

**Università degli Studi di Padova**

FACOLTÀ DI INGEGNERIA  
Corso di Laurea in Ingegneria dell'Informazione

TESI DI LAUREA TRIENNALE

# Metodi di Monte Carlo applicati a sistemi di comunicazione digitali

**LAUREANDA:**  
Giulia Agostini

**RELATORE:**  
Dr. Lorenzo Finesso

Anno Accademico 2011-2012

*Ai miei genitori,  
perchè possano sempre  
essere orgogliosi di me.*

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
<b>2</b>	<b>Metodi di Montecarlo</b>	<b>5</b>
2.1	Origini storiche . . . . .	5
2.2	Idea di base . . . . .	6
2.2.1	Relative Frequency . . . . .	7
2.2.2	Stimatori Unbiased e Consistenti . . . . .	7
2.2.3	Generatori di numeri casuali . . . . .	8
2.3	Integrazione Monte Carlo . . . . .	9
2.3.1	Concetti base . . . . .	9
2.3.2	Convergenza . . . . .	10
2.3.3	Intervalli di confidenza . . . . .	11
2.4	Metodi asintotici e metodi Markov chain Monte Carlo . . . . .	12
<b>3</b>	<b>Applicazioni ai sistemi di comunicazione</b>	<b>15</b>
3.1	Canale AWGN . . . . .	15
3.2	Binary Phase Shift Keying, BPSK . . . . .	17
3.2.1	BPSK . . . . .	17
3.2.2	Simulazione 1 . . . . .	19
3.2.3	Analisi dei risultati . . . . .	21
3.3	Binary Frequency Shift Keying, BFSK . . . . .	23
3.3.1	BFSK . . . . .	23
3.3.2	Simulazione 2 . . . . .	23
3.3.3	Analisi dei risultati . . . . .	24
3.4	Integrazione Monte Carlo in un canale AWGN . . . . .	25
3.4.1	Simulazione di Monte Carlo per il calcolo della coda di una $N(0,1)$ . . . . .	27
<b>A</b>	<b>Codice MATLAB</b>	<b>29</b>
A.1	BPSK.m . . . . .	29
A.2	BFSK.m . . . . .	30
A.3	CodaGauss.m . . . . .	31
<b>B</b>	<b>Richiami di teoria della probabilità</b>	<b>32</b>
B.1	Legge dei grandi numeri . . . . .	32
B.2	Variabili aleatorie Uniforme e Gaussiana . . . . .	32
B.3	Variabile aleatoria Binomiale . . . . .	33
	<b>Bibliografia</b>	<b>34</b>

# Capitolo 1

## Introduzione

I fenomeni aleatori permeano la vita di tutti i giorni e, in particolare, caratterizzano diversi ambiti scientifici, matematica e fisica in primis. L'interpretazione con metodi statistici di questi fenomeni conobbe una rinascita a partire dalla metà del secolo scorso con la formulazione dei metodi di Monte Carlo; traguardo raggiunto grazie all'intersezione tra le ricerche sul comportamento dei neutroni in una reazione a catena e i risultati raggiunti in ambito elettronico con la realizzazione del primo computer, ENIAC.

I metodi di Monte Carlo svolgono un ruolo determinante nella simulazione di fenomeni aleatori, nonché nell'analisi e nell'interpretazione dei dati sperimentali. Essendo i numeri casuali alla base del concetto di simulazione i metodi di Monte Carlo si servono di questi ultimi come strumento per valutare qualcosa che non è casuale, proponendosi in alternativa ai metodi deterministici: tipicamente, l'aleatorietà richiesta dal modello che si vuole analizzare viene simulata da un generatore di numeri casuali il cui output è rappresentato da una sequenza di variabili aleatorie indipendenti ed identicamente distribuite  $U(0, 1)$ , ossia variabili aleatorie continue uniformemente distribuite nell'intervallo  $(0, 1)$ . In realtà, i generatori di numeri casuali sono dei semplici programmi deterministici che producono sequenze non del tutto casuali, ma che si comportano come dei numeri casuali indipendenti, ossia come una sequenza iid di variabili aleatorie  $U(0, 1)$ ; per questo si parla di sequenze di numeri pseudo-casuali.

Alla base dei metodi di Monte Carlo vi è, dunque, l'esecuzione di esperimenti stocastici, o casuali. Un evento di interesse viene identificato e un esperimento di base viene ripetuto un gran numero di volte. Il rapporto tra il numero di occorrenze dell'evento di interesse e il numero totale di repliche dell'esperimento di base fornisce la *relative frequency* dell'evento di interesse; quest'ultima, che è una variabile aleatoria, rappresenta un buon stimatore della probabilità che l'evento di interesse si verifichi. In particolare, per stimatori unbiased e consistenti, la relative frequency converge alla probabilità dell'evento di interesse al crescere del numero di ripetizioni dell'esperimento.

I metodi di Monte Carlo consentono di simulare il comportamento di un evento di interesse e, in genere, restituiscono come risultato una variabile aleatoria le cui proprietà, quali media, varianza, funzione densità di probabilità ecc., ci forniscono importanti informazioni sulla qualità della simulazione.

Dato il mio particolare interesse per le telecomunicazioni, è mia intenzione, con questo breve elaborato, soffermarmi su una particolare applicazione dei metodi

di Monte Carlo: la simulazione del comportamento di un canale di comunicazione digitale AWGN, in modulazioni *BPSK* e *BFSK*, quando vi si trasmette un segnale binario, valutando gli effetti del rumore e, in particolare, la probabilità di errore.

Infine, intendo evidenziare in modo conciso alcune applicazioni dei metodi di Monte Carlo nel calcolo integrale e fare una breve digressione sui *metodi asintotici* e sugli MCMC, *metodi Markov chain Monte Carlo*.

## Capitolo 2

# Metodi di Montecarlo

### 2.1 Origini storiche

Nel 1945 due eventi segnarono la storia dell'umanità: la prima bomba atomica e il primo computer elettronico. Indiscutibili furono i cambiamenti apportati nell'ambito della ricerca accademica e delle scienze applicate: questi eventi portarono ad una sorta di rinascimento dei metodi matematici con l'avvento del campionamento statistico, ribattezzato da Nicholas Metropolis *Metodo di Monte Carlo*.

Nel corso del Secondo Conflitto Mondiale, un gruppo di scienziati, ingegneri e tecnici lavorarono freneticamente alla realizzazione di ENIAC, il primo computer elettronico, alla University of Pennsylvania di Philadelphia; l'idea di base era che se un circuito elettronico poteva contare, allora poteva svolgere operazioni aritmetiche e risolvere equazioni differenziali a velocità quasi incredibili. In quel periodo John von Neumann, rimasto particolarmente colpito dai problemi termonucleari sollevati nel corso del Progetto Manhattan, chiese a Nicholas Metropolis e Stanley Frankel se fossero interessati alla realizzazione di un preliminare modello computazionale di una reazione termonucleare per ENIAC; la reazione fu più che positiva. Alla fine della guerra Anthony Turkevich si unì al gruppo che, nel frattempo, aveva ricevuto da Edward Teller l'ulteriore incarico di documentare e fornire un resoconto critico dei risultati raggiunti simulando il modello elaborato su ENIAC.

Nella primavera del 1946 i risultati raggiunti portarono a guardare con ottimismo alla fattibilità di un'arma termonucleare; al gruppo si unì Stan Ulam che, grazie alla sua notevole preparazione matematica, riuscì ad intravedere una possibilità di rinascita per i metodi statistici caduti in disuso a causa della lunghezza e della tediosità del calcolo. Bastò questa sua intuizione ad innescare la scintilla che avrebbe portato allo sviluppo del metodo di Monte Carlo, il cui nome fu suggerito da Metropolis ispirato dalla famosa cittadina del Principato di Monaco meta prediletta degli amanti del gioco d'azzardo.

Primo fra tutti, von Neumann sottolineò la possibilità di utilizzare questo nuovo approccio statistico per risolvere il problema della diffusione dei neutroni nel materiale fissionabile: l'idea di fondo consisteva nel seguire il comportamento di un gran numero di catene di singoli neutroni e sviluppare una descrizione statistica della storia del singolo neutrone, tenendo conto in modo appropriato

dei vari fattori fisici e geometrici coinvolti. Fondamentale per l'applicazione del metodo era la presenza di una sorgente di numeri pseudo-casuali uniformemente distribuiti; una volta determinato un algoritmo in grado di soddisfare questo requisito, questi numeri dovevano essere trasformati in una distribuzione non uniforme caratterizzante il problema di interesse.

Ben presto il metodo di Monte Carlo rivelò la sua utilità non solo nello studio di sistemi fisici, ma anche nel calcolo integrale; sin dal principio vennero organizzate diverse conferenze internazionali per discutere le sue possibili applicazioni. Nel 1947, Enrico Fermi costruì il FERMIAC, uno strumento in grado di sviluppare la genealogia dei neutroni in due dimensioni, che ricorreva alla nuova tecnica; già negli anni '30, quando lavorava a Roma presso l'istituto di via Panisperna, il fisico italiano si era servito di stime ottenute con tecniche di campionamento statistico per lo studio del moto dei neutroni. Infine, dopo essere stato testato, con esito favorevole, con ENIAC, il metodo di Monte Carlo si affermò in modo definitivo ed è tuttora uno strumento irrinunciabile in diversi ambiti scientifici dalla fisica, alla matematica, alla finanza ecc; fondamentale per la stima e l'interpretazione di fenomeni aleatori.

## 2.2 Idea di base

La definizione formale dei metodi di Monte Carlo fu data, tra gli altri, da Halton, nel 1970, che lo descrisse come una 'tecnica' che *'rappresenta la soluzione di un problema come un parametro di un'ipotetica popolazione e ricorre ad una sequenza di numeri casuali per costruire un campione della popolazione, a partire dal quale si possono ottenere stime statistiche del parametro di interesse'*. L'idea fondamentale che sta alla base del metodo di Monte Carlo consiste nell'esprimere la soluzione del problema che voglio simulare sperimentalmente come valore atteso di una qualche variabile aleatoria  $X \sim f_X(x)$ :  $A = E[X] = \int_{-\infty}^{+\infty} x f_X(x) dx$ . Quindi, se possiamo generare  $N$  variabili aleatorie  $X_1, \dots, X_N$  iid, ossia indipendenti ed identicamente distribuite, con la stessa funzione di distribuzione di  $X$ , possiamo ricorrere all'approssimazione

$$A \approx \hat{A}_N = \frac{1}{N} \sum_{k=1}^N X_k$$

Per la legge dei grandi numeri  $\hat{A}_N \rightarrow A$ , per  $N \rightarrow \infty$ .

Le  $X_k$  e  $\hat{A}_N$  sono casuali e possono cambiare ogni volta che faccio girare il programma; mentre il numero che ci interessa,  $A$ , non è casuale.

Infine, è importante chiarire che una simulazione Monte Carlo descrive una simulazione nel corso della quale il parametro di un sistema viene stimato ricorrendo ai metodi di Monte Carlo, mentre la stima di Monte Carlo è il processo di stima del valore di un parametro ottenuto eseguendo ripetutamente un esperimento stocastico, o casuale.

### 2.2.1 Relative Frequency

La stima di Monte Carlo si basa sul concetto di *relative frequency*; il primo passo per definirla consiste nello specificare un esperimento casuale, ossia un esperimento il cui esito non può essere predetto in modo esatto, ma può essere definito statisticamente, ed un evento di interesse. Un evento è il risultato, o l'insieme dei risultati, associato all'esecuzione di un esperimento casuale.

Il prossimo passo nel metodo di Monte Carlo consiste nel ripetere l'esperimento un numero significativo di volte,  $N$ . Contiamo quindi il numero di occorrenze,  $N_A$ , corrispondenti all'evento di interesse,  $A$ ; la probabilità che l'evento  $A$  si verifichi è approssimata dalla relative frequency, che è definita dal rapporto tra il numero di occorrenze dell'evento e il numero di esecuzioni dell'esperimento,  $\frac{N_A}{N}$ . La probabilità dell'evento  $A$  è ottenuta ripetendo l'esperimento casuale un numero infinito di volte:

$$Pr(A) = \lim_{N \rightarrow \infty} \frac{N_A}{N}$$

Per  $N < \infty$  la quantità  $\frac{N_A}{N}$  è uno stimatore di  $Pr(A)$ , denotato con  $\widehat{Pr}(A)$ . È importante evidenziare che, per valori finiti di  $N$ ,  $N_A$  è una variabile aleatoria e di conseguenza  $Pr(A)$  è una variabile aleatoria; la statistica di questa variabile aleatoria determina la precisione dello stimatore e quindi la qualità della simulazione.

### 2.2.2 Stimatori Unbiased e Consistenti

Gli stimatori di Monte Carlo devono soddisfare importanti proprietà. Innanzitutto, vogliamo che lo stimatore sia *unbiased*; ossia, se  $\widehat{A}$  rappresenta la stima del parametro di interesse,  $A$ , vogliamo che

$$E[\widehat{A}_N] = A$$

In altre parole, desideriamo che in media il risultato corretto sia raggiunto. Inoltre vogliamo che i valori stimati abbiano una piccola varianza, proprietà che quasi sempre tende a manifestarsi in modo significativo al crescere del numero di repliche dell'esperimento casuale. Stimatori di questo tipo si dicono *consistenti*:

$$\sigma_{\widehat{A}_N}^2 \rightarrow 0 \quad \text{per} \quad N \rightarrow \infty$$

Per stimatori unbiased e consistenti, l'errore

$$e = \widehat{A}_N - A$$

ha media nulla e varianza,  $\sigma_e$ , convergente a zero per  $N \rightarrow \infty$ .

### 2.2.3 Generatori di numeri casuali

Le variabili aleatorie usate nei metodi di Monte Carlo sono generate da un generatore di numeri pseudo-casuali. La procedura `double rng()` è un perfetto generatore di numeri casuali

```
for( k=0; k<n; k++ ) U[k] = rng();
```

infatti, produce un array di variabili aleatorie uniformi standard iid.

Il computer è una macchina deterministica; un generatore di numeri pseudo-casuali simula l'aleatorietà senza avere realmente una natura aleatoria.

**Definizione 1.** *Un generatore di numeri pseudo-casuali è una quintupla  $X = (S, s_0, T, U, G)$ , dove  $S$  è un insieme finito di stati,  $s_0 \in S$  è lo stato iniziale, o seme, la mappa  $T : S \rightarrow S$  è la funzione di transizione,  $U$  è l'insieme finito dei simboli d'uscita e  $G : S \rightarrow U$  è la funzione d'uscita.*

Lo stato del generatore è inizialmente  $s_0$  ed evolve in accordo con la relazione  $s_n = T(s_{n-1})$ , per  $n = 1, 2, 3, \dots$ . Al passo  $n$ , il generatore produce in uscita il numero  $u_n = G(s_n)$ ; i vari  $u_n$  sono detti osservazioni e sono i numeri casuali prodotti dal generatore. Chiaramente, la sequenza degli stati  $s_n$  è periodica, poichè lo spazio di stato  $S$  è finito; perciò, ad un certo punto il generatore deve rivisitare uno stato precedentemente visto, cioè  $s_j = s_i$  per qualche  $j > i \geq 0$ . Quindi, necessariamente  $s_{j+n} = s_{i+n}$  e  $u_{j+n} = u_{i+n} \quad \forall n \geq 0$ . La lunghezza del periodo è il più piccolo intero  $\rho > 0$  tale che per qualche intero  $\tau \geq 0$  e  $\forall n \geq \tau$ ,  $s_{\rho+n} = s_n$ . Il più piccolo  $\tau$  con questa proprietà è detto transitorio. Spesso  $\tau = 0$  e la sequenza è detta puramente periodica. È importante sottolineare che lunghezza del periodo non può superare  $|S|$ , ossia la cardinalità dello spazio di stato; i buoni generatori tipicamente hanno  $\rho$  molto vicina a  $|S|$ , per evitare un inutile spreco di memoria.

Nel caso esaminato, il seme è la raccolta di  $n$  variabili intere da 32 bit ciascuna; la funzione che aggiorna il valore del seme (funzione di transizione) è  $\Phi(s)$  e la funzione d'uscita è  $\Psi(s)$ . La funzione di transizione produce un nuovo seme:  $s' = \Phi(s)$ , e la funzione d'uscita produce un numero floating point (in virgola mobile), ossia un numero reale, casuale:  $u = \Psi(s) \in (0, 1)$ . La chiamata `u=rng()`; ha l'effetto

```
s ← Φ(s); return u = Ψ(s)
```

I più semplici e più largamente utilizzati generatori di numeri casuali sono i *linear congruential generators*, LCG: il seme è un numero intero appartenente all'intervallo  $[0, m)$  e, al passo  $n$ , lo stato è un intero  $s_n$  e la funzione di transizione,  $T$ , è definita dalla relazione

$$s_n = T(s_{n-1}) = (as_{n-1} + c) \bmod m$$

dove  $m > 0, a > 0$  e  $c$  sono interi chiamati rispettivamente modulo, moltiplicatore e costante additiva. In questo caso, l'operazione **mod m** consiste nel considerare il più piccolo resto non negativo modulo  $m$ .

Lo spazio  $S$  è l'insieme  $0, \dots, m-1$ ; per produrre valori nell'intervallo  $[0, 1]$ , basta semplicemente definire la funzione d'uscita  $G$  come  $u_n = G(s_n) = s_n/m$ .

Quando  $c = 0$ , si parla di *multiplicative linear congruential generator*, ML-CG. La massima lunghezza del periodo di un LCG è  $m$  generalmente; per un MLCG non può superare  $m - 1$ , poichè  $s_n = 0$  è uno stato da evitare. Due popolari valori di  $m$  sono  $m = 2^{31} - 1$  e  $m = 2^{32}$ .

## 2.3 Integrazione Monte Carlo

### 2.3.1 Concetti base

Assumiamo di voler calcolare l'integrale

$$I = \int_0^1 g(x) dx$$

dove  $g(x)$  è una funzione non nulla nell'intervallo di integrazione. Dalla teoria della probabilità sappiamo che il valore atteso di  $g(x)$  è dato da

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x) dx$$

dove  $f_X(x)$  è la funzione densità di probabilità della variabile aleatoria  $X$ . Se la funzione densità di probabilità di  $X$  soddisfa  $f_X(x) = 1$  nell'intervallo  $(0, 1)$  ed è zero altrove, segue che  $E[g(X)] = I$ . Perciò, se  $U$  è una variabile aleatoria uniformemente distribuita nell'intervallo  $(0, 1)$ , segue che

$$I = E[g(U)]$$

Sfruttando le considerazioni fatte a proposito della relative frequency possiamo scrivere

$$\lim_{N \rightarrow \infty} \left[ \frac{1}{N} \sum_{i=0}^N g(U_i) \right] = E[g(U)] = I$$

Perciò, simuliamo l'integrale per campionarlo in  $N$  punti dell'intervallo  $(0, 1)$ ; il valore atteso dei campioni poi fornisce uno stimatore per il valore dell'integrale.

La simulazione Monte Carlo di un sistema fa la stessa cosa; poichè solitamente non abbiamo un'espressione in forma chiusa per la statistica sufficiente nella regione di errore, campioni della statistica vengono generati usando una simulazione del sistema.

Possiamo quindi considerare la seguente approssimazione per lo stimatore di Monte Carlo dell'integrale

$$\frac{1}{N} \sum_{i=0}^N g(U_i) = \hat{I}_N$$

Riassumendo, lo stimatore per un integrale viene ottenuto valutando la funzione  $g(x)$  in  $N$  punti casuali distribuiti uniformemente nell'intervallo di interesse

e facendone la media.

Questo processo può essere applicato ad un qualsiasi integrale proprio:

$$I = \int_a^b g(x)dx, \quad \text{con } a, b \in \mathfrak{R}$$

Detta  $X$  una variabile aleatoria uniforme nell'intervallo  $(a,b)$ ,  $X \sim U(a, b)$ , la cui funzione densità di probabilità è

$$f_X(x) = \begin{cases} \frac{1}{b-a}, & \text{se } x \in (a, b) \\ 0, & \text{altrimenti} \end{cases}$$

Dopo una serie di passaggi otteniamo

$$\begin{aligned} I &= \int_a^b g(x)dx \\ &= (b-a) \int_a^b g(x) \frac{1}{b-a} dx \\ &= (b-a) \int_a^b g(x) f_X(x) dx \\ &= (b-a) E(X) \end{aligned}$$

Se ora consideriamo  $N$  variabili aleatorie iid  $X_1, \dots, X_N$ , uniformemente distribuite nell'intervallo  $(a,b)$ ,  $\hat{I}_N = (b-a) \frac{1}{N} \sum_{i=1}^N X_i$  può essere considerato un buon stimatore di  $I$ . Quindi, per  $N \rightarrow \infty$ , possiamo scrivere

$$I = (b-a)E(X)$$

Qualsiasi integrale proprio può dunque essere calcolato usando i metodi di Monte Carlo.

### 2.3.2 Convergenza

Supponiamo di dover calcolare il valore di un integrale,  $I$ , e che  $N$  campioni casuali, denotati con  $X_i$ , siano disponibili. Consideriamo come stimatore di  $I$

$$\hat{I}_N = (b-a) \frac{1}{N} \sum_{i=1}^N X_i$$

Assumiamo che gli  $N$  campioni siano iid; la media aritmetica dei campioni è data da

$$E \left\{ \frac{1}{N} \sum_{i=1}^N X_i \right\} = \frac{1}{N} \sum_{i=1}^N E(X_i) = \frac{NI}{N} = I$$

ossia lo stimatore  $\hat{I}_N$  è unbiased. Avendo assunto campioni indipendenti, la loro varianza è

$$\sigma_{\hat{I}_N}^2 = \frac{1}{N^2} \sum_{i=1}^N \sigma_x^2 = \frac{N\sigma_x^2}{N^2} = \frac{\sigma_x^2}{N}$$

ossia lo stimatore è consistente. Assumendo che  $X_i = g(U_i)$ , la varianza dei campioni,  $\sigma_x^2$ , è data da

$$\sigma_x^2 = \int_0^1 g^2(u)du - \left[ \int_0^1 g(u)du \right]^2 \quad (2.1)$$

Essendo lo stimatore unbiased e consistente, si può ottenere un'accurata stima dell'integrale se  $N$  è sufficientemente grande e se la varianza dei campioni è piccola. Perciò, le stime di Monte Carlo di un integrale possono essere molto precise, per un dato valore di  $N$ , se  $g(u)$  è approssimativamente costante nell'intervallo di integrazione, ossia se il suo valore oscilla poco rispetto al valor medio. Di conseguenza, se  $g(u)$  è costante nell'intervallo di integrazione, la stima fornita da  $\hat{I}_N$  è precisa per  $N = 1$ .

### 2.3.3 Intervalli di confidenza

La qualità di uno stimatore  $\hat{I}_N$  è spesso espressa in termini di intervallo di confidenza, che ci fornisce la probabilità  $(1 - \alpha)$  che le stime cadano in un dato intervallo di valori,  $(\pm \beta \sigma_{\hat{I}_N})$ . Vi sono, perciò, due parametri di interesse: la probabilità, determinata dal valore di  $\alpha$ , e l'intervallo, determinato dal valore di  $\beta$ .

L'intervallo di confidenza è definito dall'espressione

$$Pr \left\{ I - \beta \sigma_{\hat{I}_N} \leq \hat{I}_N \leq I + \beta \sigma_{\hat{I}_N} \right\} = 1 - \alpha \quad (2.2)$$

ossia ci riferiamo all'intervallo  $I \pm \beta \sigma_{\hat{I}_N}$  come l'intervallo di confidenza  $1 - \alpha$ .

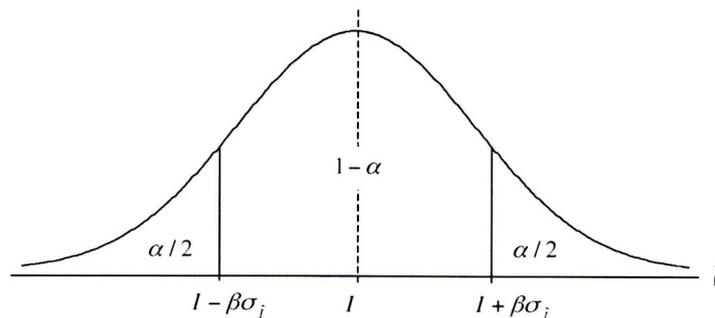


Figura 2.1: Intervallo di confidenza

Consideriamo ora il valore del parametro  $\beta$ ; l'equazione (2.2) può essere riscritta in funzione dell'errore  $\widehat{I}_N - I$

$$Pr \left\{ -\beta\sigma_{\widehat{I}_N} \leq \widehat{I}_N - I \leq +\beta\sigma_{\widehat{I}_N} \right\} = 1 - \alpha$$

dove  $\sigma_{\widehat{I}_N} = \sigma_x/\sqrt{N}$  con  $\sigma_x$  determinata dalla (2.1). Assumendo che l'errore  $\widehat{I}_N - I$  sia una variabile aleatoria gaussiana, la funzione densità di probabilità di  $\widehat{I}_N - I$  è approssimata da

$$\frac{1}{\sqrt{2\pi}\sigma_{\widehat{I}_N}} \exp \left( -\frac{(\widehat{I}_N - I)^2}{2\sigma_{\widehat{I}_N}^2} \right)$$

Notiamo che  $\widehat{I}_N - I$  è una variabile aleatoria a media nulla, poichè lo stimatore  $\widehat{I}_N$  è unbiased. Segue che

$$Pr \left\{ \widehat{I}_N - I \geq \beta\sigma_{\widehat{I}_N} \right\} = \frac{1}{\sqrt{2\pi}\sigma_{\widehat{I}_N}} \int_{\beta\sigma_{\widehat{I}_N}}^{\infty} \exp \left( -\frac{t^2}{2\sigma_{\widehat{I}_N}^2} \right) dt$$

Con un semplice cambio di variabile,  $y = t/\sigma_{\widehat{I}_N}$ , otteniamo

$$Pr \left\{ \widehat{I}_N - I \geq \beta\sigma_{\widehat{I}_N} \right\} = \frac{1}{\sqrt{2\pi}} \int_{\beta}^{\infty} \exp(-y^2/2) dy = Q(\beta)$$

dove  $Q(\cdot)$  rappresenta la coda di una gaussiana. Dalla figura 2.1 segue

$$Pr \left\{ \widehat{I}_N - I \geq \beta\sigma_{\widehat{I}_N} \right\} = Q(\beta) = \frac{\alpha}{2}$$

quindi

$$\beta = Q^{-1}\left(\frac{\alpha}{2}\right)$$

Come è evidente dalla figura 2.1, la probabilità che la stima di  $I$  cada nell'intervallo  $I \pm Q^{-1}(\alpha/2)\sigma_x/\sqrt{N}$  è  $1 - \alpha$ , dove  $\sigma_x$  è data dalla (2.1). Le quantità  $\pm Q^{-1}(\alpha/2)\sigma_x/\sqrt{N}$  determinano i limiti superiore ed inferiore dell'intervallo di confidenza, ovviamente  $\sigma_x$  deve essere nota.

## 2.4 Metodi asintotici e metodi Markov chain Monte Carlo

I metodi di Monte Carlo trovano, infine, importanti applicazioni nei problemi di calcolo del max e/o del minimo di una funzione. Sia  $g(x)$  una funzione concava con un solo punto di massimo, ciò che vogliamo calcolare è  $x^* = \max[g(x)]$ . Innanzitutto ci interessa calcolare

$$L(\lambda) = \int_{\mathbb{R}} h(x) \exp[\lambda g(x^*)] dx$$

utilizzando uno sviluppo in serie di Taylor troncato al secondo ordine per approssimare il comportamento della funzione  $g(x)$  in un intorno del punto  $x^*$ , otteniamo

$$L(\lambda) = \int_{\mathfrak{R}} h(x) \exp \left[ \lambda g(x^*) + \lambda g'(x^*)(x - x^*) + \frac{\lambda}{2} g''(x^*)(x - x^*)^2 \right] dx \quad (2.3)$$

dove  $g'(x^*)$  rappresenta la tangente alla curva nel punto di massimo,  $x^*$ , che è nulla. La (2.3) si può dunque riscrivere

$$L(\lambda) = \exp[\lambda g(x^*)] \int_{\mathfrak{R}} h(x) \exp \left[ \frac{\lambda}{2} g''(x^*)(x - x^*)^2 \right] dx$$

dove l'espressione  $\exp \left[ \frac{\lambda}{2} g''(x^*)(x - x^*)^2 \right]$  ricorda la funzione densità di probabilità di una gaussiana:  $f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{1}{2} \frac{(x-\bar{x})^2}{\sigma^2} \right]$ ; ponendo  $\bar{x} = x^*$  e, essendo  $g''(x^*) < 0$ ,  $\sigma^2 = -\frac{1}{\lambda g''(x^*)}$ , con qualche passaggio otteniamo

$$\begin{aligned} L(\lambda) &= \exp[\lambda g(x^*)] \int_{\mathfrak{R}} h(x) \exp \left[ \frac{\lambda}{2} g''(x^*)(x - x^*)^2 \right] dx \\ &= \sqrt{2\pi\sigma^2} \exp[\lambda g(x^*)] \int_{\mathfrak{R}} h(x) \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[ -\frac{1}{2} \frac{(x-x^*)^2}{\sigma^2} \right] dx \\ &= \sqrt{2\pi \left[ -\frac{1}{\lambda g''(x^*)} \right]} \exp[\lambda g(x^*)] E[h(x)] \end{aligned}$$

tenendo conto del fatto che  $X \sim N(x^*, \sigma^2)$ . Se ci poniamo nel caso particolare  $h(x) = x$  e  $E[h(x)] = E(x) = x^*$ , otteniamo

$$L(\lambda) = \int_{\mathfrak{R}} x \exp[\lambda g(x)] dx \cong \sqrt{2\pi \left[ -\frac{1}{\lambda g''(x^*)} \right]} \exp[\lambda g(x^*)] x^* \quad (2.4)$$

Poichè la quantità  $\int \exp[\lambda g(u)] du$  è costante una volta calcolato l'integrale, normalizzo  $L(\lambda)$

$$H(\lambda) = \frac{L(\lambda)}{\int \exp[\lambda g(u)] du} = \frac{\int x \exp[\lambda g(x)] dx}{\int \exp[\lambda g(u)] du} \quad (2.5)$$

Esaminando la quantità  $f_X(x) = \frac{\exp[\lambda g(x)]}{\int \exp[\lambda g(u)] du} \geq 0 \quad \forall x$  ed integrando osserviamo che  $\int f_X(x) dx = 1$ , ossia che  $f_X(x)$  è una densità di probabilità. Attraverso una serie di passaggi, a partire dalla (2.5), otteniamo

$$\begin{aligned} H(\lambda) &= \int x f_X(x) dx \\ &= \frac{1}{\int \exp[\lambda g(u)] du} \int x \exp[\lambda g(x)] dx \\ &= \frac{\exp[\lambda g(x^*)] x^* \sqrt{2\pi \left[ -\frac{1}{\lambda g''(x^*)} \right]}}{\exp[\lambda g(x^*)] \sqrt{2\pi \left[ -\frac{1}{\lambda g''(x^*)} \right]}} \\ &= x^* \end{aligned}$$

osservando che  $\int \exp[\lambda g(u)] du = L(\lambda)$  per la (2.4), imponendo  $h(x) = 1$  e, di conseguenza,  $x^* = 1$ . Quindi

$$H(\lambda) = \int x \frac{\exp[\lambda g(x)] dx}{\int \exp[\lambda g(u)] du} = x^*$$

ma  $x^*$  è il valore atteso di una variabile aleatoria. Se quindi considero  $n$  variabili aleatorie iid  $X_1, \dots, X_n$ , aventi densità di probabilità  $f_X(x) = \frac{\exp[\lambda g(x)]}{\int \exp[\lambda g(u)] du}$  e definisco  $x^* = \frac{1}{n} \sum_{i=1}^n X_i$ ; per la *legge dei grandi numeri*  $x^* \rightarrow E(X) = E(X_i)$  per  $n \rightarrow \infty$ . Naturalmente, affinché ciò sia possibile devo avere un generatore di numeri casuali con densità di probabilità  $f_X(x)$ , detta densità di Gibbs. I metodi che si servono di questo tipo di approccio sono detti *metodi asintotici*, o anche metodi di Laplace, poichè  $L(\lambda) = \int_{\mathbb{R}} h(x) \exp[\lambda g(x)]$  rappresenta la trasformata di Laplace della funzione  $h(x)$  se  $g(x) = x$  e per il fatto che la stima che poi si va effettivamente a considerare è  $\lim_{\lambda \rightarrow \infty} L(\lambda)$ . Tuttavia generare variabili aleatorie con una funzione densità di probabilità del tipo  $f_X(x)$  è tutt'altro che semplice: il problema maggiore è rappresentato dal fattore di normalizzazione  $\int \exp[\lambda g(u)] du$ .

Un approccio alternativo per fronteggiare lo stesso tipo di problemi si basa sulla generazione di una catena di Markov in cui le probabilità di transizione da uno stato all'altro dipendono solo da  $\exp[\lambda g(x)]$ : asintoticamente la catena si stabilizza e diventa una sequenza di variabili aleatorie iid con la densità di probabilità,  $f_X(x)$ , voluta; tuttavia, le variabili aleatorie di una catena di Markov non sono indipendenti, quindi bisogna attendere che la catena si stabilizzi e poi simulare il problema con i *Metodi Markov Chain Monte Carlo*, MCMC. Gli MCMC sono particolarmente utili per simulare il comportamento di componenti che dipendono da fattori di normalizzazione poco noti, difficili da calcolare, come nel caso dell'espressione  $\int \exp[\lambda g(u)] du$ .

## Capitolo 3

# Applicazioni ai sistemi di comunicazione

### 3.1 Canale AWGN

Un qualsiasi sistema di trasmissione può essere modellato come indicato nella figura 3.1

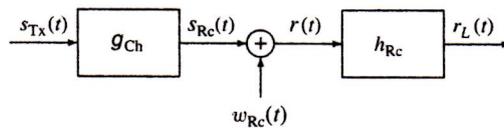


Figura 3.1: Modello di un sistema di trasmissione

Un primo blocco rappresenta il canale fisico, che può essere di vario tipo, attraverso cui viene la trasmissione del segnale  $s_{T_x}(t)$ , ossia di ciò che vogliamo trasmettere; il canale solitamente introduce una significativa attenuazione del segnale in ingresso restituendo  $s_{R_c}(t)$ . A questo si somma il rumore  $w_{R_c}(t)$  originando il segnale  $r(t)$  ricevuto in ingresso dal secondo blocco che rappresenta il ricevitore; quest'ultimo si comporta come un filtro caratterizzato da risposta impulsiva  $h_{R_c}(t)$ , restituendo in uscita il segnale  $r_L(t)$ , che è quello che noi effettivamente riceviamo.

Riassumendo

$$s_{R_c}(t) = (g_{Ch} * s_{T_x})(t) \quad \text{e} \quad r(t) = s_{R_c}(t) + w_{R_c}(t)$$

dove  $g_{Ch}(t)$  è la risposta impulsiva del canale.

Nelle simulazioni che affronteremo ci porremo in condizioni ideali, assumeremo cioè che il canale sia AWGN, ossia caratterizzato da rumore *gaussiano, bianco e additivo*; bianco significa che la sua densità spettrale di potenza (che indica come la potenza del segnale si distribuisce al variare della frequenza) è costante, ossia piatta, nella banda di interesse; gaussiano poichè la sua funzione densità di

probabilità è una gaussiana; infine, additivo poichè due rumori di origine anche diversa si sommano originando un unico disturbo.

Sotto tali ipotesi e considerando un segnale passa banda con banda stretta, in modo da assicurare che tutti i parametri dipendenti dalla frequenza siano costanti, il rapporto segnale rumore, SNR, che esprime il rapporto tra la potenza del segnale e la potenza del rumore si può scrivere come

$$SNR = \frac{p_{s_{Rc}}}{p_{w_{Rc}}} = \frac{P_{s_{Rc}}}{P_{w_{Rc}}}$$

dove le  $p$  rappresentano le potenze elettriche e le  $P$  le potenze statistiche; l'uguaglianza è resa possibile dalle ipotesi assunte che garantiscono la costanza dei parametri in frequenza.

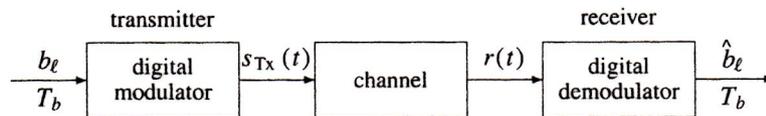


Figura 3.2: Schema di trasmissione digitale

Con il termine *modulazione* si intende il processo di trasformazione dell'informazione generata da una sorgente in un segnale, detto forma d'onda, che è adatto per essere trasmesso attraverso un canale di comunicazione con lo scopo di convogliarlo ad un ricevitore. Quando l'informazione è rappresentata da una sequenza di bits,  $b_l$ , si parla di *modulazione digitale*; lo strumento che esegue la trasformazione della sequenza di bits in un segnale continuo nel tempo,  $s_{Tx}(t)$ , è detto modulatore digitale; quest'ultimo impiega due forme d'onda per generare  $s_{Tx}(t)$ .

Il canale modifica il segnale trasmesso introducendo distorsione, interferenza e rumore; in generale

$$r(t) = s_{Rc}(t) + w_{Rc}(t), \quad s_{Rc}(t) = (g_{Ch} * s_{Tx})(t)$$

dove  $w_{Rc}(t)$  è rumore AWGN con media nulla e densità spettrale di potenza costante,  $P_{w_{Rc}}(f) = N_0/2$ .

Infine, dalla figura 3.2 traspare come il canale sia seguito da un blocco, il demodulatore digitale, ossia il ricevitore, con il compito di decifrare i bits originariamente inviati sulla base del segnale ricevuto,  $r(t)$ .

A causa della distorsione e del rumore introdotti dal mezzo di trasmissione, la sequenza binaria decifrata dal ricevitore,  $\hat{b}_l$  può contenere errori, ossia  $\hat{b}_l \neq b_l$  per qualche valore di  $l$ . Ci sono diverse tecniche di modulazione le cui prestazioni sono valutate in base alla *probabilità di errore sul bit*,  $Pr_{bit} = Pr[\hat{b}_l \neq b_l]$ .

Per stimare le prestazioni di un sistema di comunicazione digitale usando una simulazione Monte Carlo,  $N$  simboli vengono fatti passare attraverso il sistema (più precisamente attraverso un modello del sistema simulato da un computer) e viene conteggiato il numero di errori di trasmissione  $N_E$ . Se ci sono  $N_E$  errori nella trasmissione di  $N$  simboli, lo stimatore della probabilità di errore è

$$\widehat{P}_E = \frac{N_E}{N}$$

È importante analizzare le proprietà dello stimatore, ossia se è unbiased e consistente.

In un ambiente AWGN gli errori che si verificano sono indipendenti e il numero di errori  $N_E$  nella trasmissione di  $N$  simboli è descritto da una distribuzione binomiale. Sofferamoci ad analizzare il comportamento statistico di  $\widehat{P}_E$ ; iniziamo determinandone media e varianza: la probabilità di avere  $k$  errori nella trasmissione di  $N$  simboli è data da

$$p_{N_E}(k) = \binom{N}{k} P_E^k (1 - P_E)^{N-k}$$

dove  $\binom{N}{k}$  è un coefficiente binomiale e  $P_E$  è la probabilità di errore in una singola trasmissione.

La media e la varianza di una variabile aleatoria avente distribuzione binomiale sono

$$E(N_E) = NP_E \quad \text{e} \quad \text{var}(N_E) = \sigma_{N_E}^2 = NP_E(1 - P_E)$$

Di conseguenza, la media dello stimatore di Monte Carlo per la probabilità d'errore è

$$E(\widehat{P}_E) = \frac{E(N_E)}{N} = \frac{NP_E}{N} = P_E \quad (3.1)$$

ossia lo stimatore è unbiased.

La varianza dello stimatore di Monte Carlo per la probabilità d'errore è

$$\sigma_{\widehat{P}_E}^2 = \frac{\sigma_{N_E}^2}{N^2} = \frac{P_E(1 - P_E)}{N} \quad (3.2)$$

che mostra che lo stimatore è consistente, in quanto la varianza converge a 0 per  $N \rightarrow \infty$ . È importante sottolineare che le formule (3.1) e (3.2) sono valide solo assumendo che gli errori sono eventi tra loro indipendenti.

Essendo lo stimatore unbiased, sappiamo che in media la simulazione Monte Carlo fornisce un risultato preciso; inoltre, sarebbe utile che lo stimatore avesse una varianza piccola, in tal modo le stime giacerebbero nelle vicinanze del valore reale che stiamo stimando. Se lo stimatore è unbiased e consistente sappiamo, infine, che per ridurre la varianza dello stimatore è necessario simulare un gran numero di trasmissioni.

## 3.2 Binary Phase Shift Keying, BPSK

### 3.2.1 BPSK

La *Phase Shift Keying* è un tipo di modulazione digitale passa-banda. Sia  $h_{Tx}(t)$  un impulso ad energia finita,  $E_h$ , e  $\varphi_0$  una fase arbitraria; in generale

$$\varphi_n = \frac{\pi}{M}(2n - 1) + \varphi_0, \quad n = 1, \dots, M$$

e la generica forma d'onda trasmessa è data da

$$s_n(t) = h_{Tx}(t) \cos(2\pi f_0 t + \varphi_n) \quad (3.3)$$

cioè il segnale è ottenuto scegliendo uno degli  $M$  possibili valori di fase di un segnale sinusoidale con frequenza  $f_0$ , modulato da  $h_{Tx}(t)$ . Un'espressione alternativa alla (3.3) è

$$s_n(t) = \cos(\varphi_n)h_{Tx}(t) \cos(2\pi f_0 t) - \sin(\varphi_n)h_{Tx}(t) \sin(2\pi f_0 t) \quad (3.4)$$

$s_n(t)$  è quindi la combinazione lineare di due segnali l'uno modulato sul seno e l'altro modulato sul coseno, ossia di due portanti in quadratura, in quanto sfasate di  $\pi/2$ . L'energia dell'impulso modulante,  $h_{Tx}(t)$ , viene dunque divisa tra le due sinusoidi; una base ortonormale è data da

$$\begin{aligned} \phi_1(t) &= +\sqrt{\frac{2}{E_h}}h_{Tx}(t) \cos(2\pi f_0 t) \\ \phi_2(t) &= -\sqrt{\frac{2}{E_h}}h_{Tx}(t) \sin(2\pi f_0 t) \end{aligned}$$

La rappresentazione vettoriale delle forme d'onda è

$$s_n = \sqrt{\frac{E_h}{2}}[\cos(\varphi_n), \sin(\varphi_n)]$$

queste ultime giacciono su di una circonferenza di raggio  $\sqrt{\frac{E_h}{2}}$  e stanno tra loro ad una distanza angolare pari a  $\varphi_n$ .

Nel caso della *Binary Phase Shift Keying*  $M = 2$ , perciò  $\varphi_1 = \varphi_0$  e  $\varphi_2 = \pi + \varphi_0$ , quindi la base ortonormale è costituita da un unico elemento

$$\phi_1(t) = \sqrt{\frac{2}{E_h}}h_{Tx}(t) \cos(2\pi f_0 t + \varphi_0)$$

ossia lo spazio di segnale è unidimensionale, e la rappresentazione vettoriale delle forme d'onda è

$$s_1 = \sqrt{\frac{E_h}{2}}, \quad s_2 = -\sqrt{\frac{E_h}{2}}$$

La BPSK è una modulazione antipodale, ossia  $s_2(t) = -s_1(t)$ , e la probabilità di errore è

$$Pr_{bit} = Q\left(\sqrt{\frac{2E_s}{N_0}}\right)$$

dove  $E_s = E_b/2$  è l'energia della singola forma d'onda. Infine, poichè  $p_{sRc} = P_{sRc} = \frac{E_b}{2T}$  e  $\sigma_{wRc}^2 = \frac{N_0 B}{2}$ , dove  $B = \frac{1}{T}$ , posso scrivere

$$Pr_{bit} = Q\left(\sqrt{2SNR_{BPSK}}\right)$$

### 3.2.2 Simulazione 1

Prima di svolgere la nostra prima simulazione assumiamo alcune ipotesi semplificative che:

- non vi sia pulse shaping al trasmettitore;
- il canale sia AWGN;
- i simboli generati dalla sorgente siano indipendenti ed equiprobabili;
- non vi siano filtri nel sistema e, quindi, non vi sia interferenza intersimbolo.

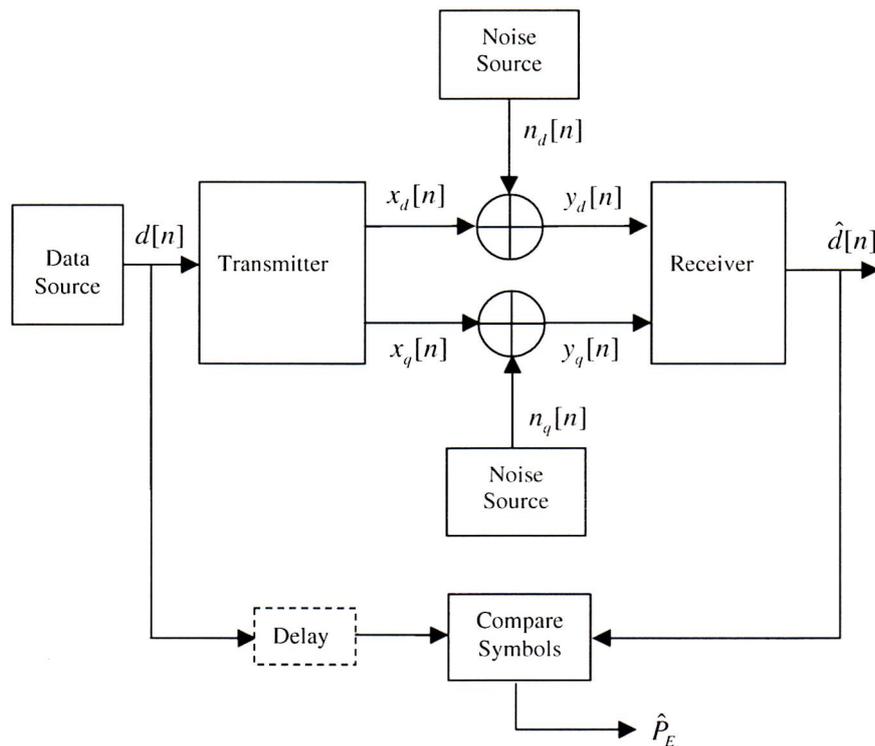


Figura 3.3: Modello di simulazione per un semplice sistema di comunicazione

Il basilare modello di simulazione è illustrato nella figura 3.3; a causa dell'assenza di filtri il ritardo attraverso il sistema è nullo, quindi possiamo trascurare

il rispettivo blocco nella nostra simulazione. Con le precedenti ipotesi, l'unica fonte di errore è il rumore del canale di comunicazione. Distinguiamo, dunque, le componenti *diretta* e *in quadratura* del segnale trasmesso, indicate rispettivamente con  $x_d(t)$  e  $x_q(t)$ ; il vantaggio di questo approccio consiste nel poter specificare le componenti dello spazio di segnale usando un singolo campione per ogni simbolo trasmesso.

In questo modo, il segnale passa-banda all'uscita del modulatore può essere espresso come

$$x(t, n) = A_c \cos \left[ 2\pi f_c t + k_m d[n] + \theta \right]$$

dove  $A_c$  rappresenta l'ampiezza dell'onda portante,  $k_m$  è una costante che dipende dal tipo di modulazione,  $d[n]$  è l' $n$ -esimo simbolo trasmesso ( $d[n] = 0$  o  $1$ ), e  $\theta$  è la fase di riferimento, che assumeremo nulla. La complex envelope di  $x(t, n)$  è funzione solo dell'indice  $n$  ed è data da

$$\tilde{x}[n] = A_c \exp \{ k_m d[n] + \theta \}$$

Di conseguenza, le componenti sono

$$x_d[n] = A_c \cos(k_m d[n]) \quad (3.5)$$

e

$$x_q[n] = A_c \sin(k_m d[n]) \quad (3.6)$$

Per determinare la probabilità d'errore come funzione di  $E_b/N_0$ , il valore di  $E_b$  viene mantenuto costante e la potenza del rumore viene incrementata nella banda di interesse. Sappiamo che la varianza del rumore è legata alla densità spettrale di potenza del rumore

$$\sigma_n^2 = \frac{N_0 f_s}{2}, \text{ ossia } N_0 = \sigma_n^2 \frac{2}{f_s}$$

dove  $f_s$  è la frequenza di campionamento. Infine, il rapporto segnale-rumore, SNR, è definito come  $E_b/N_0$  quindi

$$SNR = \frac{f_s E_b}{2\sigma_n^2}$$

Se l'energia e la frequenza di campionamento sono entrambe normalizzate ad 1, otteniamo

$$\sigma_n = \sqrt{\frac{1}{2} \frac{1}{SNR}}$$

questa espressione sarà utile per determinare la deviazione standard del rumore nelle simulazioni.

Per generare le componenti diretta e in quadratura nello spazio di segnale di un segnale soggetto ad una modulazione BPSK, imponiamo  $A_c = 1$  e  $k_m = \pi$ .

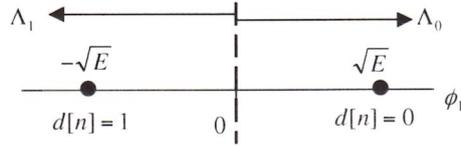


Figura 3.4: Spazio di segnale di una BPSK

Dalla (3.5) e dalla (3.6) otteniamo

$$x_d[n] = \cos(\pi d[n]) = \begin{cases} 1, & d[n] = 0 \\ -1, & d[n] = 1 \end{cases}$$

Inoltre, essendo

$$x_q[n] = \sin(\pi d[n])$$

$x_q[n] = 0$  per qualsiasi valore di  $d[n]$ . Perciò lo spazio di segnale è unidimensionale, come precedentemente illustrato al paragrafo 3.2.1, quindi solo le componenti dirette del segnale e del rumore devono essere generate nella simulazione. La figura illustra anche le regioni di decisione,  $\Lambda_0$  e  $\Lambda_1$ , se il segnale cade nella regione  $\Lambda_0$  il ricevitore prende la decisione  $\hat{d}[n] = 0$ , se il segnale cade, invece, nella regione  $\Lambda_1$  prende la decisione  $\hat{d}[n] = 1$ . La soglia tra le due regioni è zero, come accade sempre nel caso di segnali equiprobabili, con la stessa energia, in un ambiente AWGN.

La regola di decisione è perciò

$$\hat{d}[n] = \begin{cases} 0, & y_d[n] > 0 \\ 1, & y_d[n] < 0 \end{cases}$$

dove  $y_d[n]$  è la componente diretta del segnale ricevuto.

### 3.2.3 Analisi dei risultati

Utilizzando il programma Matlab *BPSK.m*, presente nell'appendice A.1, per implementare la simulazione, si ottengono i grafici riportati in seguito; esaminandoli è evidente la conferma di quanto affermato in precedenza, ossia che per ottenere una buona simulazione del comportamento dell'evento di interesse lo stimatore deve essere consistente; in altre parole deve avere varianza piccola e, poichè nel caso esaminato  $\sigma_{P_E}^2 = \frac{P_E(1-P_E)}{N}$ , come dimostrato al paragrafo 3.1, quest'ultima diminuisce all'aumentare del numero di campioni esaminati: per N piccolo, vedi le figure 3.5 e 3.6, la stima si discosta sensibilmente dalla previsione teorica, mentre all'aumentare di N i due grafici si avvicinano sempre di più sino quasi a sovrapporsi per valori di N molto elevati, vedi figura 3.10.

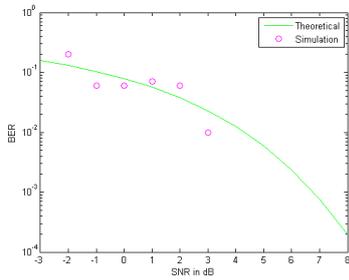


Figura 3.5:  $N = 100$

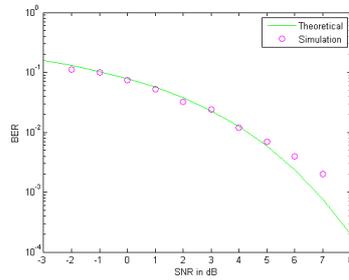


Figura 3.6:  $N = 1000$

Figura 3.7: BER in funzione di SNR in dB per piccoli valori di  $N$

Inoltre, è possibile notare che l'affidabilità dello stimatore diminuisce all'aumentare del rapporto segnale-rumore, ciò è dovuto al fatto che il numero degli errori conteggiati diminuisce: una possibile soluzione sarebbe mettere in relazione  $N$  e SNR oppure eseguire la simulazione fino a quando ad ogni valore di SNR corrisponde lo stesso numero di errori conteggiati.

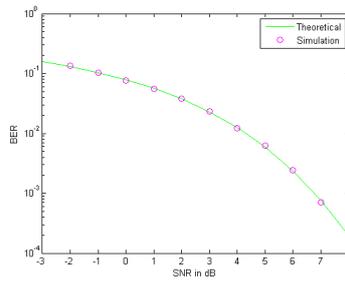


Figura 3.8:  $N = 10000$

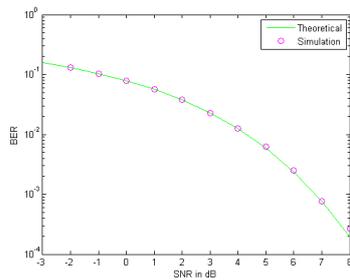


Figura 3.9:  $N = 100000$

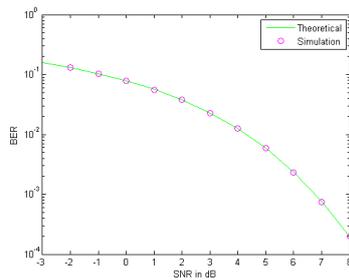


Figura 3.10:  $N = 1000000$

Figura 3.11: BER in funzione di SNR in dB per elevati valori di  $N$

## 3.3 Binary Frequency Shift Keying, BFSK

### 3.3.1 BFSK

La *Binary Frequency Shift Keying* è un tipo di modulazione la cui chiave è lo spostamento in frequenza. Tipicamente le due forme d'onda utilizzate sono

$$s_1(t) = \begin{cases} A \cos(2\pi(f_0 - f_d)t + \varphi_0), & \text{se } 0 < t < T \\ 0, & \text{altrimenti} \end{cases}$$
$$s_2(t) = \begin{cases} A \cos(2\pi(f_0 + f_d)t + \varphi_0), & \text{se } 0 < t < T \\ 0, & \text{altrimenti} \end{cases}$$

dove  $\varphi_0$  è una fase arbitraria.

Le due forme d'onda hanno la stessa energia,  $E_1 = E_2 \cong \frac{1}{2}A^2T$ , dove A e T sono rispettivamente l'ampiezza ed il periodo delle due sinusoidi.

Infine, la probabilità d'errore è data da

$$Pr_{bit} = Q\left(\sqrt{\frac{A^2T[1 - \text{sinc}(4f_dT)]}{2N_0}}\right)$$

### 3.3.2 Simulazione 2

Assumiamo le stesse ipotesi elencate nel paragrafo 3.2.2 e riteniamo valide le considerazioni preliminari fatte per la simulazione 1; per generare le componenti diretta e in quadratura nello spazio di segnale di un segnale soggetto ad una modulazione BFSK, imponiamo  $A_c = 1$  e  $k_m = \pi/2$ . Dalla (3.5) e dalla (3.6) otteniamo

$$x_d[n] = \begin{cases} 1, & d[n] = 0 \\ 0, & d[n] = 1 \end{cases}$$

e

$$x_q[n] = \begin{cases} 0, & d[n] = 0 \\ 1, & d[n] = 1 \end{cases}$$

Perciò lo spazio di segnale è bidimensionale, quindi entrambe le componenti diretta e in quadratura del segnale e del rumore devono essere generate nella simulazione. La figura illustra anche le regioni di decisione,  $\Lambda_0$  e  $\Lambda_1$ , se il segnale cade nella regione  $\Lambda_0$  il ricevitore prende la decisione  $\hat{d}[n] = 0$ , se il segnale cade, invece, nella regione  $\Lambda_1$  prende la decisione  $\hat{d}[n] = 1$ .

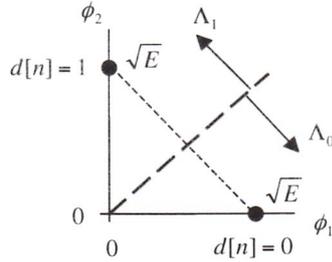


Figura 3.12: Spazio di segnale di una BFSK

La regola di decisione è perciò

$$\hat{d}[n] = \begin{cases} 0, & y_d[n] > y_q[n] \\ 1, & y_d[n] < y_q[n] \end{cases}$$

dove  $y_d[n]$  e  $y_q[n]$  sono rispettivamente le componenti diretta e in quadratura del segnale ricevuto.

### 3.3.3 Analisi dei risultati

Utilizzando il programma Matlab *BFSK.m*, presente nell'appendice A.2, per implementare la simulazione, si ottengono i seguenti grafici:

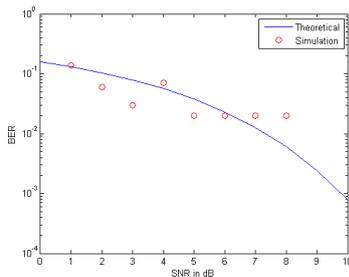


Figura 3.13:  $N = 100$

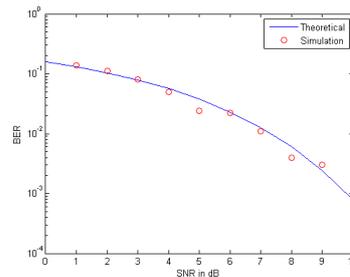


Figura 3.14:  $N = 1000$

Figura 3.15: BER in funzione di SNR in dB per valori piccoli di N

In questo caso valgono le stesse considerazioni fatte per la precedente simulazione: dai grafici emerge l'importanza che lo stimatore sia consistente; infatti, poichè anche in questo caso  $\sigma_{\hat{P}_E}^2 = \frac{P_E(1-P_E)}{N}$ , come dimostrato al paragrafo

3.1, la varianza dello stimatore diminuisce all'aumentare del numero di campioni esaminati: per  $N$  piccolo, vedi le figure 3.13 e 3.14, la stima si discosta sensibilmente dalla previsione teorica, mentre all'aumentare di  $N$  i due grafici si avvicinano sempre di più sino quasi a sovrapporsi per valori di  $N$  molto elevati, vedi figura 3.18. Infine, ancora una volta l'affidabilità dello stimatore diminuisce all'aumentare del rapporto segnale-rumore, a causa del minor numero di errori conteggiati; possibili azioni correttive sono analoghe a quelle precedentemente esposte nel paragrafo 3.2.3.

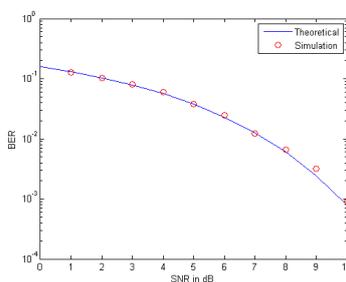


Figura 3.16:  $N = 10000$

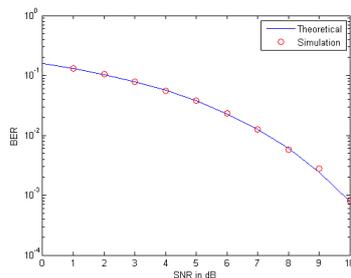


Figura 3.17:  $N = 100000$

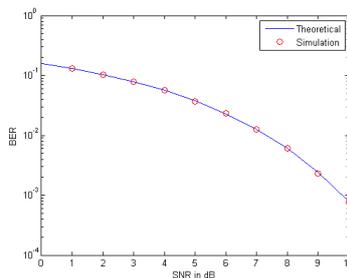


Figura 3.18:  $N = 1000000$

Figura 3.19: BER in funzione di SNR in dB per valori elevati di  $N$

### 3.4 Integrazione Monte Carlo in un canale AWGN

L'integrazione Monte Carlo è uno strumento molto utile nell'ambito delle telecomunicazioni. Nel processo di demodulazione il ricevitore deve decifrare il segnale ricevuto mediante la cosiddetta regola di decisione: questa si basa sulla suddivisione dello spazio di segnale in in  $M$  regioni di decisione, dove  $M$  è il numero di segnali modulanti; se il segnale ricevuto appartiene alla regione  $j$ -esima,

allora il ricevitore restituirà come risultato il segnale corrispondente al  $j$ -esimo segnale modulante. In un canale AWGN le regioni di decisione sono legate alla probabilità condizionata di trasmettere correttamente o meno il simbolo in ingresso, e quest'ultima è una gaussiana con media determinata dal simbolo trasmesso e varianza determinata dal rumore del canale. Infatti la probabilità di trasmettere correttamente è data da

$$Pr(C) = \sum_{j=1}^M Pr(\rho \in R_j | a_0 = j) p_j = \sum_{j=1}^M \int_{R_j} p_{r|a_0}(\rho|j) p_j d\rho$$

dove  $a_0$  è il simbolo trasmesso,  $p_{r|a_0}$  è la probabilità condizionata di ricevere  $r$  dato che ho trasmesso  $a_0$ , e  $p_j$  è la probabilità a priori di trasmettere il simbolo. Le densità di probabilità condizionate, da  $d[n] = 0$  e  $d[n] = 1$ , sono rappresentate nella figura 3.20, dove  $k_T$  è la soglia del ricevitore.

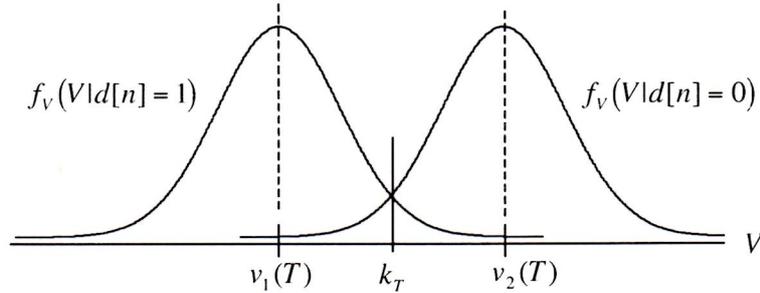


Figura 3.20: Funzioni densità di probabilità condizionata in modulazione binaria con rumore gaussiano

La probabilità di errore condizionata, per  $d[n] = 1$

$$Pr(E|d[n] = 1) = \int_{k_T}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{(x - \nu_1(T))^2}{2\sigma_n^2}\right] dx \quad (3.7)$$

e analogamente

$$Pr(E|d[n] = 0) = \int_{-\infty}^{k_T} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left[-\frac{(x - \nu_2(T))^2}{2\sigma_n^2}\right] dx \quad (3.8)$$

Segue che la stima della probabilità di errore del sistema

$$P_E = \frac{1}{2} Pr(E|d[n] = 1) + \frac{1}{2} Pr(E|d[n] = 0)$$

coinvolge la stima del valore di un integrale.

### 3.4.1 Simulazione di Monte Carlo per il calcolo della coda di una $N(0,1)$

Dalla (3.7) e dalla (3.8) emerge chiaramente come la stima della probabilità di errore si riduca alla stima del valore di un integrale che non è altro che la coda di una gaussiana:  $Q(\alpha) = \int_{\alpha}^{\infty} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2) dx$ . In questo caso, i metodi di Monte Carlo forniscono un utile strumento per la stima di  $Q(\alpha)$ , con  $\alpha$  fissato; infatti  $Q(\alpha)$  si può anche riscrivere come

$$\begin{aligned} Q(\alpha) &= \int_{-\infty}^{+\infty} \mathbb{1}_{[\alpha, +\infty)}(y) f_Y(y) dy \\ &= \int_{\alpha}^{+\infty} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}y^2) dy \\ &= P(Y > \alpha) \end{aligned}$$

dove la v.a.  $Y \sim N(0,1)$ . Definito l'evento  $A = Y > \alpha$ , si può infine notare che

$$Q(\alpha) = E[X_A]$$

dove  $X_A$  è la v.a. indicatrice dell'evento  $A$ , definita nel seguente modo:

$$X_A(\omega) = \begin{cases} 1, & \omega \in A \\ 0, & \omega \notin A \end{cases}$$

ossia  $X_A(\omega) \sim B(P(A))$ , dove  $P(A) = P(X_A(\omega) = 1)$ ; quindi

$$E[X_A] = P(A) = P(Y > \alpha)$$

Se ora considero  $N$  v.a. iid  $X_{A_1}, \dots, X_{A_N} \sim B(P(Y_i > \alpha))$  per la legge dei grandi numeri

$$P(Y > \alpha) = E[X_{A_i}] \cong \frac{1}{N} \sum_{i=1}^N X_{A_i} = \frac{N[Y_1, \dots, Y_N > \alpha]}{N}$$

Nell'appendice A.3 è riportato un semplice programma Matlab, *CodaGauss.m*, che basandosi sui risultati appena illustrati e sfruttando le proprietà della funzione **rand** consente di calcolare con una discreta precisione il valore di  $Q(\alpha)$ , con  $\alpha$  fissato dall'utente.

La tabella seguente riporta alcuni valori di  $Q(\alpha)$  ottenuti mediante simulazione accanto ai valori effettivi:

$\alpha$	$Q(\alpha)$	Risultato simulazione
0	0,5000000	0,4999080
0,3	0,3821000	0,3831970
0,5	0,3085000	0,3103880
1	0,1587000	0,1608150
1,25	0,1056000	0,1069910
1,75	0,0400600	0,0400590
2	0,0227500	0,0223300
2,5	0,0062100	0,0055850
3	0,0013500	0,0010210
3,5	0,0002326	0,0001220

Esaminando la tabella è evidente come la precisione dei risultati ottenuti diminuisca all'aumentare di  $\alpha$ , questo è dovuto alla struttura stessa di una distribuzione normale standard che va rapidamente a zero, di conseguenza già per  $\alpha \geq 3$   $Q(\alpha)$  assume valori molto piccoli, dell'ordine di  $10^{-3}$  o  $10^{-4}$ , quindi per avere una buona approssimazione è necessario ricorrere a centinaia di milioni di campioni che Matlab non è in grado di supportare nel corso di una semplice simulazione.

# Appendice A

## Codice MATLAB

### A.1 BPSK.m

```
% simulazione BPSK %

clear all
snrdB_min = -3; % limite inferiore SNR in dB
snrdB_max = 8; % limite superiore SNR in dB
snrdB = snrdB_min:1:snrdB_max;
Nsymbols = input('Inserire un numero di simboli: ');
snr = 10.^(snrdB/10); % conversione da dB
h = waitbar(0, 'Simulazione in corso');
len_snr = length(snrdB);
for j=1:len_snr % incremento SNR
    waitbar(j/len_snr)
    sigma = sqrt(1/(2*snr(j))); % deviazione standard del rumore
    error_count = 0;
    for K=1:Nsymbols % inizio ciclo di simulazione
        d = round(rand(1)); % dato trasmesso
        x_d = 2*d - 1; % output del trasmettitore
        w_d = sigma*randn(1); % rumore
        y_d = x_d + w_d; % input del ricevitore
        if y_d > 0 % condizione del test
            d_est = 1; % regola di decisione
        else
            d_est = 0; % regola di decisione
        end
        if (d_est ~= d)
            error_count = error_count + 1; % contatore errori
        end
    end
    %fine ciclo di simulazione
    errors(j) = error_count; % memorizzazione conteggio errori
end
close(h)
```

```

ber_sim = errors/Nsymbols;          % stima BER (probabilità di errore)
ber_theor = qfunc(sqrt(2*snr));      % BER teorica
semilogy(snrdB,ber_theor,'g',snrdB,ber_sim,'om')
axis([snrdB_min snrdB_max, 0.0001 1])
xlabel('SNR in dB')
ylabel('BER')
legend('Theoretical','Simulation')

```

## A.2 BFSK.m

```

% simulazione BFSK %

clear all
snrdB_min = 0; % limite inferiore SNR in dB
snrdB_max = 10; % limite superiore SNR in dB
snrdB = snrdB_min:1:snrdB_max;
Nsymbols = input('Inserire un numero di simboli: ');
snr = 10.^(snrdB/10); % conversione da dB
h = waitbar(0, 'Simulazione in corso');
len_snr = length(snrdB);
for j=1:len_snr % incremento SNR
    waitbar(j/len_snr)
    sigma = sqrt(1/(2*snr(j))); % deviazione standard del rumore
    error_count = 0;
    for K=1:Nsymbols % inizio ciclo di simulazione
        d = round(rand(1)); % dato trasmesso
        if d == 0
            x_d = 1; % componente diretta output del trasmettitore
            x_q = 0; % componente in quadratura output del trasmettitore
        else
            x_d = 0;
            x_q = 1;
        end
        w_d = sigma*randn(1); % componente diretta rumore
        w_q = sigma*randn(1); % componente in quadratura rumore
        y_d = x_d + w_d; % componente diretta input del ricevitore
        y_q = x_q + w_q; % componente in quadratura input del ricevitore
        if y_d > y_q % condizione del test
            d_est = 0; % regola di decisione
        else
            d_est = 1; % regola di decisione
        end
        if (d_est ~= d)
            error_count = error_count + 1; % contatore errori
        end
    end
    errors(j) = error_count; % memorizzazione conteggio errori
end

```

```

end
close(h)
ber_sim = errors/Nsymbols;           % stima BER (probabilità di errore)
ber_theor = qfunc(sqrt(snr));        % BER teorica
semilogy(snrdB,ber_theor,snrdB,ber_sim,'or')
axis([snrdB_min snrdB_max, 0.0001 1])
xlabel('SNR in dB')
ylabel('BER')
legend('Theoretical','Simulation')

```

### A.3 CodaGauss.m

```

% simulazione di Monte Carlo per il calcolo della coda di una N(0,1) %

clear all
N = 10000000;           % Q(a) è molto piccolo quindi servono molti campioni
a = input('Inserire argomento della coda: ');
M = rand(N,12);        % creo Nx12 v.a. U(0,1)
counter = 0;           % inizializzo contatore
for i=1:N
    Y = 0;             % creo la v.a. Y
    for j=1:12        % calcolo Y in modo che sia una N(0,1)
        W = M(i,j) - 0.5;
        Y = Y + W;
    end
    end
    if Y > a           % valuto la condizione
        counter = counter + 1; % incremento il contatore
    end
end
Q = counter/N;        % valore stimato di Q
fprintf('Il valore desiderato è: %f\n', Q)

```

## Appendice B

# Richiami di teoria della probabilità

Fondamentali per comprendere il funzionamento del metodo di Monte Carlo sono alcuni concetti di base.

### B.1 Legge dei grandi numeri

**Teorema 1.** Sia  $X_1, \dots, X_N$  una sequenza di  $N$  variabili aleatorie iid, ossia indipendenti ed identicamente distribuite, aventi valore atteso  $E(X) = \mu$ , e sia  $\hat{X}_N = \frac{1}{N} \sum_{k=1}^N X_k$ , allora

$$\hat{X}_N \xrightarrow{P} \mu$$

ovvero  $\lim_{N \rightarrow \infty} Pr\left(|\hat{X}_N - \mu| > \epsilon\right) = 0, \quad \forall \epsilon > 0.$

### B.2 Variabili aleatorie Uniforme e Gaussiana

Due comuni variabili aleatorie continue, cui si ricorre spesso nei metodi di Monte Carlo, sono l'*uniforme* e la *gaussiana*, o *normale*. Per ciascuna c'è una versione standard e una versione generale che è facile esprimere in funzione di quella standard.

Una variabile aleatoria uniforme standard,  $U$ , ha densità di probabilità

$$f_U(u) = \begin{cases} 1, & \text{se } 0 \leq u \leq 1 \\ 0, & \text{altrimenti} \end{cases}$$

Essendo la densità costante,  $U$  è uniformemente distribuita nell'intervallo  $[0, 1]$ . Una variabile aleatoria,  $Y$ , uniformemente distribuita nel generico intervallo  $[a, b]$ , ha densità di probabilità

$$g_Y(y) = \begin{cases} \frac{1}{b-a}, & \text{se } a \leq y \leq b \\ 0, & \text{altrimenti} \end{cases}$$

Il valore atteso di  $Y \sim U(a, b)$  è

$$E(Y) = \frac{a+b}{2}$$

e la varianza

$$\text{var}(Y) = \frac{(b-a)^2}{12}$$

Una variabile aleatoria normale standard,  $Z \sim N(0, 1)$ , avente cioè media nulla e varianza unitaria, ha densità di probabilità

$$f_Z(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}$$

Mentre una variabile aleatoria normale generica,  $X \sim N(\mu, \sigma^2)$ , avente cioè media  $\mu$  e varianza  $\sigma^2$ , è data dall'espressione  $X = \sigma Z + \mu$  ed ha densità di probabilità

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$$

Il valore atteso di  $X \sim N(\mu, \sigma^2)$  è

$$E(X) = \mu$$

e la varianza

$$\text{var}(X) = \sigma^2$$

### B.3 Variabile aleatoria Binomiale

Importante è anche la variabile aleatoria discreta binomiale,  $X \sim \text{Bin}(n, p)$ , avente densità di probabilità

$$p_x(k) = \binom{n}{k} p^k (1-p)^{n-k}, \text{ con } k = 0, 1, 2, \dots, n$$

dove  $k$  è il numero di casi favorevoli, ossia il numero di occorrenze dell'evento che ci interessa, e  $n$  è il numero totale di osservazioni; mentre,  $p$  è la probabilità che l'evento di interesse si verifichi.

Il valore atteso di una  $X \sim \text{Bin}(n, p)$  è

$$E(X) = np$$

e la varianza

$$\text{var}(X) = np(1-p)$$

# Bibliografia

- [1] N. Metropolis, *The beginning of the Monte Carlo Method*, Los Alamos Science Special Issue, (1987), pp.125-130
- [2] N. Metropolis, S. Ulam, *The Monte Carlo Method*, Journal of the American Statistical Association, Vol.44, No.247.(1949), pp.335-341
- [3] Tranter, Shanmugan, Rappaport, Kosbar, *Principles of Communication Systems Simulation with Wireless Applications*, Prentice Hall, 2004, chapter 9
- [4] *Handbook on Simulation*, Ed.: Jerry Banks, Wiley, 1998, chapter 4
- [5] L. Vangelista, N. Laurenti, T. Erseghe, R. Corvaja, A. Zanella, M. Rossi, L. Badia, *Principles of Communications Networks and Systems*, Ed.: N. Benvenuto, M. Zorzi, Wiley, 2011, chapters 4-5
- [6] S. M. Ross, *Calcolo delle probabilità*, Apogeo, Milano 2007, capitoli 4-8