» Introduzione

Questi temi sono semplicemente un'elaborazione dei miei appunti personali ovviamente corretti e verificati. Vi possono servire solamente come riferimento per avere un'idea su cosa si stia parlando.
Il consiglio è quello di mettervi anche voi a farli e a scriverli… inserendo magari qualche formula e qualche particolare in più… perché con questi temi capite qualcosa, ma non passate l'esame.

Roberto

» Il ponte radio

Collegamenti possono essere backhaul (PDH dai 2 (E1) ai 34 (16xE1) Mbit/s) oppure backbone (SDH a 155 Mbit/s). Gli SDH si usano anche in sostituzione o in protezione ad altri collegamenti (magari in fibra ottica) nel caso questi ultimi si rompano.
L'installazione di un ponte radio richiede il rilascio da parte del Ministero di una licenza che ti assegni la frequenza. Infatti ci sono diverse bande che al loro interno contengono delle canalizzazioni suddivise a loro volta in ulteriori fette stabilite dalle norme ETSI. In base a queste norme ti viene data anche una maschera dalla quale il tuo spettro non può assolutamente uscire.

:: Lato trasmissione

L'antenna poi è collegata ad una ODU (outdoor unit) e tramite ad un coassiale ad una IDU (indoor unit). Nel caso di sistemi SDH è tutto indoor per questioni di raffreddamento.
L'IDU è composta da una interfaccia NxE1 e da informazioni di servizio, quali ad esempio il controllo automatico di potenza che regola autonomamente la potenza a seconda delle condizioni climatiche.
I dati che arrivano bianchi dal coassiale, vengono mappati secondo una codifica di Gray (16-QAM), filtrati con un filtro formatore (in genere a radice di Nyquist) che mi da lo spettro giusto da inserire nella maschera da poter trasmettere dopo di che si converte in analogico e si filtrano passa basso eventuali repliche, si modula il segnale e si trasmette.
Nell'utilizzo del filtro a radice di Nyquist devo stare attento ad effettuare un buon roff-off (il migliore è 0,2) perché altrimenti rischio, in frequenza, di uscire dalla maschera ETSI. Inoltre non posso usare un sinc classico perché avrei troppe moltiplicazioni da fare (e scaldano) e rischierei di sommare troppe code con gli impulsi vicini da saturare l'amplificatore. A tal proposito si utilizza il meccanismo di back-off tale per cui mando una potenza media all'amplificatore minore rispetto a quella di saturazione: tutto ciò è strettamente legato alla progettazione del filtro di trasmissione.
Nell'operazione di filtraggio, effettuata con un FIR, si fa precedere uno zero padding in modo da poter utilizzare un filtro meno critico in quanto le repliche sono isolate una dall'altra dagli zeri. Però devo stare attento a non esagerare perché poi ho il DAC ed aumentare la banda di conversione costa (e molto!) perciò si raggiunge un ZP di 4 che è un buon compromesso.
Un altro appunto è che la modulazione seno-coseno (Q-I) finale non deve avvenire ad alte frequenze perché, sebbene i due oscillatori siano collegati (è lo stesso solamente sfasato di 90°), a frequenze alte si rischia di avere anche un errore di fase, con uno sfasamento anche velocissimo che creerebbe enormi problemi in ricezione, onde per cui prima si modula e poi si porta a radio frequenza.

:: Lato ricezione

In ricezione mi arriva il segnale con tutti i suoi raggi riflessi. Con un dispositivo pregiato effettuo l'amplificazione a bassa cifra di rumore a larga banda, poi filtro la banda che mi interessa, riporto a basse frequenze il segnale ed inizio il processo di demodulazione. Converto in banda base (I-Q), amplifico (con un amplificatore a guadagno controllato in automatico, cioè varia se succede qualcosa, tipo piove…), filtro (come sempre dopo un'amplificazione) per evitare di ripiegare il rumore in banda e campiono il segnale accertandomi che ci sia perfetto sincronismo di simbolo. Poi si passa all'equalizzazione ed al decisore a soglia.

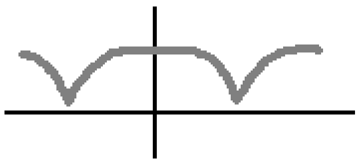
» Equalizzazione

L'equalizzatore serve per minimizzare i disturbi di ISI e del rumore. È un filtro che viene messo in lato ricezione dopo il campionatore (perciò è discreto) che deve compensare la risposta del canale.
Una volta ottenuta la risposta del canale in modo tale da avere una bassa probabilità di errore nella decisione dei simboli ricevuti, l'obiettivo dell'equalizzatore è di continuare ad inseguire e compensare eventuali variazioni del canale purché, però, queste siano abbastanza lente da consentire all'equalizzatore il tempo di stabilizzarsi.
Per determinare i coefficienti del filtro di equalizzazione si usano metodi diversi: zero forcing lineare, che è in grado di eliminare ISI, e MMSE (minimo errore quadratico medio).

$\{a_k\}$ -> $H_T(f)$ -> $H_C(f)$ + AWGN -> $H_R(f)$ -> campiono a kT -> $\{y_k\}$ -> C(z) equalizzatore e poi $\{x_k\}$ va al decisore e ridiventa $\{a_k\}$ stimato
raggruppo ed ho: $\{a_k\}$ -> G(f) -> campiono a kT -> $\{y_k\}$ e vado all'equalizzatore

:: Zero forcing lineare

Nel dominio delle trasformate zeta, affinché venga rimossa completamente l'ISI, voglio che G(z)*C(z)=1 e praticamente C(z)= 1/G(z) cioè l'inversione del sistema di trasmissione.



Solamente che quando ho zeri vicini al cerchio di raggio unitario il modulo di G(z) negli zeri tende ad annullarsi (e quindi annulla quella frequenza) e, in presenza di rumore, quando poi vado ad invertire la funzione ottengo un amplificazione eccessiva (al limite infinita, perché la V diventa un picco) del rumore: effetto di *noise enhancement*. Pertanto incrementando il rumore (la sua varianza), peggiorano le prestazioni: questo è il punto debole dello zero forcing.

Calcolo l'autocorrelazione $\psi_g(z)=G(z)G(z^{-1})^*$ per cui C(z)= 1/G(z)= $\frac{G(z^{-1})^*}{\psi_g(z)}$

:: MMSE (Minimo errore quadratico medio)

Per evitare il problema di prima (del rumore), sono disposto anche ad accettare un po' di ISI.
Definisco $J= E\{|x_k-a_k|^2\}= E\{|u_k|^2\}$ l'errore quadratico medio (MSE) e vedo di minimizzarlo. Approssimo u_k ad una variabile casuale gaussiana con media zero e varianza J e cerco il valore di C(z) che mi minimizza l'errore quadratico medio J.

L'espressione di C(z) che minimizza l'MSE è C(z)= $\frac{\sigma_a^2 G(z^{-1})^*}{\sigma_a^2 \psi_g(z) + No}$ dove si vede subito che stavolta è indirettamente proporzionale al rumore. Nel caso di No=0 riottengo lo zero forcing.

Come realizzo C(z)? teoricamente è un IIR che però da troppi problemi sia di instabilità che di tempo di elaborazione (devo aspettare che completi l'anello).
Meglio sarebbe un FIR che non ha problemi di tempo e funziona in serie, nel senso che ad ogni colpo di clock qualcosa entra nei registri e qualcos'altro esce. Non è importante che l'elaborazione avvenga in tempo reale, ma è fondamentale che ad ogni T entrino dei dati e ci sia sempre qualcosa in uscita.
Quando il FIR ha più di 32 prese (presa = coeff. del filtro) si preferisce passare in frequenza e lavorare sulla FFT.
Adesso devo trasformare la C(z) di prima che è IIR in una FIR. Un metodo per evitare di calcolare l'inverso della matrice di autocorrelazione, operazione sempre complessa, è quello di utilizzare l'algoritmo del gradiente.

:: Algoritmo del gradiente

Se **c** è il vettore dei coefficienti del filtro FIR ad ogni passo devo calcolare il c ottimo tramite un'operazione iterattiva tale per cui per trovare il vettore c_{j+1} mi baso su quello precedente c_j.
L'algoritmo del gradiente prevede proprio questo: calcolo il vettore al passo j+1 partendo da quello al passo j e muovendomi in senso opposto alla direzione di massima crescita (cioè guardo la derivata del vettore al passo precedente e se è negativa mi sposto a destra -verso lo zero- (se è positiva vai verso sinistra) di un piccolo passo). Dopo di che ricalcolo la derivata e continuo cosi fino a quando trovo lo zero ed ho l'ottimo.
All'ottimo ho che $E\left\{y_k^*(x_k - a_k)\right\}=0$

È, inoltre, possibile effettuare un'approssimazione del valore atteso $E\left\{y_k^*(x_k - a_k)\right\}$ calcolando una media temporale su N campioni e sommandoci un vettore casuale a media nulla. In questo modo si parlerà dell'algoritmo del gradiente stocastico che non necessita del bisogno di conoscere nulla riguardo al canale né riguardo alla matrice di autocorrellazione, ma solamente di conoscere la sequenza trasmessa (ovviamente tramite una corretta stima).

:: Equalizzazione in frequenza

Si svolge praticamente in due fasi: implementazione a blocchi di un FIR e calcolo veloce della convoluzione mediante FFT.
Suddivido la sequenza di $\{y_k\}$ in ingresso in blocchi di N campioni. Ad ogni istante temporale k, faccio entrare i vari blocchi nel FIR mantenendo i coefficienti fissi nell'istante di blocco, cambiandoli e aggiornandoli poi a blocchi anche loro all'istante di blocco successivo. Per passare poi in ambito frequenziale effettuo una FFT dei coefficienti del filtro (ottengo $\underline{c}(n)$); vado a prendere l'ultimo vettore del blocco e il primo e faccio l'FFT anche con loro (ottengo $\underline{y}(n)$), dopo di che antitrasformo mediante IFFT il prodotto tra $\underline{c}(n)$ e $\underline{y}(n)$. Occhio che sono tutti vettori.

» OFDM

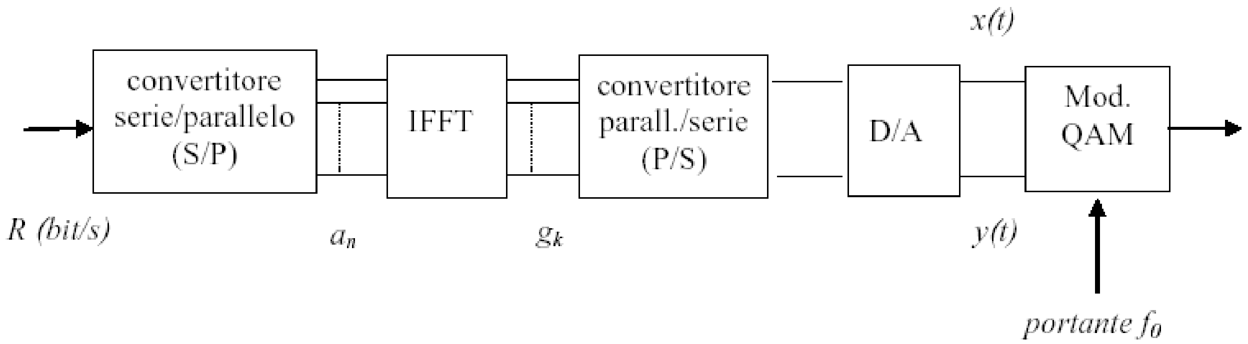
Venuta di moda adesso per W-Lan, DVB-T, DAB, W-Man, A-DSL…
L'idea di base della modulazione OFDM è quella di scomporre il flusso dei dati da trasmettere in N flussi in parallelo da trasmettere con un insieme di portanti spaziate Δf, in modo tale che non ci sia interferenza tra i flussi.
L'ortogonalità si ottiene con un T_S= N/R (R rate in trasmissione) dei simboli trasmessi sulle sottoportanti e con la relazione Δf= $1/T_s$
Con l'OFDM trasmetto N flussi ciascuno a velocità N/R in N sottobande larghe Δf= B/N ottenendo quindi una funzione di trasferimento del canale per ciascuna sottobanda abbastanza non distorcente permettendomi di semplificare le operazioni di equalizzazione.

Durante l'intervallo di simbolo T_S= N/R trasmetto un segnale che è praticamente la modulazione in ampiezza di N sottoportanti a frequenza $f_n = n/T_S$ con i simboli dell'informazione $a_0, a_1, \ldots a_{N-1}$.

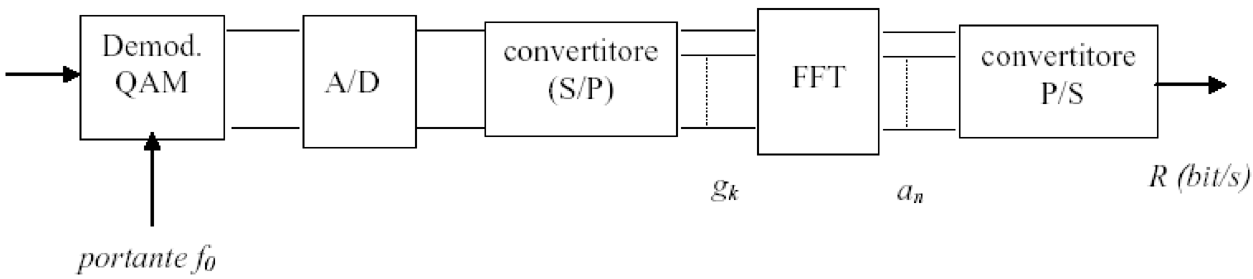
$$g(t) = \sum_n a_n e^{j2\pi f_n t} \qquad 0 \le n \le N-1$$

Essendo poi le sottoportanti ortogonali nell'intervallo T_s, in lato ricezione potrò estrarre senza interferenze i simboli a_n.
Il segnale g(t) costruito con le N sottoportanti andrà poi traslato in frequenza nella banda del canale trasmissivo modulando su una opportuna portante f_0.

In trasmissione si prendono gli N simboli a_n da trasmettere e mediante IFFT si vanno a generare gli N valori complessi $g_0, g_1, \ldots g_{N-1}$ poi si mettono in serie (distanziati di T_S/N) e, mediante conversione DAC, si generano i due segnali (uno reale ed uno immaginario) da inviare al modulatore che li trasla sulla portante f_0 e li trasmette al canale.

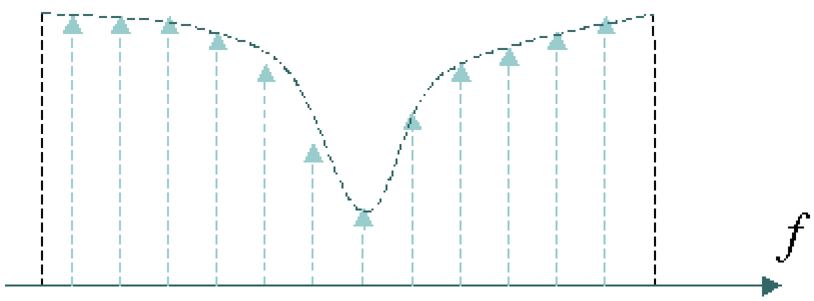


In ricezione, si demodula, dapprima, la portante f_0 estraendo i due segnali (Re e Imm), si campiona a passo T_S/N e riottengo i campioni complessi g_k, li passo nella FFT dove escono distribuiti parallelamente sulle varie frequenze, li moltiplico per l'inverso della funzione di trasferimento del canale (sostanzialmente un'equalizzazione in frequenza) $\frac{1}{H(f_k)}$. Un decisore ricostruirà poi (causa rumore) i simboli di informazione a_n.



:: Equalizzazione

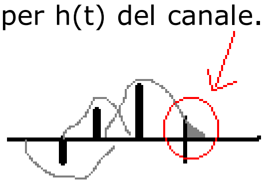
Con l'OFDM l'equalizzazione è abbastanza semplice, ma il problema è dato dalla presenza del rumore.



Essendo il segnale a banda molto stretta (perché le sottoportanti son tutte impacchettate), quando passo su un canale selettivo in frequenza il segnale passa sostanzialmente inalterato, ma il rumore va ad aggiungersi alla sottobanda che cade nel buco e non si va a spalmare su tutte come in altri casi (non OFDM). Perciò si deve trovare la soluzione di trasmettere tanti dati, molto compattati, sulle portanti laterali e, magari, non trasmettere nulla dove ho il buco. Man mano che mi avvicino alle frequenze con più rumore metto sempre meno roba. Ciò, però, è possibile solo su canali con funzioni di trasferimento statiche, perché altrimenti ogni volta cambia tutto. È il caso del doppino telefonico (ADSL) che è dove l'OFDM rende di più. Non è assolutamente possibile per canali che cambiano rapidamente, quali quelli radiomobili dove tra l'altro si aggiunge anche l'effetto doppler; è abbastanza più complicato per canali tipo DVB-T dove tra trasmettitore e singola antenna ci sono canali che variano per ogni utente.

:: dettagli convoluzioni

Tutti i campioni che escono dalla IFFT (a blocchi) vanno poi convoluti linearmente (nel tempo) per h(t) del canale. Questo crea, per i campioni agli estremi del blocco, l'uscita dal blocco stesso



del segnale e quindi un disturbo. Poi in ricezione dopo la FFT, quando moltiplico per 1/H(f) effettuo nel tempo una convoluzione ciclica che quindi non può invertire la convoluzione lineare fatta in trasmissione. Un metodo per far si che le due operazioni siano uguali è quella di zero-paddare ampiamente in trasmissione così ho un blocco e una sequenza di zeri e la parte di

disturbo non mi da più fastidio (perché in quel blocco dopo è tutto zero). Il fatto è che questa soluzione porta ad uno spreco di spazio ed a problemi all'antenna che vede segnale - silenzio e si spegne – segnale e riparte…
Una soluzione migliore è quella di forzare in qualche modo il canale affinché al ricevitore arrivi qualcosa di simile ad una convoluzione ciclica. Praticamente prendo i primi campioni di ogni blocco della IFFT e li ricopio in coda (oppure gli ultimi campioni del blocco e li rimetto in testa) e poi in ricezione toglierò la coda o il preambolo aggiunto. In questo modo il disturbo che ho tolto alla fine me lo riporto in testa e io vedo una convoluzione ciclica perché quanto prima usciva dal blocco ed entrava nel successivo, adesso rientra nello stesso blocco all'inizio. Tutto questo ha un costo in consumo di banda e potenza a causa della presenza del codice ciclico che poi viene buttato in ricezione.

Un'altra cosa sull'IFFT è che, generalmente, vengono spente le bande più esterne (cosi la banda è più chiara ed ha meno interferenze) e su alcune frequenze si trasmettono dei segnali pilota, noti anche al ricevitore sia in ampiezza che in fase, che mi permettono di sapere con precisione la funzione di trasferimento del canale (ad es. se il pilota è una delta, ricevo delta*H(f) ed ho la H(f)).

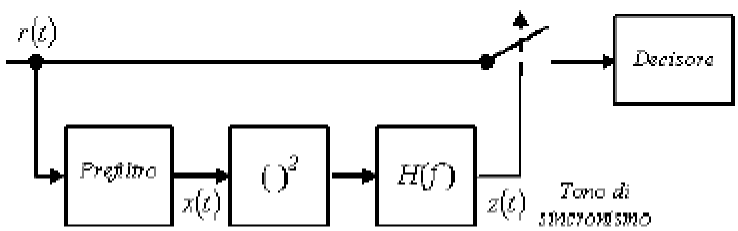
» Il sincronismo

Per poter eseguire una corretta rivelazione dei simboli, un ricevitore deve essere in grado di conoscere dei parametri di sincronizzazione.
Quando si trasmette con banda limitata è possibile non avere ISI se i filtri di trasmissione, di canale e di ricezione soddisfano la condizione di Nyquist. Inoltre, è necessario che il segnale ricevuto venga campionato in istanti di tempo opportuni.

Il sincronismo è fondamentale anche quando si va a fare la demodulazione coerente dove si deve disporre di un oscillatore con stessa frequenza e fase della portante in arrivo. Per questi scopi si utilizzano dei sincronizzatori di portante, in grado di compensare eventuali errori tra l'oscillatore e la reale portante.

Il campionatore (negli istanti $kT+t_0$) è pilotato da un'onda quadra che, ad esempio, sui fronti di salita va a campionare il segnale. Noi vogliamo che i fronti di salita coincidano quanto più possibile con gli istanti di campionamento ottimi. Perciò è necessario che tale onda sia ben sincronizzata.
Una soluzione, quella ottima, sarebbe avere l'onda quadra generata direttamente dalla sorgente, ma per ovvie ragioni ciò è impossibile.

:: Sincronizzatore a quadratore (è ad anello aperto)



Dal segnale ricevuto $r(t) = \sum_k a_k g(t - kT - t_0)$ [1] devo ricavarmi T e il ritardo t_0 dovuto al canale.

Ricavare T non è abbastanza complicato, elevo al quadrato il segnale (quindi $r^2(t)$) che diventa un segnale con valor medio non più nullo, ma periodico di periodo T (supponendo g(t) una forma d'onda nota, di Nyquist) dopo di che lo sviluppo in serie di Fourier, filtro attorno ad 1/T nominale in modo da ricavare la sinusoide a frequenza 1/T (frequenza di simbolo), con un rilevatore a soglia la trasformo nell'onda quadra cercata ed, alla fine, ritardo tutto di 90° in modo da avere il fronte di salita laddove la sinusoide ha il massimo.

[1] Segnale ciclostazionario con a_k sequenza di simboli i.i.d. con var 1 e valore medio 0

Da segnalare che il tempo di simbolo T ricavato non potrà mai essere uguale a quello originale, a causa delle limitazioni dei componenti elettronici ed alla presenza della parte casuale dei dati, ma è indispensabile che lo sia almeno il suo valore medio poi ci sarà un certo *jitter* che lo caratterizzerà… l'importante è che ciò sia adeguatamente sotto controllo.
Trascurando il *jitter* dovuto a rumore termico, l'errore dato dai dati casuali si può gestire se si soddisfano le seguenti due condizioni:
- il filtro H(f) deve essere un passa banda reale, simmetrico pari intorno alla frequenza 1/T e con banda inferiore ad 1/T (in modo da non prendere le code delle sinusoide a 2/T, 3/T…)
- il prefiltro, prima dell'elevamento al quadrato, deve rendere G(f) reale, con simmetria pari intorno ad 1/2T e con banda inferiore ad 1/2T.

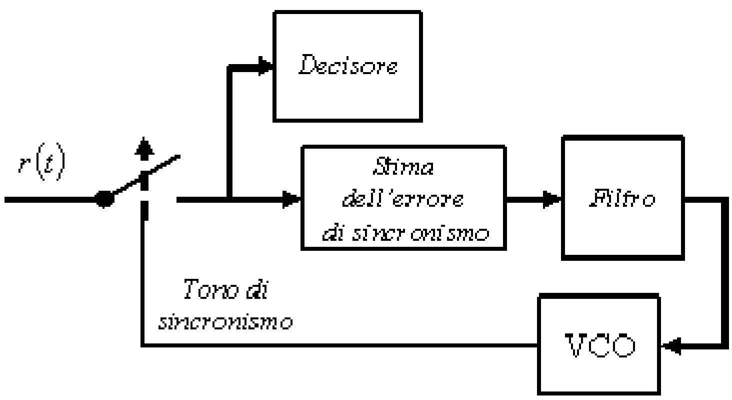
Con queste due condizioni ottengo un annullamento del disturbo proprio negli istanti di campionamento corretti.

:: Sincronizzatore di Oerder e Meyr

Una volta noto T devo trovare il ritardo t_0 con cui si "muove" l'onda quadra.
L'algoritmo OeM prevede che si campioni a frequenza 4/T (almeno) dopo di che si elevano al quadrato i campioni del segnale e si demodula con la sinusoide complessa a frequenza −1/T (cioè moltiplico per $e^{-j\pi\frac{k}{2}}$) e filtro tutto con un passa basso che mi toglie la solita componente aleatoria. Il segnale entra in una ROM contenente diversi valori di arctg ed in base all'ingresso viene restituito il valore stimato di t_0 rispetto alla fase del segnale entrante nella ROM.
Ottenuto il valore stimato di t_0 vado ad effettuare, mediante un attuatore, un'interpolazione per correggere eventuali ritardi. L'attuatore (tempo-discreto) è praticamente un sinc variabile nel tempo con fase lineare (oppure anche un filtro FIR) che non modifica la banda del segnale, ma semplicemente ritardi proporzionalmente al t_0.
Dopo l'interpolazione si decima la sequenza di 4 (poiché avevo campionato a 4 volte la frequenza 1/T).

:: Sincronizzatore di Gardner (retroazionato)

Ha il vantaggio di non avere ritardi, perché nel caso ce ne fossero, la retroazione fa in modo che si possano controllare.
Ricevo il segnale $r(t) = \sum_k a_k g(t - kT)$ dove il ritardo, come prima, è stato assorbito dal campionatore, e vado a campionarlo in t= $kT/2+t_0$.
Per valutare l'errore t_0 con cui vado a campionare potrei utilizzare il teorema del gradiente (con le derivate). Misuro la derivata della potenza di r(t) all'istante t_0 e, se $t_0=0$ (cioè istante ottimo) avrei derivata nulla, mentre se il $t_0>0$ ho un ritardo del campionatore (derivata negativa). Un rivelatore va a calcolare la differenza $y_k(t_0) = r_{2k}(r_{2k+1} - r_{2k-1})$ fra due campioni (la derivata viene approssimata quindi con le differenze finite) e, come per il quadratore, si va a calcolare il valore medio che, essendo r(t) ciclostazionario, sarà ancora periodico di periodo T e ne posso fare lo sviluppo in serie di Fourier.
Considerando poi G(f) a banda limitata tra 1/T e −1/T ottengo le sole componenti a k= ±1. Ho delle sinusoidi non in t (variabili), ma in t_0 (delle costanti), filtro tutto con un passa basso in modo da ridurre la parte aleatoria che si è aggiunta ed ottengo un segnale che pilota un VCO (oscillatore controllato in tensione) il quale a sua volta pilota una sinusoide (convertita in onda quadra con un decisore a soglia) che controlla il campionatore.



» Codifica di canale

Ogni canale è caratterizzato dalla propria capacità (che è proprietà sola del canale e non del sistema) data dalla relazione $C = B\log_2(1 + SNR)$. Questa formula non include equalizzatore, roll-off…, ma dice solamente quanti simboli accetta al secondo e qual è il SNR.
Per un semplice canale AWGN (entra il segnale e s'attacca il rumore) mi basta sapere la potenza del segnale, quella del rumore e la banda e sono in grado di qualificare il segnale.
C è in bit/sec* hertz cioè indica quanti bit al secondo riesco a mettere in 1 Hz (comunque C è a-dimensionale).
La cosa importante del teorema è che se mi baso su C riesco ad avere ZERO errori. Perciò fino a C bit/sec è possibile su quel canale disporre tx e rx in modo da non avere errori.
La codifica è utile per aumentare la protezione del segnale trasmesso contro i disturbi presenti sul canale di trasmissione. Affinché ciò accada, si trasforma la sequenza dei simboli emessi dalla sorgente in una nuova sequenza codificata in modo da essere meno vulnerabile rispetto al rumore del canale.
Ma com'è possibile avere zero errori come dice Shannon? Dobbiamo pensare di essere in uno spazio n-dimensionale (magari anche con n che tende all'infinito) dopo di che il numero delle n-dimensioni è legato alla banda mentre la rappresentazione dei punti nello spazio è la codifica.
Considerando la distribuzione dei punti nello spazio, il reticolo esagonale è più efficace di quello rettangolare perché riesce a contenere più sfere nello stesso spazio oppure ne riesco a mettere una quantità uguale in un minore spazio. Le sfere indicano il volume entro cui ricade ogni punto, sono inizialmente molto piccole, poi, a mano a mano che si somma e peggiora il rumore, le sfere si allargano. Fintanto che le sfere rimangono contenute all'interno del reticolo (come detto quello esagonale è il migliore) riesco a non interferire con i reticoli vicini e ad avere l'errore nullo. È compito del codificatore trovare il codice ottimo per definire i centri delle sfere, mentre il decodificatore dovrà capire in quale reticolo mettere il punto ricevuto.

Per effettuare la codifica ci sono vari metodi e varie teorie. Ci sono codici lunghi come i turbo codici e quelli di Gallager oppure ci sono codici corti (in genere sui 100/200 bit) quali i codici convoluzionali e quelli a blocco (tipo il Reed Solomon).

:: Codice a blocco binario (sono senza memoria)

Si scrive c(n,k,d) con k bit ingresso, n bit uscita, d minima distanza di Hamming. A causa della ridondanza sarà $n \geq k$ mentre il rate del codice è $k/n \leq 1$
Nell' encoder entreranno k bit e ne usciranno n (ad esempio entra 1 esce 111) in quanto ho messo bit di ridondanza (nell'esempio bit di ripetizione). Oppure anche il codice di parità c(n,n-1,2) dove agli n-1 bit ne attacchi uno per rendere pari il numero di 1 nella sequenza (es: entra 000 esce 0000, entra 010 esce 0101, entra 011 esce 0110).

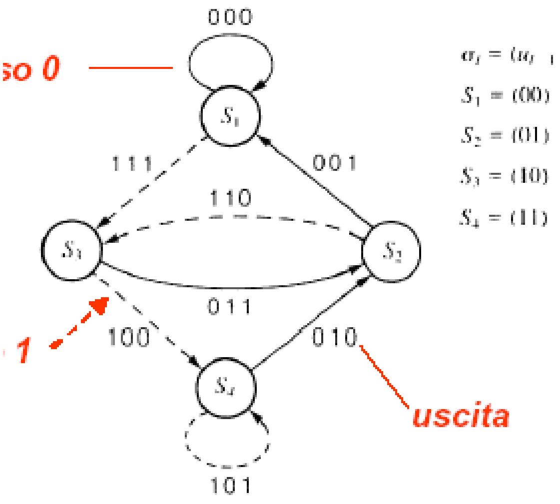
Se da un lato la codifica è utile a proteggere le sequenze trasmesse dal rumore del canale, dall'altro lato codificare comporta aggiungere dei simboli e perciò aumentare il rate di trasmissione rendendo necessario disporre di una banda più larga oppure, nel caso ciò non fosse possibile, ridurre il rate di trasmissione dell'informazione.

:: Decodifica

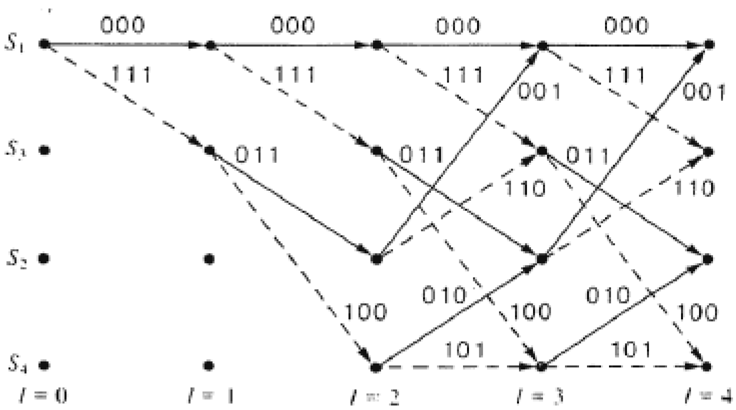
La sorgente emette il vettore di informazione a k bit, l'encoder lo trasforma in vettore di codice a n bit, si modula, si trasmette, si aggiunge AWGN e si riceve un vettore.
Il vettore ricevuto viene decodificato per riottenere il vettore di codice (questa è l'operazione più complicata) poi con un encoder^{-1} si va a ricostruire il vettore di informazione.

Se effettuo una demodulazione *hard* il demodulatore, sulla base del segnale ricevuto, esegue le decisioni sui bit e manda in uscita la sequenza dei bit decisa. Mentre con una *soft* il demodulatore elabora il segnale ricevuto e per ogni bit calcola uno o più parametri per capire se decidere per 0 o per 1.

:: Codici convoluzionali



A differenza dei codici a blocco, sono codici con memoria, perché una parola di codice non dipende solamente dai suoi bit in ingresso, ma anche da quelli relativi agli istanti precedenti.
Per tenere conto di quanto avvenuto in precedenza si usa un'architettura a stati, il cui funzionamento ed evoluzione può essere descritto da macchine a stati finiti o diagrammi a traliccio.
Nella descrizione tramite il diagramma degli stati finiti, ogni configurazione dei dati in memoria viene raffigurata con un cerchio, mentre le transizioni sono degli archi; ogni arco viene "etichettato" con la parola di codice generata dalla relativa transizione. Una tipica macchina a stati finiti ha per ogni bit in ingresso tre in uscita. Ad esempio se entrasse la sequenza {11011} uscirebbe {111 100 010 110 100}. Il rate diventa, ovviamente, 1/3.

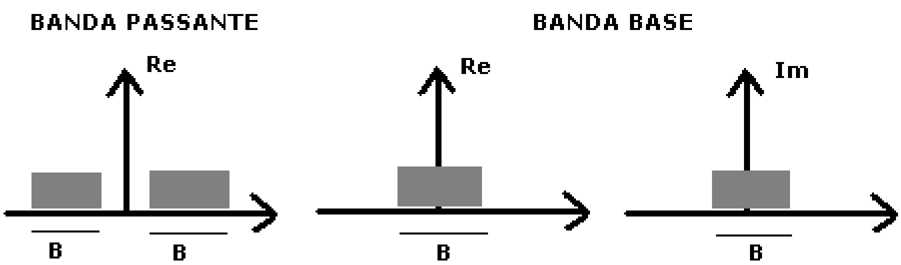


Se, invece, considero diagrammi a traliccio, riesco ad introdurre anche la variabile temporale. Infatti ad ogni istante ho una sezione che mi rappresenta tutti i possibili stati e le possibili transizioni. Con questa tecnica si segue un particolare ramo del traliccio in modo tale che in fase di decodifica sia possibile ripercorrerlo esattamente nel verso opposto. L'algoritmo che permette di decodificare correttamente la parola di codice è l'algoritmo di Viterbi.

In ricezione, l'algoritmo di Viterbi, si basa su una decodifica a massima verosimiglianza (= a minima distanza): tra tutte le possibili sequenze trasmesse, si sceglie quella che presenta la minima distanza rispetto alla sequenza ricevuta. Si parte da uno stato noto, si guarda la prima sequenza di bit ricevuti e si confronta con tutte quelle possibili e si sceglie quella con distanza minima di Hamming, quindi si passa alla sequenza di bit ricevuti successiva e si ripete l'operazione sommandosi al percorso precedente. Cosi avanti finché non si è percorso il tragitto più breve, tenendo in memoria ogni volta solamente il percorso sopravvissuto e scartando gli altri.

» Appunti di interesse

- f continua (f=0) non è misurabile perché richiederebbe un tempo infinito in quanto è una costante da -∞ a +∞
- un segnale è ciclostazionario quando ha media, varianza, autocorrelazione,… periodici nel tempo)
- Segnale G(f) a banda limitata: vuol dire che è nullo al di fuori di 1/T e –1/T.



» Il recupero di portante

Il sincronismo è fondamentale quando si va a fare la demodulazione coerente dove si deve disporre di un oscillatore con stessa frequenza e fase della portante in arrivo (ovviamente la portante nominale è nota, ma come sempre ricevo qualcosa di diverso). Per questi scopi si utilizzano dei sincronizzatori di portante, in grado di compensare eventuali errori tra l'oscillatore e la reale portante.
Al ricevitore arriva il simbolo complesso A_k (complesso perché se è una QAM o PSK siamo nel piano di Gauss) che è moltiplicato per la portante, essendo stato modulato in trasmissione, e viene moltiplicato per quello che esce da un VCO (S_k), ottenendo Z_k. Se la frequenza è giusta, abbiamo demodulato correttamente il nostro simbolo, se invece c'è un errore di fase, questo errore rimane nel termine $e^{j\Delta\varphi}$ che c'è in Z_k. Il nostro obiettivo è adesso quello di misurare questo $\Delta\varphi$: infatti lo scopo è trovare la differenza tra la frequenza a cui arriva il segnale e quella con cui si demodula (che è data appunto dal VCO): con questa differenza andremo poi a pilotare l'ingresso del VCO.
Per effettuare questa misura sappiamo che A_k è un simbolo complesso e, quindi, porta con se la sua fase che ci darebbe fastidio nel misurare $\Delta\varphi$ perché ne influenzerebbe la misura. Per eliminare la fase di A_k mettiamo un decisore a soglia che prenda Z_k e ricostruisca A_k, ne facciamo il complesso coniugato e moltiplichiamo lo stesso Z_k per A_k^*. Il risultato che otteniamo è praticamente $A_k e^{j\Delta\varphi}$ per A_k^* cioè $Y_k = |A_k|^2 e^{j\Delta\varphi}$. In questo modo possiamo isolare $e^{j\Delta\varphi}$, calcolarne l'argomento, filtrare il risultato da eventuali fluttuazioni dovute, ad esempio, al rumore, moltiplicare il tutto per −1 e pilotare il VCO.
Ma come funziona il VCO? È un oscillatore che ha una sua frequenza naturale (che è tarata su quella nominale), ma con un ingresso positivo aumenta la frequenza e con un ingresso negativo diminuisce la frequenza. Per la precisione, per far sì che la frequenza aumenti deve arrivare in ingresso un segnale positivo costante, come un rettangolo. Un impulso, invece, non bastarebbe: gli farebbe fare solamente un saltino di fase.

Da notare che tutto il sistema soffre dell'ambiguità di fase di $k\frac{\pi}{2}$, in quanto la portante viene ricostruita a meno del segno. Questo rischierebbe di farmi ricostruire la costellazione ruotata rispetto all'originale. Uno dei modi per risolvere il problema consiste nell'utilizzo di una modulazione differenziale, dove l'informazione non è contenuta nella fase assoluta, ma nella variazione di fase tra un impulso e il successivo. Questo, però, ha lo svantaggio che un errore in decisione causa un doppio errore, in quanto l'errore viene memorizzato anche per il colpo successivo, essendo i due impulsi strettamente legati.