale N è l'amplificatore vero e proprio che agisce, in particolare, da invertitore. Questo MOS è polarizzato mediante uno schema ben noto in elettronica e detto ‘a specchio di corrente’ e costituito dai due MOS a canale P e dalla resistenza R_0 che stabilisce la corrente di polarizzazione. Il generatore $v_1(t)$ costituisce il segnale di ingresso, mentre l'uscita è prelevata al nodo 3, ovvero sul drain del MOS a canale N.

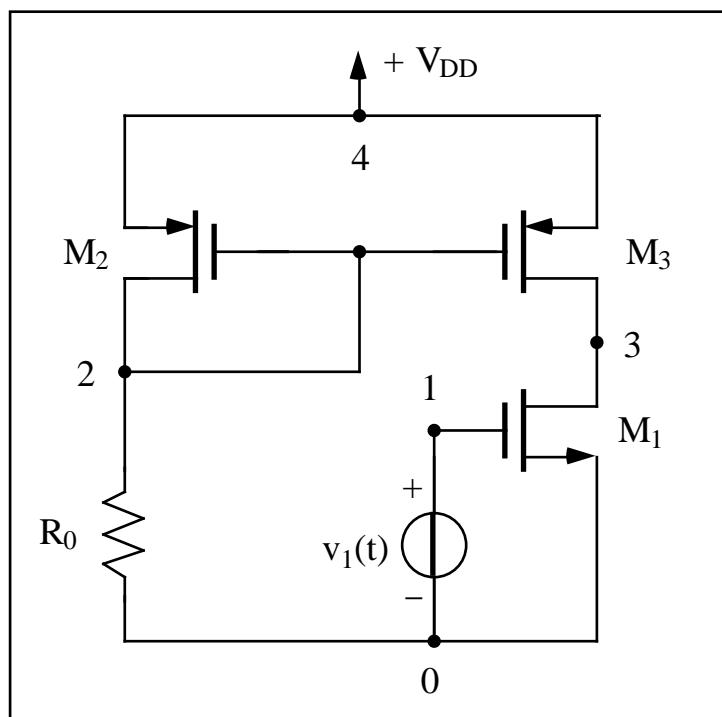


Figura 8.13: amplificatore CMOS con specchio di corrente.

Di seguito riportiamo il listato Spice che simula questo circuito.

Esempio 8.6*** Amplificatore CMOS*********.MODEL MN NMOS(LEVEL=2, VTO=1, KP=20u, LAMBDA=0.04)****.MODEL MP PMOS(LEVEL=2, VTO=-1, KP=20u, LAMBDA=0.04)*******

M1 3 1 0 0 MN L=5u W=125u

M2 2 2 4 4 MP L=5u W=125u

M3 3 2 4 4 MP L=5u W=125u

R0 2 0 10k

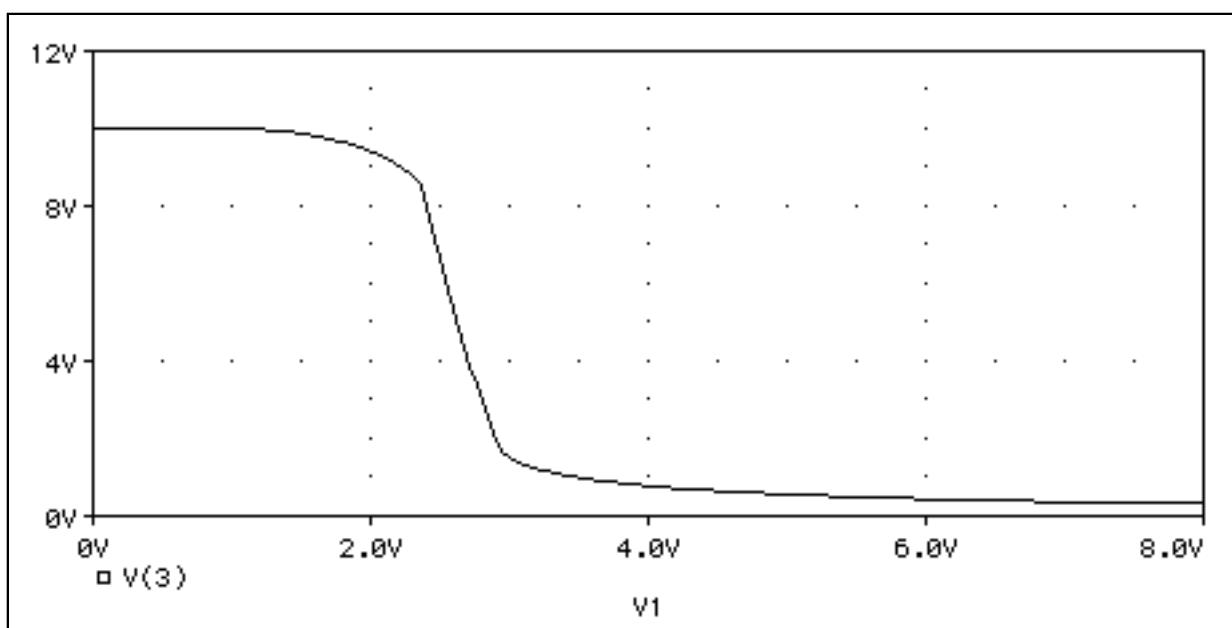
VDD 4 0 DC 10

V1 1 0 DC 1

.DC V1 0 8 0.02

.PROBE**.END**

Notiamo che nelle righe di programma dove vengono dichiarati i MOS, sono specificati anche i morsetti relativi al substrato e questi sono sempre collegati ai rispettivi source; inoltre, le dimensioni dei MOS sono identiche sia per il canale N che per il canale P. Effettuiamo un'analisi DC per ricavare la caratteristica di trasferimento del dispositivo. In Figura 8.14 possiamo vedere il grafico che ci fornisce Probe.

**Figura 8.14:** caratteristica di trasferimento dell'amplificatore CMOS.

Notate il comportamento da invertitore per cui a un segnale basso in tensione in ingresso corrisponde un segnale alto in uscita e, viceversa, a un segnale alto in tensione in ingresso corrisponde un segnale basso in ingresso. Questi due comportamenti sono separati da una zona di incertezza che è breve e abbastanza ripida.

8.5 Come ottenere i parametri Spice di un dispositivo

Nei paragrafi precedenti ci siamo resi conto che nell'analisi di una rete elettrica, oppure nella sintesi di un circuito, è importante avere a disposizione dei buoni modelli, con i parametri giusti, dei componenti a semiconduttore da simulare.

A tal scopo, la prima strada da percorrere è verificare se i componenti di interesse sono già presenti nella libreria EVAL.LIB, fornita con le versioni ‘evaluation’ di Spice, oppure, se abbiamo a disposizione una versione professionale di Spice, se sono presenti nelle numerose librerie fornite a corredo delle versioni commerciali.

Se questa strada non fornisce tutti i modelli che ci occorrono, possiamo ricorrere a una ricerca nella rete Internet utilizzandone i numerosi e diffusi ‘motori di ricerca’ a disposizione. Addirittura, sono già presenti in rete dei motori di ricerca espressamente dedicati alla caratterizzazione di componenti elettrici, come quello disponibile sul sito <http://www.datasheetlocator.com> dove, tra l'altro, sono disponibili anche i collegamenti ai siti dei maggiori produttori di semiconduttori.

Nella seguente riportiamo alcuni degli indirizzi Internet disponibili, invitandovi a visitarli anche solo per curiosità; spesso essi forniscono il modello Spice dei componenti di loro produzione.

Produttore	Indirizzo Internet
AMD (Advanced Micro Devices)	http://www.amd.com/
Burr-Brown	http://www.burr-brown.com/
Eupec GmbH	http://www.eupec.de/
Fairchild Semiconductors	http://www.fairchildsemi.com/
IBM	http://www.chips.ibm.com/
National Semiconductor	http://www.national.com/
Philips Semiconductors	http://www.semiconductors.philips.com/
ST Microelectronics	http://us.st.com/stonline/index.shtml
Siemens	http://www.siemens.de/
Texas Instruments	http://www.ti.com/sc/docs/products/index.htm
Toshiba Semiconductor	http://doc.semicon.toshiba.co.jp

Laddove anche questa strada risultasse infruttuosa, dovreste ricorrere direttamente al ‘data sheet’ del componente che vi fornirà i suoi parametri elettrici e le curve caratteristiche di funzionamento. Bisogna dire, infine, che il passaggio da questi dati al modello Spice non è affatto semplice e si basa molto sul dettaglio dei dati di partenza a disposizione e sulla conoscenza, più approfondita possibile, che voi avete del dispositivo stesso!

8.6 L’amplificatore operazionale

È giunto il momento di introdurre l’amplificatore operazionale e i modelli di questo dispositivo simulabili con Spice. Si tratta di un dispositivo a tre morsetti (a parte le alimentazioni ed eventuali morsetti di controllo) di cui uno è l’uscita e gli altri due sono ingressi. Uno dei due ingressi è invertente, nel senso che l’uscita ha segno opposto rispetto al segnale applicato a questo morsetto, mentre l’altro è non invertente.

Riguardo la simulazione con Spice, partiremo da un modello ideale per poi rimuovere man mano le non idealità per giungere al modello reale che rispecchia la costruzione vera e propria con resistori, diodi e transistori.

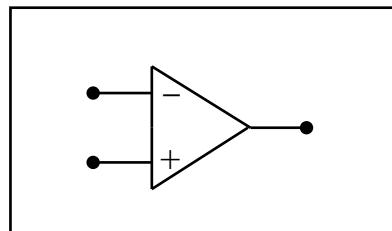


Figura 8.15: simbolo dell’amplificatore operazionale ideale.

In Figura 8.15 abbiamo rappresentato il simbolo dell’amplificatore operazionale ideale.

8.6.1 L’amplificatore operazionale ideale

Le caratteristiche di un amplificatore operazionale ideale sono:

- resistenza di ingresso infinita e, quindi, assorbimento di corrente nullo dal generatore di segnale in ingresso;
- resistenza di uscita nulla e comportamento equivalente a quello di un generatore di tensione ideale;

- guadagno di tensione ‘ad anello aperto’ infinito.

La dizione ‘ad anello aperto’ sottintende che l’amplificatore operazionale nasce per essere retroazionato e quindi connesso ‘ad anello chiuso’, in modo da presentare guadagno finito.

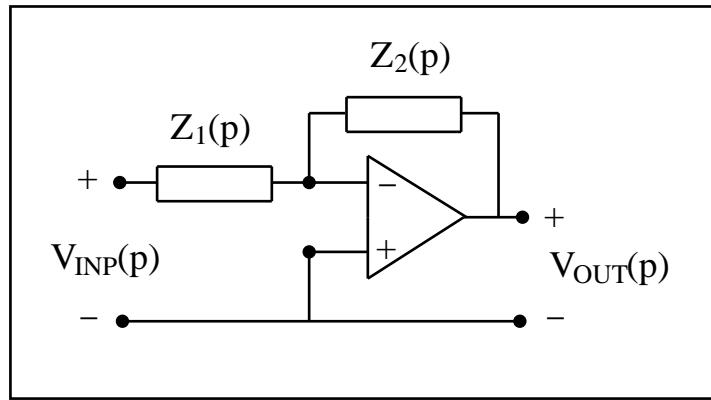


Figura 8.16: configurazione non invertente.

Esempio 8.7

* Circuito integratore con operazionale ideale

*

```
.SUBCKT OPIDE 1 2 3
EG 3 0 1 2 1e6
.ENDS OPIDE
*
Vin 1 0 AC 1m PWL(0 0 0.001 5 1 5 1.001 -5 2 -5)
R1 1 2 1meg
C1 2 3 1u
XOP 0 2 3 OPIDE
.AC LIN 1000 1m 1k
.TRAN 10m 2
.PROBE
.END
```

In Figura 8.16 abbiamo rappresentato una tipica configurazione detta invertente poiché l’ingresso è sul morsetto omonimo. Per questo circuito, operante nel dominio di Laplace, vale la relazione

$$V_{OUT}(p) = -\frac{Z_2(p)}{Z_1(p)} V_{INP}(p),$$

per cui il rapporto tra le due impedenze operatoriali definisce la funzione di trasferimento o, più semplicemente, il guadagno dell'amplificatore. Se l'impedenza $Z_2(p)$ è, ad esempio, quella di un condensatore, il circuito realizza la funzione di integrazione. Al di sotto della Figura 8.16 abbiamo riportato un esempio che realizza questa configurazione.

L'amplificatore operazionale ideale è simulato mediante il sottocircuito OPIDE che contiene un generatore di tensione controllato in tensione, che presenta un'impedenza infinita ai due morsetti di ingresso, mentre il guadagno è fissato a 10^6 che non è infinito ma costituisce un valore adeguato per la maggior parte delle applicazioni. Notate che il generatore di ingresso 'Vin' definisce sia un generatore sinusoidale sia una forma d'onda costante a tratti (utile per l'analisi in transitorio). Il resistore e il condensatore realizzano un circuito con costante di tempo pari a 1 secondo. Effettueremo sia l'analisi in transitorio per un tempo pari a 2 secondi sia quella in frequenza: cominciamo con un transitorio.

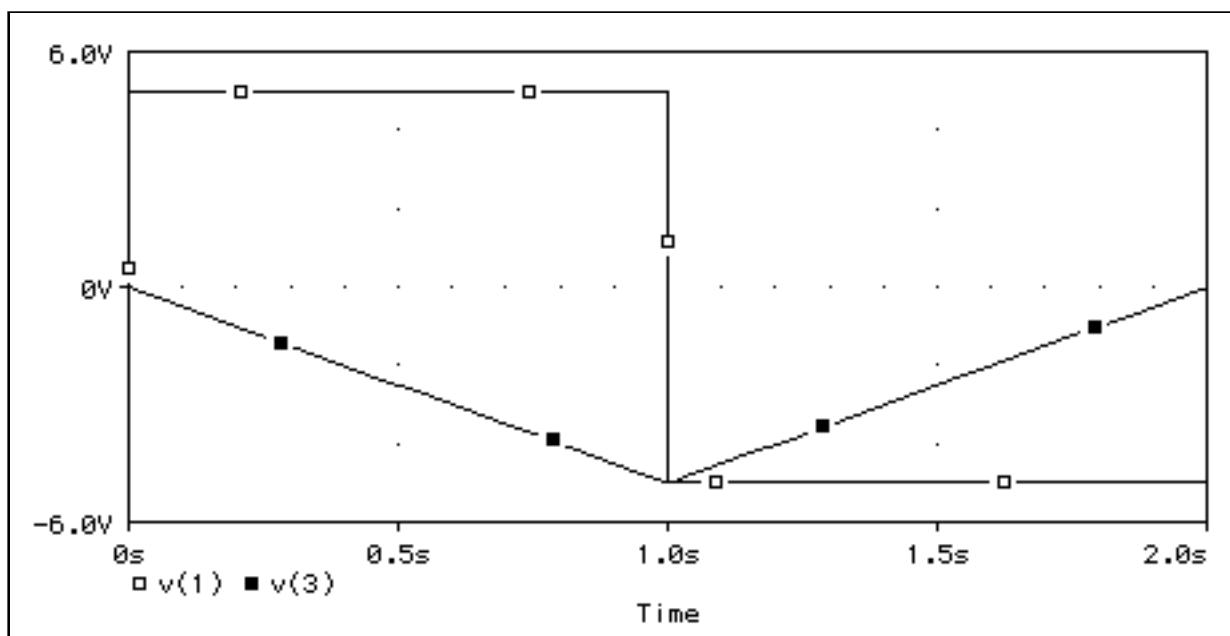


Figura 8.17: analisi in transitorio del circuito integratore.

In Figura 8.17 è mostrata la tensione di ingresso che è costante e pari a 5 V fino a $t = 1 \text{ s}$, dopodiché cambia segno fino all'istante $t = 2 \text{ s}$; la tensione di uscita rappresenta l'integrale nel tempo (a parte l'inversione di segno). Si nota che il comportamento è pressoché ideale. In Figura 8.18 possiamo notare invece il comportamento in frequenza del circuito integratore con il caratteristico 'polo' nell'origine.

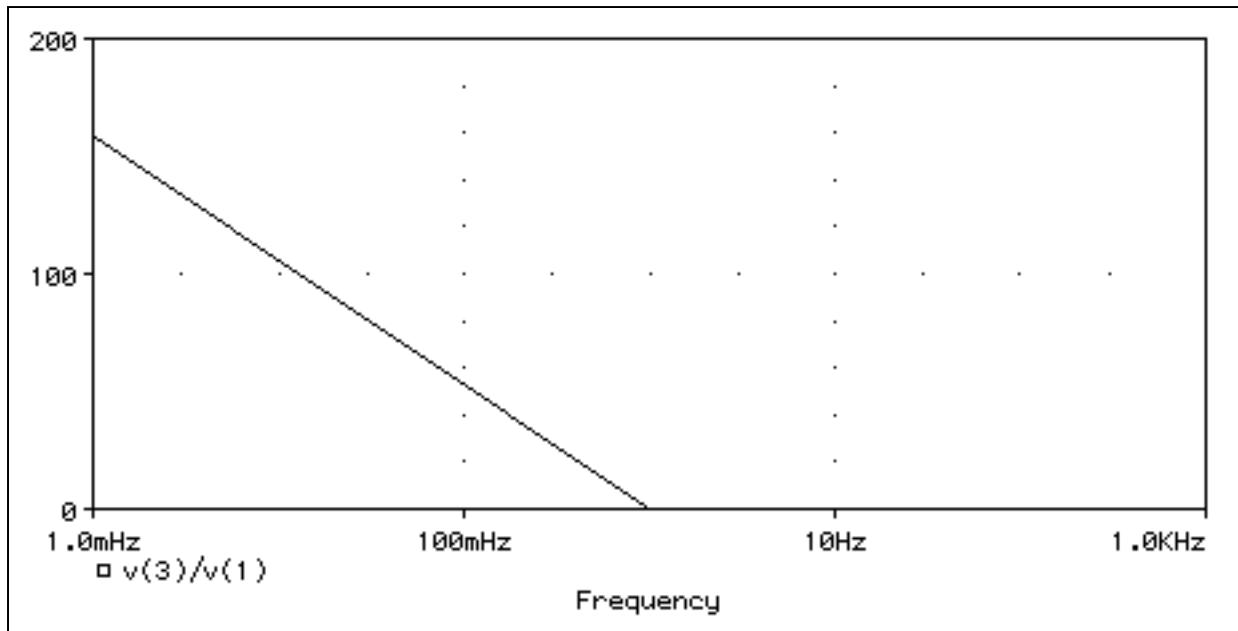


Figura 8.18: analisi in frequenza del circuito integratore.

A commento di questo circuito, diciamo che esso può essere realizzato nella realtà con ottima approssimazione, mentre il circuito derivatore, che in teoria potrebbe essere ottenuto scambiando la posizione della resistenza e della capacità del precedente circuito, trova nella realizzazione pratica notevoli problemi, scontrandosi con la non idealità dei componenti, a partire dall'amplificatore operazionale stesso.

8.6.2 L'amplificatore operazionale reale

Il modello ideale dell'amplificatore operazionale è di grande utilità dal punto di vista dell'insegnamento e per eseguire manualmente dei rapidi calcoli. Però il comportamento reale degli operazionali è comunque lontano da quello ideale e inoltre la simulazione di circuiti complessi con operazionali ideali può dar luogo a problemi di stabilità della simulazione.

Nel modello di amplificatore operazionale si possono includere varie non idealità alcune delle quali molto semplici, come le resistenze di ingresso e uscita di valore finito e la limitazione del valore assoluto dell'uscita a valori minori dell'alimentazione. Altre caratteristiche, come la risposta in frequenza avente una banda finita e il fenomeno dello 'slew rate', sono più difficilmente realizzabili attraverso modelli equivalenti.

Ovviamente la realizzazione del circuito vero e proprio risolve ogni problema di ‘veridicità’ della risposta però appesantisce molto la simulazione perché un solo operazionale può contenere centinaia di componenti per cui, laddove è possibile, bisogna ricorrere a modelli semplificati.

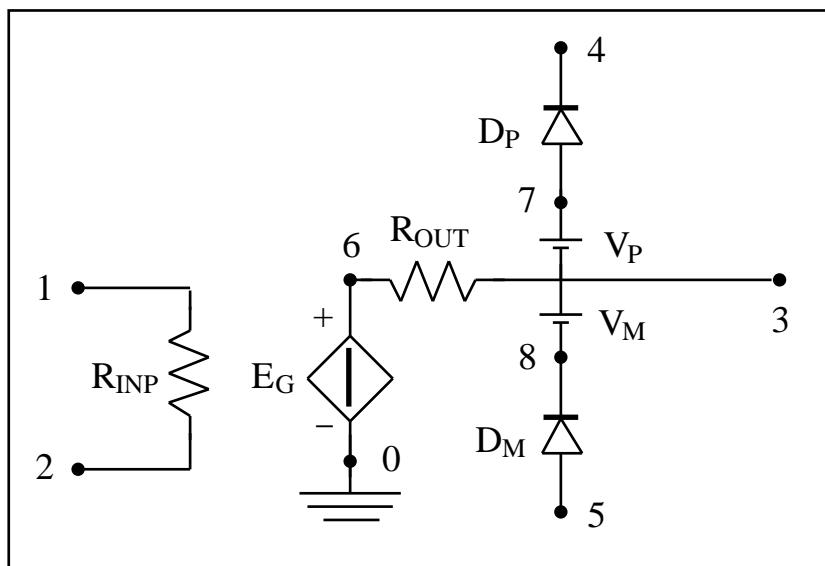


Figura 8.19: modello equivalente per un operazionale reale.

Ad esempio qui di seguito è riportato un sottocircuito che realizza un operazionale con uscita limitata e resistenze di ingresso e uscita finite.

.SUBCKT	OPREA	1	2	3	4	5
*						
Rin	1 2	1meg				
Rout	6 3	10m				
EG	6 0	1 2	1e6			
VP	7 3	1.4				
VM	3 8	1.4				
DP	7 4	DMOD				
DM	5 8	DMOD				
.MODEL	DMOD	D(RS=1n)				
*						
.ENDS	OPREA					

I diodi e le batterie in serie all’alimentazione servono a realizzare la funzione di limitatori per la tensione di uscita; il valore limite è fissato dalle tensioni di alimentazione.

Di seguito riportiamo un semplice circuito amplificatore, in configurazione invertente e con guadagno pari a 2, che utilizza il precedente modello.

Esempio 8.8

* Operazionale reale

*

```
V1      1      0      SIN(0 8 1k 0 0)
R1      1      2      10
R2      2      3      20
RL      3      0      8
X1      0      2      3      4      5      OPREA
VP      4      0      10
VM      5      0      -10
.TRAN   0.4m    2m
.PROBE
.END
```

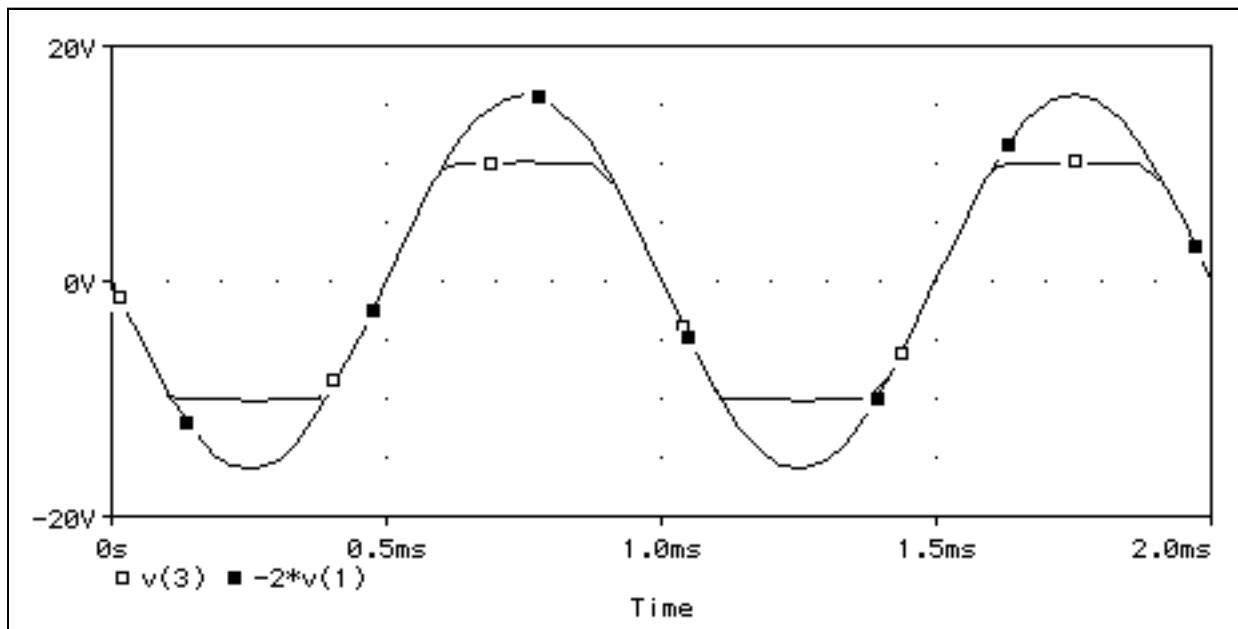


Figura 8.20: uscita di un amplificatore operazionale reale.

In Figura 8.20 sono rappresentati i segnali in gioco del circuito descritto nell'esempio 8.8. Notiamo che l'uscita 'v(3)' dell'amplificatore operazionale risulta distorta e tagliata a 10 V che è il valore dell'alimentazione.

Il segnale '- 2 v(1)' rappresenta, invece, l'uscita che fornirebbe l'amplificatore operazionale se fosse ideale.

8.7 L'oscillatore a ponte di Wien

In questo paragrafo, come esempio di utilizzazione degli amplificatori operazionali in Spice, ci occuperemo dello studio dell'oscillatore a ponte di Wien. Ricordiamo che gli oscillatori sono circuiti elettronici che forniscono un segnale oscillante in uscita la cui frequenza è prefissata dalle caratteristiche stesse del circuito senza necessitare di alcun segnale in ingresso. Essi sono una sorta di generatore di segnale e possono essere utilizzati come ingresso per altri circuiti.

Lo schema di principio di un oscillatore a ponte di Wien è estremamente semplice; si tratta di un amplificatore operazionale connesso in configurazione non invertente, quindi con guadagno $1 + R_2/R_1$; nel percorso di retroazione è inserita una rete RC serie e una parallelo. In Figura 8.21 è visibile lo schema di principio di questo oscillatore.

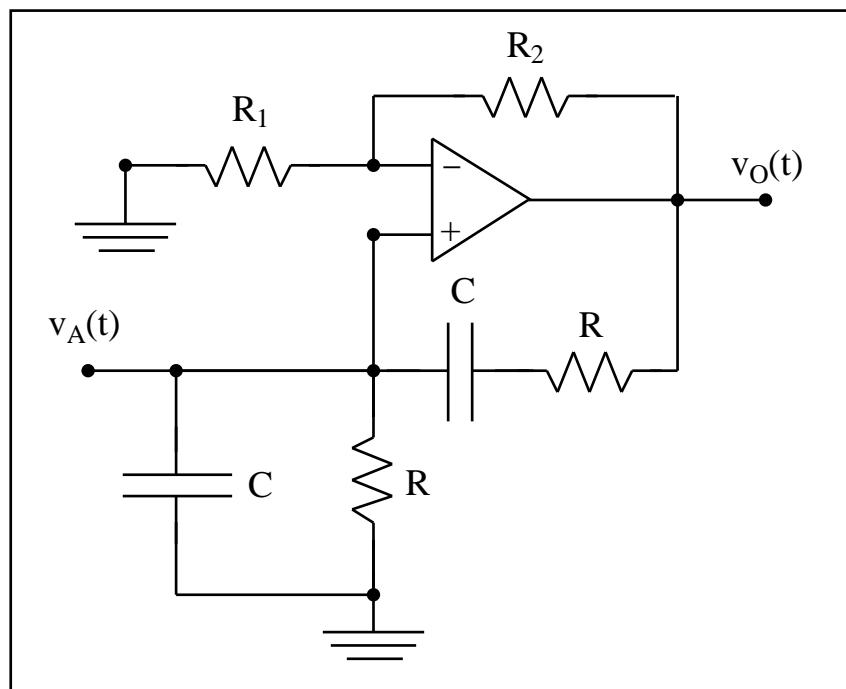


Figura 8.21: schema di principio per l'oscillatore a ponte di Wien.

Il circuito oscillerà alla pulsazione $\omega_0 = 1/RC$ e l'oscillazione si manterrà stabile in ampiezza se $R_2/R_1 = 2$ perché, in tal caso, il guadagno sarà unitario. In realtà, per innescare l'oscillazione è necessario che il guadagno complessivo sia lievemente maggiore dell'unità, per cui si rendono necessari circuiti di stabilizzazione non lineari che all'avvio permettano le oscillazioni e che poi le mantengano stabili.

In Figura 8.22 abbiamo rappresentato una possibile soluzione realizzativa dell'oscillatore. Il potenziometro R_1 deve essere regolato in maniera da innescare

le oscillazioni, poi quando queste aumenteranno in ampiezza, la conduzione dei due diodi farà diminuire la resistenza effettiva tra i nodi 3 e 6 per cui diminuirà il guadagno e si instaurerà un equilibrio. L'ampiezza del segnale di uscita può essere controllata tramite la posizione del potenziometro. Ecco di seguito il listato Spice per simulare il circuito.

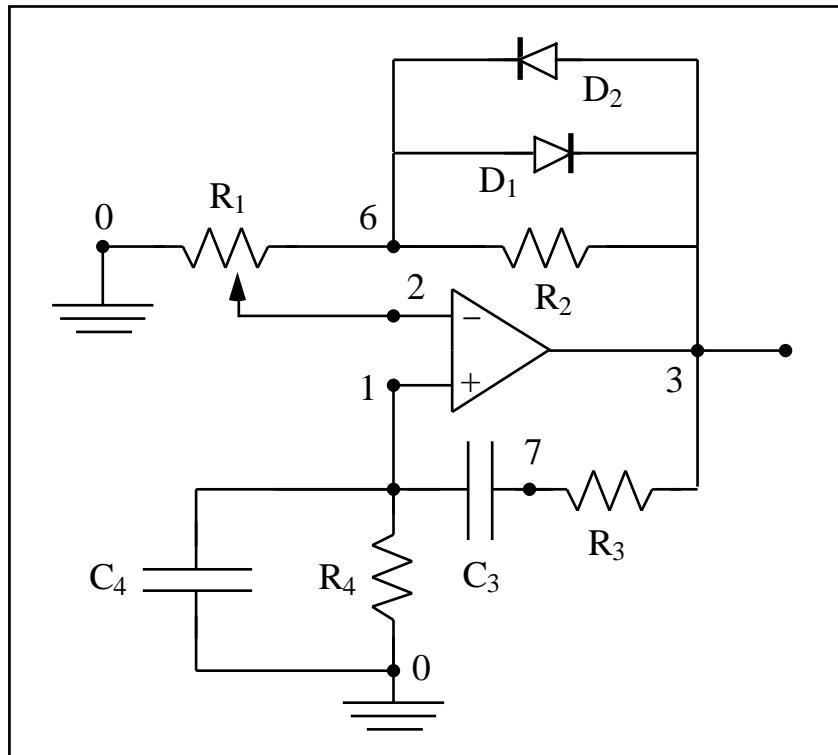


Figura 8.22: schema realizzativo per l'oscillatore a ponte di Wien.

Esempio 8.9

* Oscillatore a ponte di Wien

*

```
.MODEL D1N4148 D(IS=0.1p, RS=16, CJO=2p, TT=12n, BV=100,
IBV=0.1p)
*
.LIB EVAL.LIB
*
VCC 4 0 PWL(0 0 100n 15)
VEE 5 0 PWL(0 0 100n -15)
XAMP 1 2 4 5 3 UA741
D1 6 3 D1N4148
D2 3 6 D1N4148
R1a 2 0 15k
R1b 2 6 35k
```

```

R2    6    3    10k
R3    3    7    10k
C3    7    1    16n
R4    1    0    10k
C4    1    0    16n
*
.OPTIONS ITL5=0
.TRAN    200u    20m    0    200u
.PROBE
.END

```

Il circuito contiene la definizione del diodo 1N4148, poi include la libreria EVAL.LIB in cui è presente il sottocircuito dell'operazionale μ A741. L'alimentazione dell'operazionale è data come 'Piece Wise Linear' con un fronte di salita di 100 ns; questo 'transitorio' è già sufficiente a innescare le oscillazioni. L'opzione 'ITL5=0' comunica al simulatore che il numero di iterazioni per la convergenza del calcolo di ogni punto del transitorio può essere anche infinito (in questo caso 0 significa infinito!). Il valore di default per questa opzione è 5000, e un qualsiasi altro valore, ad esempio 20000, può essere stabilito con 'ITL5=20000'.

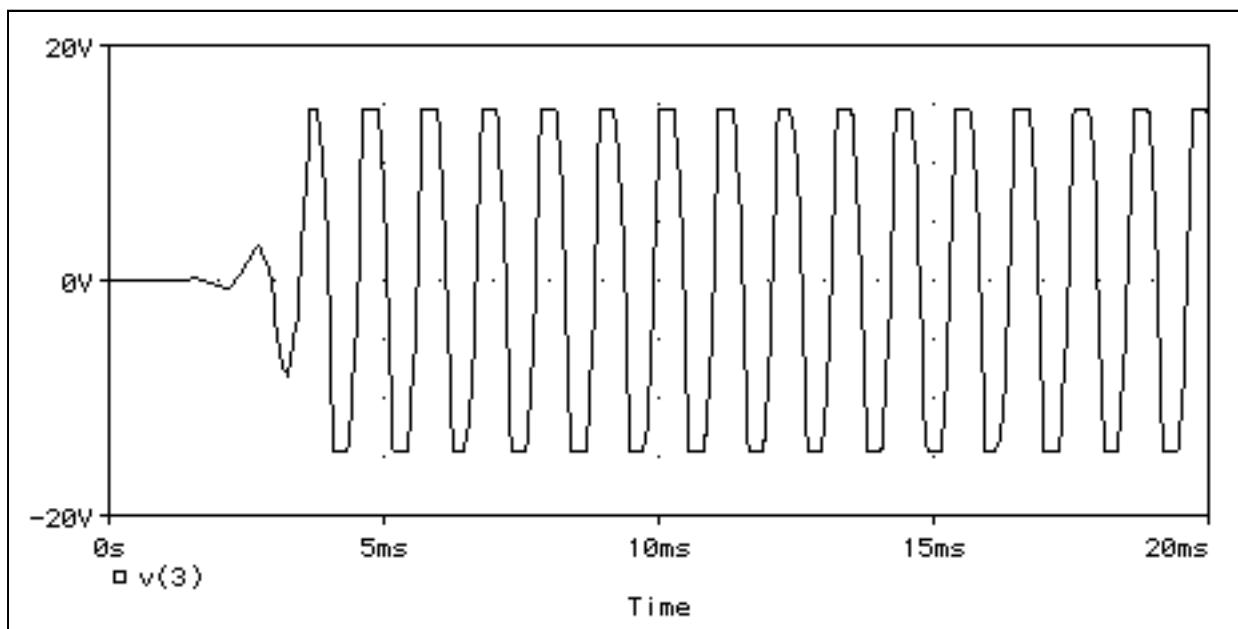


Figura 8.23: uscita dell'oscillatore a ponte di Wien.

8.8 Modello completo di amplificatore operazionale

Il circuito che ci accingiamo a presentare è un esempio di amplificatore operazionale completo, costituito cioè da transistori e resistenze. In Figura 8.24 possiamo notare lo schema di tale circuito.

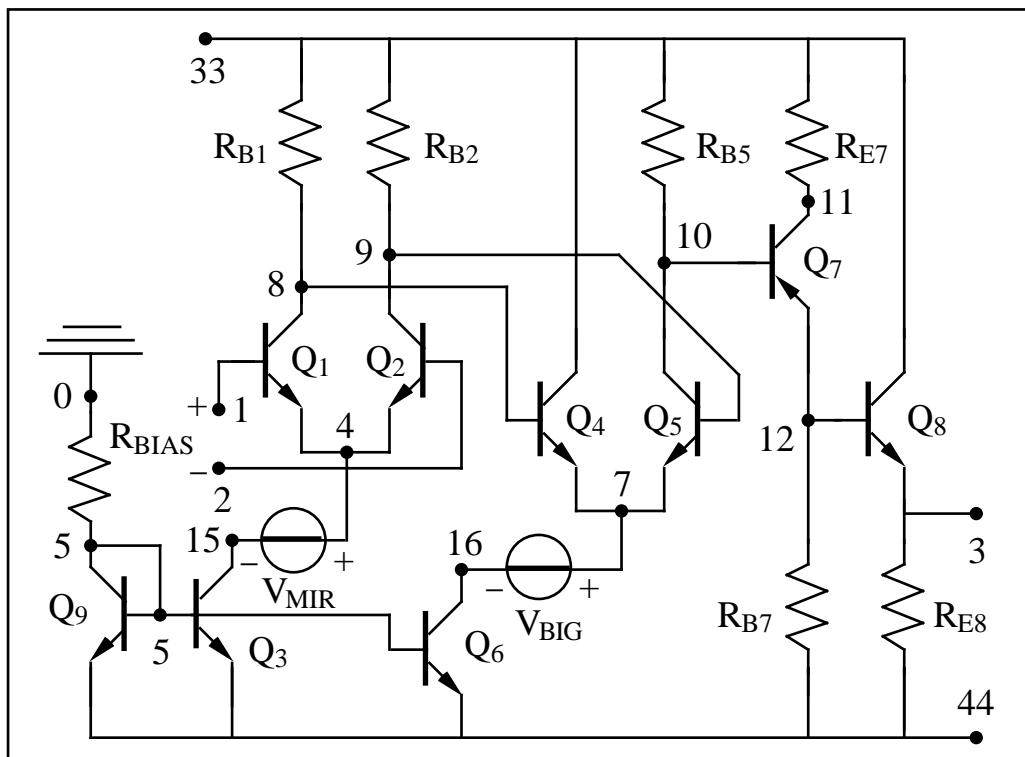


Figura 8.24: schema di amplificatore operazionale.

Il circuito presenta uno stadio differenziale all'ingresso, i cui transistori sono polarizzati da una coppia di transistori connessi a ‘specchio di corrente’. Dopo un’altra coppia di transistori in connessione differenziale, ritroviamo l’unico transistore ‘pnp’ e lo stadio di uscita, costituito da un transistore connesso a collettore comune. Di seguito è riportato il listato Spice che realizza questo circuito.

```
.SUBCKT OPAMP 1 2 3 33 44
*
* 1    Ingresso non invertente
* 2    Ingresso invertente
* 3    Uscita
* 33   Alimentazione positiva
* 44   Alimentazione negativa
*
.MODEL QN NPN(BF=100,RB=100,TF=80P,CJE=5.5p,VJE=0.75
+ MJE=0.5,BR=5.6,TR=3n,CJC=1.6P,VJC=0.75,MJC=0.36,VAF=100,IS=10f)
```

```

*
.MODEL QNBIG NPN(BF=100,RB=25,TF=80P,CJE=22P,VJE=0.75
+ MJE=0.5,BR=5.6,TR=3n,CJC=6.4P,VJC=0.75,MJC=0.36,VAF=100,IS=10f)
*
.MODEL QP NPN(BF=50,RB=33,TF=80P,CJE=5.5P,VJE=0.75
+ MJE=0.5,BR=5.6,TR=3n,CJC=1.6p,VJC=0.75,MJC=0.36,VAF=50,IS=10f)
*
RB1    33     8   20k
RB2    33     9   20k
RB5    33    10   3k
RB7    12    44  15.7k
RE7    33    11  2.3k
RE8     3    44   3k
RBIAS   0     5  28.6k
*
Q1     8     1   4   QN
Q2     9     2   4   QN
Q3    15     5  44   QN
Q4    33     8   7   QN
Q5    10     9   7   QN
Q6    16     5  44  QNBIG
Q7    12    10  11   QP
Q8    33    12   3   QN
Q9     5     5  44   QN
*
VMIR   4    15   0
VBIG   7    16   0
.ENDS    OPAMP

```

Provate voi, per esercizio, ad utilizzare il precedente sottocircuito per realizzare il circuito integratore e quello derivatore: notate le differenti approssimazioni rispetto ai circuiti ideali corrispondenti.

Capitolo 9

Altri programmi ed Opzioni

9.1 Introduzione

9.2 Il post-processore Probe

9.3 Il programma Schematics

9.4 Altri programmi

9.5 Opzioni

9.6 Bibliografia

Sommario

In questo capitolo passeremo in rassegna i principali programmi forniti a corredo di Spice ed approfondiremo il comando ‘.OPTIONS’.

9.1 Introduzione

In questo ultimo capitolo diamo una breve descrizione dei principali programmi forniti a corredo di Spice, il più famoso e utilizzato dei quali è sicuramente *Probe*. Questi programmi accessori sono utili per la visualizzazione grafica dei risultati ottenuti dalla simulazione, come nel caso di Probe, oppure servono a immettere i componenti e le connessioni del circuito in maniera grafica, utilizzando ‘da video’ la libreria di componenti fornita con Spice, come nel caso di *Schematics*.

Bisogna dire però che questi programmi non sono presenti in tutte le versioni di Spice e spesso non hanno le stesse funzionalità e gli stessi comandi. Perciò di essi daremo solo una breve descrizione, convinti che, in questi casi, l’uso costituisce la migliore strada per imparare a usarli.

Nel seguito del capitolo, parleremo anche del comando ‘.OPTIONS’, di cui abbiamo già dato qualche cenno nei precedenti capitoli. Esso serve semplicemente a immettere e specificare i valori numerici di alcune opzioni del programma e, per questo motivo, forniremo un elenco più o meno completo di tutte le opzioni disponibili ponendo l’accento su quelle più utilizzate.

Concluderà il capitolo un piccolo elenco bibliografico di pubblicazioni su Spice.

9.2 Il post-processore Probe

Il programma Probe, presente in quasi tutte le versioni di Spice, tra cui quella PSpice della Microsim Corporation, costituisce un aiuto determinante per visualizzare i risultati delle simulazioni. Possiamo dire che se Spice è la mente, Probe è il braccio. È lo strumento che ci permette di visualizzare agevolmente i risultati ottenuti dal simulatore.

Abbiamo già introdotto all’inizio di questo libro la possibilità che Spice ci offre anche di stampare tabelle di dati (o anche grafici) nel file .OUT, un file di tipo testo (ASCII), per cui visualizzabile semplicemente su ogni computer, dai PC ai Macintosh fino ai terminali dei grandi computer funzionanti con il sistema operativo UNIX. Ma la versatilità di Probe è indubbiamente da preferire laddove esista un terminale grafico, oppure quando si usino le versioni di Spice sui personal computer funzionanti in ambiente Windows o Macintosh.

Probe, nelle versioni più recenti, ha un’interfaccia grafica con menu a tendine e permette di aprire, chiudere, salvare con nome i file, oppure stampare i risultati sulla stampante di sistema.

Il collegamento funzionale con Spice è dato dai file con l’estensione .DAT creati dal simulatore stesso. Aprendo i file con questa estensione, avremo a disposizione

tutte le analisi contenute nel file stesso, una per volta, selezionata subito dopo l'apertura del file.

Entrando più nel dettaglio, è possibile visualizzare sia più tracce e più segnali sullo stesso grafico, sia visualizzare grafici diversi e indipendenti tra loro. Poi, a seconda dell'analisi effettuata, cambierà la grandezza sull'asse delle ascisse. Ad esempio, un'analisi in transitorio avrà il tempo come variabile sulle ascisse, laddove un'analisi in continua avrà la grandezza del circuito che varia nella riga '.DC' stessa. Se variano due grandezze, allora una sarà sull'asse delle ascisse e per ogni grandezza sull'asse delle ordinate ci saranno più curve, ognuna delle quali avente come parametro l'altra grandezza variabile dell'analisi '.DC'.

Infine, un'analisi '.AC' produrrà un'asse delle ascisse con la frequenza che varia secondo il tipo di analisi scelta: lineare, a decadi o a ottave.

Spesso sono presenti anche alcune funzionalità di post-elaborazione grafica. È possibile visualizzare funzioni elementari delle variabili del circuito, ad esempio il logaritmo naturale di una corrente oppure l'inverso della tangente del rapporto tra due tensioni. Esiste, poi, la possibilità di inserire etichette, frecce, linee in modo da rendere più chiara l'interpretazione di particolari punti del grafico. Vi è pure la possibilità di una ricerca automatica del massimo e del minimo assoluto del grafico visualizzato, oppure la ricerca dei punti in cui la derivata numerica del grafico è unitaria (funzione molto utile per lo studio dei circuiti digitali).

Un'altra possibilità notevole, che abbiamo usato anche noi nel corso del libro, è quella di stampare i grafici su file secondo i formati più conosciuti come Jpeg, Eps, Hgl oppure formati tipici del sistema operativo come Bmp per PC e Pict per Macintosh.

9.3 Il programma Schematics

Schematics è un programma introdotto nelle ultime versioni di Pspice per PC e costituisce una sorta di CAD (Computer Aided Design) per costruire circuiti.

Dopo l'avvio del programma, compare sullo schermo una finestra che, oltre ai soliti menù a tendine, presenta una griglia di punti: questo è il nostro 'tavolo di lavoro'. È possibile aprire una finestra in cui si visualizzano i simboli grafici dei componenti che vogliamo utilizzare (resistenze, induttanze, generatori, elementi non lineari). All'atto della scelta si immetteranno anche i parametri caratteristici del componente, ad esempio per un resistore apparirà il riquadro in cui scrivere il valore della resistenza. Poi, ritornati sulla griglia, con l'aiuto del puntatore del mouse, potremo disporre il componente sulla griglia e connetterlo, mediante fili 'virtuali' agli altri componenti già presenti. Una volta terminato il disegno, non è necessario numerarne i nodi. L'unico nodo che bisogna specificare è quello che

indica il riferimento di tensione, ciò viene fatto connettendo al nodo prescelto il simbolo grafico della ‘terra’.

Nella finestra Schematics è possibile specificare il tipo e i parametri delle analisi che si vogliono effettuare e poi vedere i risultati tramite Probe.

L’utilità maggiore di Schematics è che è possibile ‘vedere’ il circuito mentre si forma e inoltre una stampa del disegno ottenuto rende più chiara l’interpretazione da parte di chiunque. Bisogna porre particolare attenzione alle interconnessioni tra i componenti in quanto eventuali ‘falsi contatti’ (visivamente ci sembra di aver effettuato la connessione tra due componenti, mentre il simulatore non la ha accettata) provocano un diverso comportamento della rete difficilmente prevedibile.

Ovviamente il programma, per effettuare delle analisi, è costretto comunque a generare un file .CIR a partire dal disegno del circuito, perché il simulatore Spice non interpreta il disegno ma solo il file .CIR.

Lo svantaggio principale di tale tecnica di costruzione è che dipende fortemente dalla versione di Spice che usate, per cui spostarvi su un altro computer, o su un altro sistema operativo potrebbe farvi perdere la maggior parte del lavoro. È un po’ come se al posto di imparare la lingua inglese con cui potete dialogare in quasi tutto il mondo, voi imparaste un dialetto locale di un paesino dell’Inghilterra; perdereste la maggior parte dei vantaggi dell’Inglese!

Il consiglio a tal proposito è di partire sempre con l’imparare lo strumento generale quale è il linguaggio Spice tipo testo che è più facilmente controllabile dall’utente e poi, una volta impadronitisi del linguaggio, provare anche con il CAD grafico Schematics.

Notiamo infine che nella ultima versione per Macintosh non è presente il programma Schematics.

9.4 Altri programmi

Le varie installazioni di Spice hanno anche alcuni programmi accessori come un Editor di testi specifico per la generazione dei file .CIR oppure programmi, come Parts, per la creazione di nuovi modelli di dispositivi da impiegare in Spice. Molte delle questioni trattate nel presente capitolo, infatti, sono relative all’uso di PSpice; l’analisi approfondita degli altri programmi che nel frattempo sono stati proposti sul mercato non soltanto esula dagli scopi propostici nello scrivere questo manuale, ma ci sembra anche piuttosto sterile. Siamo sicuri che, una volta acquisiti i fatti fondamentali studiando una particolare realizzazione del programma Spice,

vi risulterà molto semplice comprendere tutti gli accrescimenti e le novità introdotte nelle nuove versioni.

9.5 Opzioni

In questo paragrafo forniremo un elenco dettagliato delle opzioni di simulazione a disposizione di cui possiamo cambiare il valore nel file ‘.CIR’. L’uso di queste opzioni permette di variare alcuni parametri interni utilizzati nei vari algoritmi numerici, o più semplicemente permette di variare i risultati presentati nel file ‘.OUT’ o, infine, consente di stabilire parametri globali quali la temperatura di funzionamento e le dimensioni dei MOSFET, ad esempio.

L’istruzione di cui stiamo parlando, peraltro già introdotta nei capitoli precedenti in qualche esempio, è del tipo

```
.OPTIONS    OPZ1,      OPZ2=VAL1,      OPZ3=VAL2,  . . .
```

Innanzitutto bisogna dire che vi sono opzioni di due tipi: un tipo per cui bisogna specificare un valore associato all’opzione e un altro tipo che invece non ha associato alcun valore. Queste ultime sono opzioni di tipo ‘binario’, la loro presenza o assenza esaurisce già il loro significato. Nella riga generica precedente OPZ1 è un’opzione di tipo binario, mentre OPZ2 e OPZ3 hanno bisogno di un valore. L’ordine con cui si specificano le opzioni non è assolutamente importante e possono essere utilizzate più righe .OPTIONS: il loro effetto è analogo a mettere tutte le opzioni in una sola riga che va a capo. Se una stessa opzione è specificata più volte, il simulatore considera solo l’ultimo valore come valido.

Cominciamo con l’elenco e la funzione delle opzioni di tipo ‘binario’.

ACCT - La presenza di questa opzione indica che vogliamo maggiori informazioni nel file .OUT relative alle analisi, .AC o altro, che abbiamo richiesto al simulatore.

EXPAND - La presenza di questa opzione permette la creazione dell’elenco delle caratteristiche di tutti i dispositivi creati all’interno degli eventuali sottocircuiti presenti.

LIBRARY - Fornisce l’elenco delle linee prelevate dalle librerie di componenti incluse nel file.

LIST - Fornisce un elenco più approfondito dei dispositivi utilizzati nel circuito simulato.

NODE - Fornisce la tabella dettagliata dei nodi impiegati nel circuito e dei dispositivi ad essi connessi.

NOECHO - La presenza di questa opzione evita che nel file ‘.OUT’ sia ripetuto l’insieme di righe che costituivano il file .CIR.

NOMOD - Evita che nel file .OUT siano listati i modelli e i valori dei parametri dei componenti utilizzati e il loro aggiornamento quando sia presente l’istruzione ‘.TEMP’ che fa ripetere l’analisi a varie temperature di lavoro.

NOPAGE - Sopprime la creazione di pagine diverse nel file ‘.OUT’ per ogni analisi e le relative intestazioni formate da asterischi, date, e così via. In altri termini, rende più breve il file ‘.OUT’.

OPTS - È l’opzione nell’opzione! Indica al simulatore che desideriamo conoscere i valori di tutte le opzioni, specificate e non, nel file .OUT.

Ovviamente, l’assenza di queste opzioni provoca l’effetto opposto a quello specificato.

Passiamo ora all’elenco delle opzioni che hanno bisogno di un valore ad esse associato. In questo caso forniremo anche il valore di default e l’eventuale unità di misura.

DISTRIBUTION - Indica il tipo di distribuzione statistica da usare per la simulazione ‘Monte Carlo’. Il valore di default è UNIFORM, esistono, però, anche altre possibilità tra le quali GAUSS (distribuzione di tipo gaussiano).

ABSTOL - L’accuratezza da impiegare per la valutazione delle correnti nella soluzione di sistemi non lineari. L’unità di misura è l’ampere e il valore di default è 1 pA. Il più piccolo valore di corrente osservabile è quello specificato da questa opzione.

CHGTOL - L’accuratezza da impiegare per la valutazione delle cariche. L’unità di misura è il coulomb e il valore di default è 0.1 pC.

CPTIME - Il tempo massimo a disposizione per il processore per effettuare tutte le analisi richieste. L'unità di misura è il secondo e il valore di default è un milione di secondi, ovvero poco più di 11 giorni pieni di elaborazione! Ovviamente, se viene superato il valore limite, il programma uscirà senza completare quanto richiesto. Questa opzione serve soprattutto in ambiente UNIX dove i tempi del processore centrale, su cui 'girano' i programmi, sono prefissati; pertanto, questa opzione evita che sia il sistema operativo ad interrompere la simulazione.

DEFAD - Questa opzione specifica il valore di default da impiegare come area per il drain nella definizione dei MOSFET (AD). L'unità di misura è il metro quadro e il valore di default è zero. Sappiamo che per questi dispositivi è necessario specificare le dimensioni fisiche perché molti parametri dipendono da queste ultime. Per cui, con questa opzione, è possibile fissare 'una tantum' questi valori senza specificarli nelle righe del programma. Si noti che, essendo il valore di default nullo, è una opzione che va assolutamente specificata, nella riga .OPTIONS oppure nella riga del dispositivo stesso.

DEFAS - Questa opzione specifica il valore di default da impiegare come area per il source nella definizione dei MOSFET (AS). L'unità di misura è il metro quadro e il valore di default è zero. Anche per questa opzione valgono tutte le considerazioni fatte nel punto precedente.

DEFL - Questa opzione specifica il valore di default da impiegare come lunghezza nella definizione dei MOSFET (L). L'unità di misura è il metro e il valore di default è 100 μm .

DEFW - Questa opzione specifica il valore di default da impiegare come larghezza nella definizione dei MOSFET (W). L'unità di misura è il metro e il valore di default è 100 μm .

GMIN - Definisce la minima conduttanza connessa in parallelo ad una giunzione p-n. L'unità di misura è il siemens e il valore di default è 1 pS. Notate che questa opzione interviene e influenza la ricerca della soluzione di qualsiasi sistema lineare che impiega il simulatore al suo interno.

ITL1 - L'acronimo significa 'ITeration Limit', ovvero specifica il massimo numero di iterazioni da impiegare per la soluzione in continua e per la soluzione del punto iniziale di un'analisi in transitorio. Il valore di default è 100.

ITL2 - Stabilisce il limite massimo di iterazioni per il calcolo di una curva di trasferimento in continua. Il valore di default è 50.

ITL3 - Riportiamo per completezza anche quest'opzione che, però, ha valore solo in SPICE2 (una delle prime versioni del programma distribuita direttamente dall'Università della California) ed è associata a LVLTIM=1. In questo caso definisce il limite inferiore per le iterazioni relative ad ogni istante di tempo.

ITL4 - Pone il limite massimo al numero di iterazioni da effettuare ad ogni istante di tempo, nell'analisi in transitorio, prima di passare al calcolo con una costante di tempo otto volte più piccola. Il valore di default è 10.

ITL5 - Rappresenta il numero massimo totale di iterazioni da effettuare in un'analisi in transitorio e serve ad evitare analisi troppo lunghe. Il valore di default è 5000. Utilizzando il valore ITL5 = 0, disabilitiamo tale opzione, per cui il simulatore non avrà limite al numero di iterazioni e queste diventeranno infinite!

LIMPTS - Specifica il massimo numero di punti da stampare nel file '.OUT' per un'analisi '.PRINT' oppure '.PLOT'. Il valore di default è infinito. Notiamo che, ovviamente, tale opzione influenza solo il numero di punti da visualizzare e non il numero di punti da impiegare per l'analisi stessa.

NUMDGT - Indica il numero di cifre significative da stampare nelle tabelle del file '.OUT'. Il suo valore deve essere compreso tra 1 e 8, il valore di default è 4.

PIVREL - Rappresenta il rapporto tra il più piccolo pivot accettabile in una colonna e l'elemento più grande di quella colonna. Il valore di default è 10^{-3} . Questa opzione interviene nella soluzione di sistemi lineari e in particolare nella tecnica di eliminazione gaussiana con 'pivoting'.

PIVTOL - È il minimo valore accettabile di un elemento di una matrice perché questo possa essere utilizzato come pivot nella soluzione di un sistema lineare. Se nessun elemento della matrice è più grande di questo valore, allora la matrice viene dichiarata singolare. Il valore di default è 10^{-13} .

REL TOL - Definisce il massimo errore relativo ammesso per le tensioni e per le correnti perché si possa considerare raggiunta la convergenza nella soluzione di sistemi non lineari. Il valore di default è 10^{-3} .

TNOM - Stabilisce la temperatura di riferimento alla quale si suppone siano misurati tutti i parametri forniti nella definizione dei modelli dei dispositivi impiegati. L'unità di misura è il grado centigrado e il valore di default è 27°C .

VNTOL - L'accuratezza da impiegare per la valutazione delle tensioni nella soluzione di sistemi non lineari. L'unità di misura è il volt e il valore di default è $1 \mu\text{V}$. Il più piccolo valore di tensione osservabile è quello specificato da questa opzione.

9.6 Bibliografia

In questo ultimo paragrafo riportiamo l'elenco di alcuni utili testi e articoli che possono servire per approfondire gli argomenti riportati in questo testo. Notate che alcuni di essi sono scritti in lingua inglese, una lingua che voi conoscerete sicuramente perché utile al vostro lavoro attuale e futuro!

Cominciamo proprio dalla ‘concorrenza’. Ecco un elenco di libri in cui trovate più o meno (speriamo meno) la sintassi e le possibilità offerte dal nostro programma Spice:

- Andrei Vladimirescu, *Spice*, McGraw-Hill 1995 (in italiano);
- Andrei Vladimirescu, *The Spice Book*, John Wiley & Sons 1994 (in inglese);
- James W. Nilsson, *Introduction to Spice*, Addison-Wesley Publishing Company 1990, (in inglese);
- Mario Biey, Spice e PSpice: *Introduzione a Spice*, CLUT, Torino 1993 (in italiano).

Prima di concludere, è appena il caso di sottolineare che, come riferimento, potete consultare le guide fornite con il programma stesso in tutte le sue versioni.