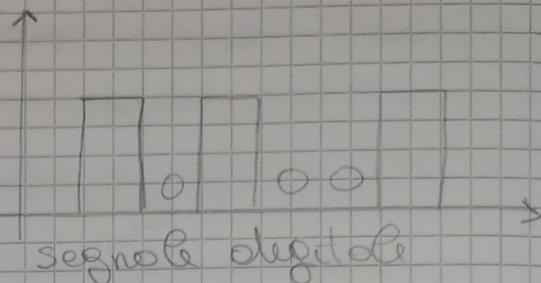
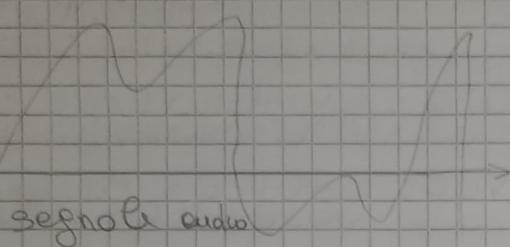


15/12/2020

Segnale audio



Il segnale audio è un segnale analogico che deve essere campionato e quantizzato:

Qualità cd \Rightarrow 44100 campioni secondo quantizzato con 16 bit. Per 2 canali ho $44100 * 16 * 2 = 176400 \text{ bit/s}$

COMPRESSEIONE 10:1 \Rightarrow Si compume un file per uolurre l'occupazion di 10 volte rispetto allo spazio occupato in qualità cd.

3 tipi di codifiche:

- Codifiche nel dominio del tempo (Loss-Less) \rightarrow codifica entropiche (basse compressioni)
- Codifiche per modelli (Lossy) \rightarrow trasmetto sonomethu
- Codifiche nel dominio delle frequenze (Lossy)

Codifiche temporali

Pulse code modulation: codifica e modulazione di impulso \Rightarrow onda psm. Compionamento con numero di bit fissi (8 kHz - 8 bit).

DPCM: differential pulse code modulation \Rightarrow non muo il numero di bit se presentativo del campione ma muo la differenza col campione psm

ADPCM: Adaptive differential pulse code modulation \Rightarrow muo le differenze ma assegno n. bit variabile

INVIO CAMPIONI

Codifiche per modelli

Il trasmettitore invia sul canale obbligatoriamente correlazistiche delle voci sulla base delle quali il ricevitore può ricostruire fedelmente possibile la voce.
=> motivo delle voci mettlice in assenza di campo
-> invio parametri vocali sbagliati a causa del troppo rumore.

Da una generazione di telefoni mobili (UMTS-GSM) cambiano i numeri di parametri inviati.

LPC → linear predictive coding => poiché invio sul canale le correlazistiche delle voci e non i campioni è come se io inviasse le informazioni che mi permettono di prevedere i campioni delle voci in ricezione
→ basso bit rate (ambito militare)

CELP → rispetto a lpc sono migliori i modi in cui il segnale viene riprodotto in ricezione (UMTS)

LPC

In fase di trasmissione si analizza il segnale vocale e si estraggono le Correlazistiche percepitive e queste poi sono inviate sul canale

In fase di ricezione un sintetizzatore preleva le correlazistiche inviate sul segnale e ricostruisce il più fedelmente possibile i campioni vocali

In fase di encoding si effettua su piccoli tratti del segnale audio → ogni 20-25^{ms} secondi poiché in questo lasso di tempo la voce (c. percepitive) è costante

Grotteistiche percezive

- 1) Timbro (Pitch): frequenza del segnale. compreso fra i 2 e 5 kHz. (massimo uscita da orecchio)
- 2) Periodo: Durata del segnale
- 3) Volume
- 4) Parametri eccezione tratto vocale:
 - ipotesi che è rendere efficiente la sintesi
 - Suoni voiced (sonori): vibrazione ^{armonica} corde vocali
 - Suoni unvoiced (sordi): corde vocali vibrono in modo impulsivo

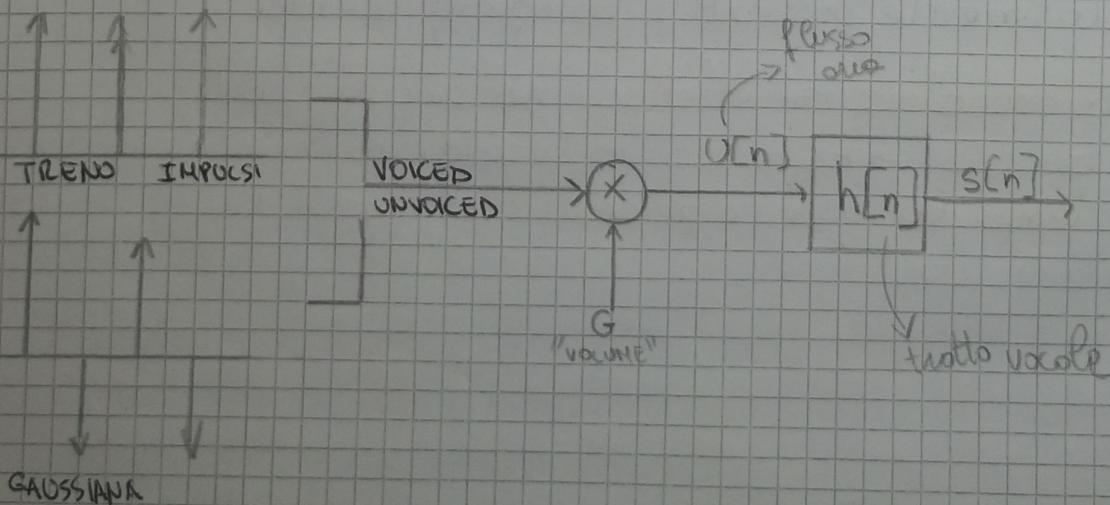
circa 20-25 ms sono calcolati

In LPC-10 si aggiungono altri 10 parametri che descrivono in eccezione il tratto vocale che ha picchiato quella voce. => definiscono le uscite impulsive del filtro che deve emulare il tratto vocale.

Per un suono sonoro sono inviati impulsi periodici ossia un treno di impulsi separati da un periodo costante.

Per un suono sordo viene inviata una gaussiana

Decoder LPC



17/12/2020

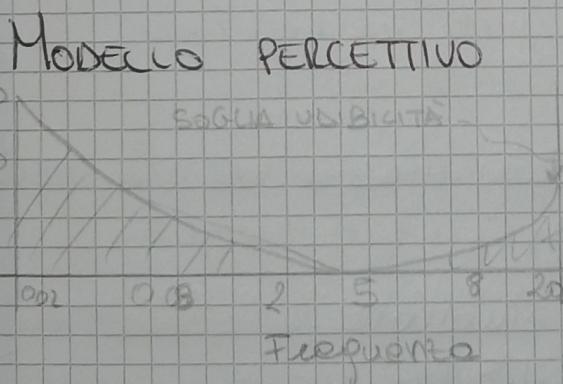
MELP: mixed excited linear prediction \Rightarrow ottenuto da una combinazione lineare di ingressi, sonori e sonori

CELP: code excited linear prediction \Rightarrow gli ingressi sono scelti in una matrice con 1024 righe e 16 colonne in cui ogni riga rappresenta un possibile ingresso \Rightarrow + complessità

L'orecchio è un filtro passa-banda (2-5 kHz)

usare i modelli sarebbe troppo complesso e non avrebbe

Poiché usare uno codice per modelli in ambito musicale risulterebbe computazionalmente oneroso si sfrutta il fatto che molte frequenze non sono percepibili all'orecchio e quindi queste frequenze saranno tagliate



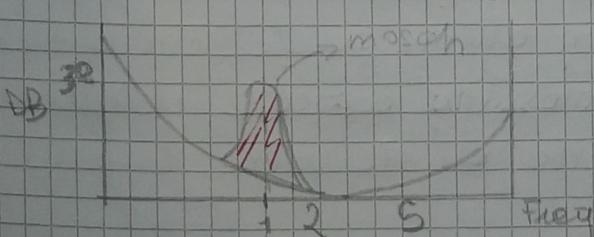
ANDAMENTO
NON LINEARE

non
comprendibili

Cioè che uccide sopra le soglie è udibile, ciò che è sotto non si sente \Rightarrow li taglio

tagliano solo sotto le soglie non basta \Rightarrow poca
complezione \rightarrow audio 2

Moschettamento orecchio in frequenza

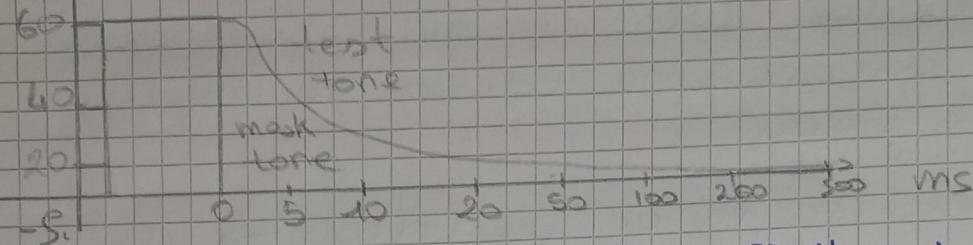


\rightarrow devo verificare cosa fa la curva di moschettamento per ogni frequenza

\rightarrow elimino i toni più bassi, che eccitano nel moschettamento

l'uomo a Rx solo tono forte e toni che superano le moschettazioni se no un suono forte centrato alla freq. di 1 kHz si nota che la curva prodotta dal suono stesso è tale da moschettare tutti i suoni col loro vicini che non superano l'intensità assunta dal suono.

Moschettamento temporale



Il tono forte produce un effetto di moschettamento su tutti i suoni che vengono piccolati nello immediatamente vicinante (tempo) con intensità minore.

Il tempo vibra alle frequenze del suono forte, per ascoltare suoni a frequenze minori deve vibrare a minor frequenza. \Rightarrow tempo di moschettamento

MPEG-1

- singolo canale
- 2 canali 32-48 kHz
- stereo semplice (insieme) 16-320 kbit/sec
- stereo composto (differente sx - dx)

Sono stati standardizzati MPEG-1 di tre tipi diversi noti come Layer:

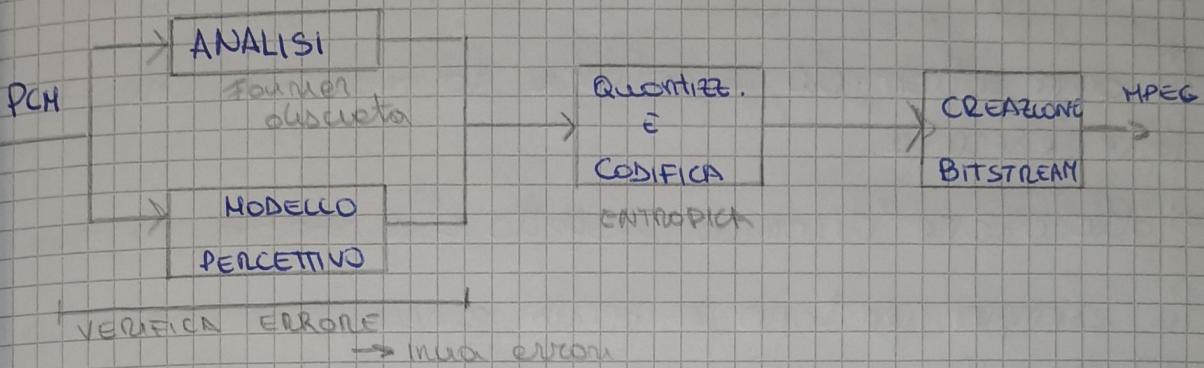
Layer 1: il + semplice studiato per avere prestazioni sope 128 kbit/sec per canale con rapporti compressione 1:4

Layer 2: evoluzione del 1° lavoro con bit rate massimi a 128 kbit/s con rapporti compressione 1:6 / 1:8

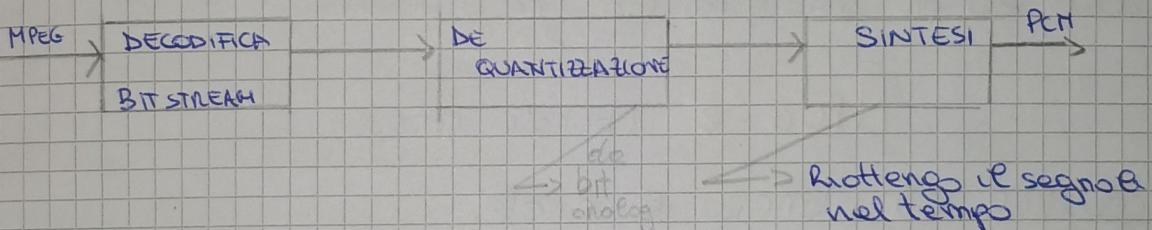
Layer 3 (MP3): più complesso con ottime prestazioni e 64 kbit per canale. Con massime capacità riduttive delle dimensioni di 12 volte. (1:12)

Si usa una trasformata di Fourier modificata (molct) \Rightarrow ossia una trasformata discreta coseno con Kernel del coseno ottimizzato per l'audio

CODIFICATORE MPEG



DECODIFICATORE MPEG



\rightarrow Analisi: trasformo il segnale dal dominio temporale al dominio delle frequenze

\rightarrow Modello perettivo: indica le componenti spettrali da eliminare

\rightarrow Quantizzazione e codifica: scelego n. bit per comporre l'MP3 ha una quantizzazione adattativa per avere meno bit trasmessi

\rightarrow Creazione stream MPEG:

! Se devo codificare un suono di poco superiore al rumore (moschicamento) posso usare un n. bit inferiore rispetto a quello per toni di molto maggiori rispetto alle soglie di rumore. Perché il suono basso è quasi inaudibile dal rumore, l'altro invece necessita migliore cod.

Sì suddivide il suono in più bande (sottobande) per permettere un'analisi più precisa del segnale.

Tra i 2 e 5 kHz ha 28 sottobande, il MPEG 3 ne usa 32 perché usuale meglio usare potenze di 2.

Per ogni sottobanda si calcola la moschiera che il tono forte esercita.

→ allocazione dinamica rispetto alla soglia di moschiera.

→ Per esempio un tono a 60 db genera moschieramento di 12 db nella banda precedente e 15 db nelle successive.
punto ne ha 10 che è 4/12 non trasmettendo il suono

Se no 35 db nelle bande successive ne possono in realtà solo 20 ph sono quei db che superano il moschieramento

↳ così uso meno bits

MPEG - 2 (bande large)

+ compressione
- qualità
- 118 Kbps minimo

→ codifica di almeno 5 canali

Si provò pure ad implementare un codificatore MPEG-2 retro-compatibile per poter anche sfruttare il vecchio standard MPEG-1. → peggioramento prestazioni

NON PIÙ STANDARDIZZATO

MPEG-2 AAC (advanced audio coding)

Si possono codificare fino a 48 canali contemporaneamente con una frequenza ^{di campionamento} che va da 8 a 96 kHz per canale.

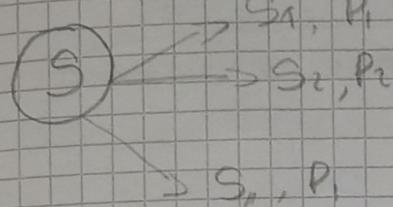
→ NO RETROCOMPATIBILE ⇒ Hardware diverso

MPEG-4

Più programmazione orientata agli oggetti ⇒ insieme armonico di tanti fili audio diversi con codifiche indipendenti a canali diversi.

07/01/2021

Codifiche di Canale



$$N = \# \text{ symb}$$

$$I_i = \log_2 \frac{1}{P_i} = -\log_2 P_i \text{ [bit]}$$

minimo numero medio di bit con cui una sorgente emette i simboli: entropia H(x)

$$H(x) = -\sum_{i=1}^n P_i \log_2 P_i \text{ [bit/symb]}$$

La codifica di sorgente ha il compito di interfacciare la sorgente (trasmettitore) con il canale di trasmissione.

→ elabora l'informazione in modo tale che il ricevitore possa capire se quella trasmissione sia affetta da errori o meno

Questo tipo di codifica viene implementata sia in trasmissione sia in ricezione => aumenta la complessità computazionale

La codifica di canale in senso delle strunghe di bit all'interno del pacchetto da trasmettere mediante un apposito algoritmo (noto sia a Rx sia a Tx come è anche nota la strungha usata sia a Rx che a Tx) ossia lo **Ridondanza** che non è vero e ripete contenuto informativo ma un artificio aggiunto al pacchetto solo per permettere a Rx di capire se c'è stato errore.

Più la ridondanza è lunga e strutturata + è facile per Rx recuperare la trasmissione corretta da quella corrotta

Poiché vado ad inserire bit in trasmissione
ondolando e richiedere + parole sto realizzando
una codifica opposta a quelle di sorgente

+ è lunga la parola d'origine meno è efficiente
la comunicazione

Esempio di Ridondanza

B come Boni

La ridondanza ci permette di separare
le parole di codice."

Il ricevitore associa una parola di codice ad
un simbolo che sia più vicino possibile
=> Regola di Assoziazione a max. Vocabolario

Bit di Parità

Sorgente che trasmette 4 symb

A → 00

B → 01

C → 11

D → 10

} Sente codice
di canali

+ efficiente codifica di canali in termini di lunghezza
di ridondante

Se il canale introduce un errore su 1 bit,
quale trasmettessi il codice sopra senza codifica
di canali, se 1 dei bit ondolasse in errore avrei
comunque una parola di codice valida => non mi
accorgo di errare => Rx decodifica con le parole
+ simili

Le parole del codice distano tutte 1 bit, infatti combinando anche i solo bit trovo comunque una codifica valida

Distanza di Hamming (d_h): numero di bit da cambiare per ottenere una nuova parola di codice valido.

Un codice con distanza di Hamming pari a 1 non consente il Rx di rilevare errori

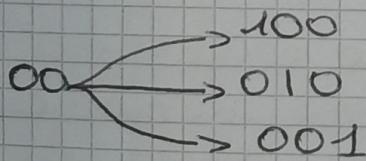
	bit parità
A \rightarrow	00
B \rightarrow	01
C \rightarrow	11
D \rightarrow	10

$$D_H = 2$$

bit di parità: bit tale per cui la somma di tutti i bit di una parola sia sempre un numero pari

\rightarrow le codifiche che non sono + lunghe 2 ma sono lunghe 3, ho introdotto codifiche di corris.

Se infatti trasmetto 00 e il corris. introduce un errore di 1 bit



SYMB ACCETTATI
000
011
110
101

\Rightarrow nessuna parola ottenuta è una parola valida. Inoltre verifica subito la presenza di un errore in quanto me lo segnala le parate. Ex. 100 mi aspetto 1 come bit parità così de farlo diventare per la somma di tutti bit

Non riesco col orribile olla codifica giuste poiché le parole valide sono tutte egualmente distanti da quelli ricevute

Il codice con bit di parità ossia con DH pari a 2 può fare detection di 1 solo errore ma non error correction

Un codice può fare detection di un numero di errori pari a $(DH-1)$

Un codice può fare error correction di un numero di bit pari a $\lceil \frac{DH-1}{2} \rceil$

Il bit di parità identifica un numero disponibile di errori ma senza sapere in che posizione

La gestione degli errori è diversa per i canali internet (TCP/IP)

Si presta attenzione alla integrità dei dati. Ossia se un pacchetto è corrotto viene subito richiesta la trasmissione del pacchetto (ARQ). La connessione è a lange banda e ha un canale di ritorno tra Rx e Tx.

È praticabile perché mi aspetto che ci siano pochi errori perciò è più efficiente richiedere la rtrasmissione del pacchetto piuttosto che aumentare la complessità comput. per risolvere l'errore

FEC: Correzione errore tramite aumento di complessità di codifica così da trovare la codifica giusta "analizzando" quelle sbagliate

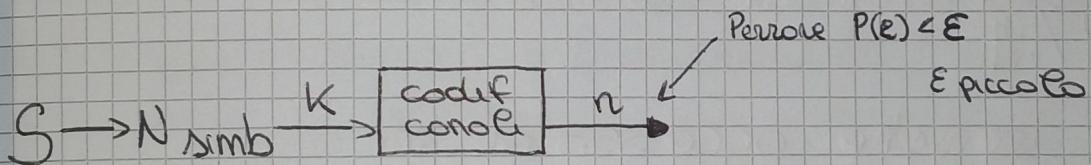
Usata in comunicazione Radio, con canali a banda bassa

Molti errori che non compromettono l'integrità del pacchetto

(Real time) su operazione che richiedono strumenti vincoli real time queste non arrivasse un pacchetto ^{gusto} o questo arrivasse con errori si sceglie di scartarlo.

Non uso Ne feci ne ARA così mantiene la fluidità delle trasmissioni piuttosto che perdere fluidità per correggere gli errori

(ARA ibrido) cerca di ricevere l'informazione corretta da quella corrotta e solo quando ci fossero troppi errori o impedire la correzione richiede la ritrasmissione del pacchetto



R_c - Rate codificatore corrente (quante cifre escono rispetto a quante ne entrano) $R_c = \frac{K}{m}$ $R = R_c \cdot H$

$$P(e) < e^{-nE(R)}$$

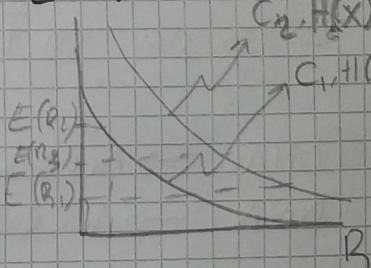
limite superiore di probabilità d'errore

$$R = R_c \cdot H(x)$$

C - cap. canale

$$C_2 > C_1$$

$$E(n) =$$



funzione convessa e decrescente positiva di R

Per diminuire $P(e) \Rightarrow e^{-nE(R)}$ oltre tendore è 0

1) \Rightarrow aumento n di minimo $\Rightarrow P(e)$ più basso sull'espon. e diminuisco R (da R_2 a $R_1 \Rightarrow$ aumentato n più R_c)

$$P(e) \rightarrow 0$$

$B \gg$

aumento redundanza + banda

2) Se $E(R_3) > E(R_1)$

esponente aumenta \rightarrow C maggiore

$P(e) > 0$, se $E(R) >$

$(C_1 \rightarrow C_2 \quad C_2 > C_1)$
aumento SNR

$$C = B(1 + \log \frac{S}{N})$$

3) $n \gg$

$$R_C = \frac{k}{m} - \text{cost.}$$

\Rightarrow non stocassendo + bonde

non migliora SNR (minimizza errore)

$P(e) \rightarrow 0$

se aumento complessità codif canali

se si progetta uno codificatore
canali così possano trasmettere con
certo piccolo

minimo errore cod canali

12/01/21

Tecniche accesso al mezzo

N utenti devono accedere al mezzo

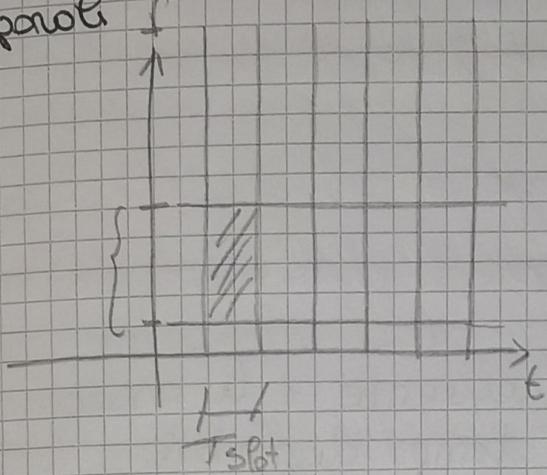
Divisione in Tempo: suddividendo l'accesso al mezzo
tramite intervalli

Divisione in Frequenze

TDMA → divisione in tempo
FDMA → divisione in frequenza

TDMA

Assegno le facoltà di trasmissione su un canale suddividendo l'accesso in intervalli temporali f



d'attente uso il canale nel solo Time slot assegnatogli con la possibilità di saturare B

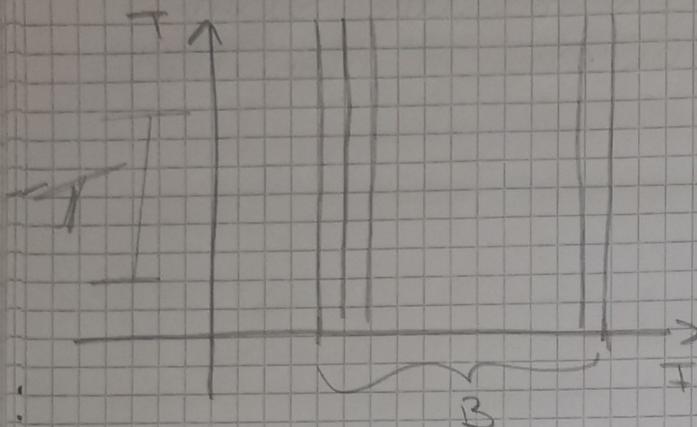
Il numero di utenti è fisso in un intervallo di tempo, di conseguenza suddivisione l'intervalle in n. utenti → assegnando un T_s a ciascuno

Tutti questi T_s formano un frame che viene inviato, queste suddivisione / trasmissione si ripete periodicamente con stessi T_s in lunghezza e tempo.

Inefficiente perché:

- 1) N utenti limitato dal numero di T_s per ogni pacchetto.
- 2) Il sistema deve essere preciso per far sì che T_{s1} interfaccia con T_{s2} , perciò montano obelli piccoli cioè a volte T_s vuote per evitare interferenze.

FDMA



Suddivisione B in sottofrequenze e le assegno ad un utente per tutto il tempo disponibile

=> banda stretta che si riduce all'aumentare degli utenti

ho bisogno di n Rx sintonizzati su frequenze diverse.

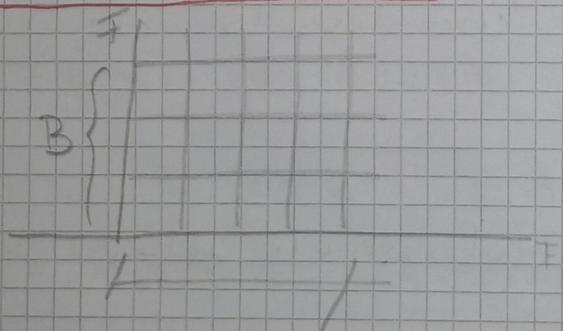
GSM (2G)

=>

FDMA/TDMA

C = 3.6 Kb/s

↳ no streaming



Assegno ad un utente una sottobanda b per un Timeslot Ts

=> Maggiore n. utenti

Svantaggi:

1) non posso avere servizi a molti bit-rate

GPRS (2.5 G)

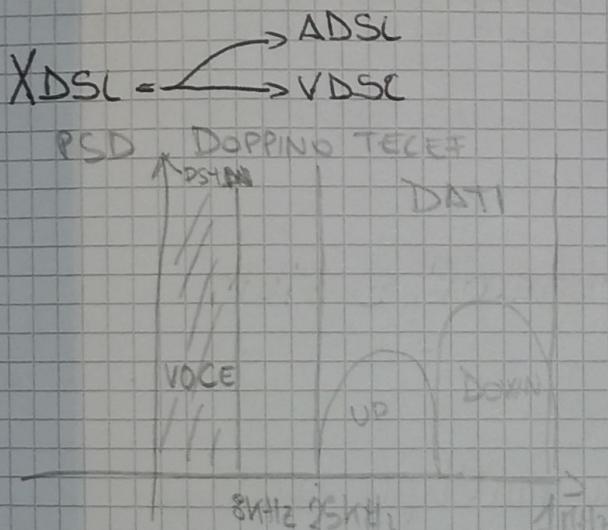
8 connessioni GSM unite insieme in parallelo
migliore accesso a rete

=> non crea una rete nuova, ma concede 8 connessioni insieme allo stesso utente

UMTS | 3G

Nuova infrastruttura

ADSL / Asymmetric digital subscribe line



Il doppio telefonico permette di sfruttare uno banda di 1 MHz di cui solo 8 kHz erano usati per voce. Senza restituire la rete fissa studiarono un modo per occupare al mezzo tranne lo stesso doppio telefonico suolo violando in banda la copertina. FDMA

S scelse di usare solo la voce fino a 25 kHz e dopo i 25 kHz si lasciano per i servizi dati. Suolo violando da 25 a 1 MHz le zone di downlink (1^ parte) e uplink

Con uplink << Downlink

quindi downlink >> veloce

poiché chi metteva a disposizione la rete forniva anche i servizi

Oggi non è così \Rightarrow stream, social netw.

Ogni bondo è suddiviso in sottocanali poiché quelli che si trovano ai bordi delle bende sono + suscettibili a interferenze

Al contrario di quelli centrali ~~che~~ che hanno meno interferenze e usano modulazione con livelli in quantità minore (QAM 16 \rightarrow 256 lev)
 \hookrightarrow scelto in base a SNR

UMTS

3 G

universal mobile terrestrial system

W-CDMA = UMTS (Europa - Giappone)
1.9 GHz

Wide band - code division multiple access
(banda larga)

codice

In America 1.9 GHz l'UMTS sfrutta il N-CDMA
dove N = narrow-band (banda stretta)

Antenne orientazione in UMTS Europeo 40-70 m sec

punto di entrata in rete il tel si eleva
sincronizzazione alla rete

F

Asf Asn Tsu

eleva copre in che punto del fiume
sono entrato in rete in modo tale
che l'opposto sia sincronizzato sul fiume
2 al punto Ts

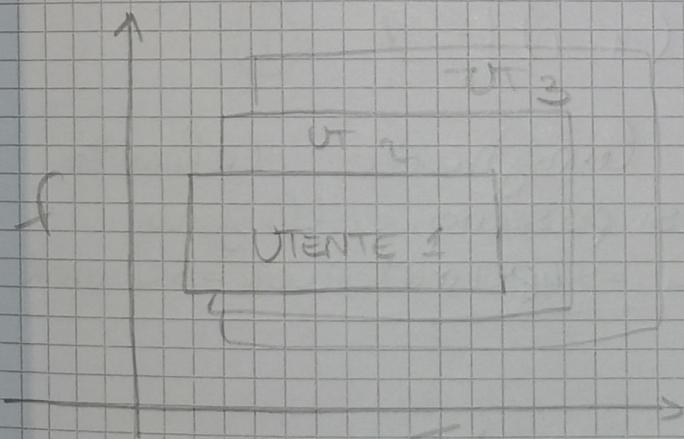
In America non ho bisogno di sincronizzazione
pk i delTs posso escludere quando la trasmissione
è iniziata poiché le antenne trasmettono tutte
allo stesso tempo.

Solo tramite GPS ero possibile la sincronizzazione
di tutte le antenne. GPS che era col USO
esclusivo americano che disturbava le frequenze
per impedire agli altri di uscirlo (solo dopo il
2000 ho GPS in America)

=> No GPS, No 3G

INFRASTRUTTURA PRIVOCUTURARIA

Poiché non sto suddividendo le bande
in tempo o in frequenze => le bande nella
sua totalità sono assegnate a 1 utente e agli
altri



- ed ogni utente si assegna un codice, con cui viene crittografata la trasmissione del singolo utente
- Codice diverso per ogni utente

- Il codice è uno sequenza di +1 e -1 creata dalla rete e assegnate all'utente

→ Codici sono ortogonali tra loro

→ prodotto scalare tra codice = 0

cross-correlazione

$$\langle c_1, c_2 \rangle = 0$$

$$\langle c_1, c_1 \rangle = 1$$

$$\begin{cases} c_1 \cdot v_1 \\ c_2 \cdot v_2 \\ c_3 \cdot v_3 \end{cases} \quad \left\{ \begin{array}{l} c_1 \cdot v_1 + c_2 \cdot v_2 + c_3 \cdot v_3 \end{array} \right.$$

Se voglio decodif. il segnale c_1 , la rete
comunica a Rx il codice per il segnale

di utente 1. \Rightarrow Rx è autorizzato e decod.
Solo segnale 1

Dato che il segnale ricevuto è la somma di tutti i codici moltiplicati per il rispettivo utente, ossia

$$R = C_1 \cdot U_1 + C_2 \cdot U_2 + C_3 \cdot U_3$$

Per decodificare U_1 ho

$$R \cdot C_1 = C_1 \cdot C_1 \cdot U_1 + C_1 \cdot C_2 \cdot U_2 + C_1 \cdot C_3 \cdot U_3$$

codici sono potenze di 2

$$C_1 = [1 \ -1 \ 1 \ -1]$$

$$C_2 = [-1 \ -1 \ +1 \ +1]$$

$$\langle C_1, C_2 \rangle = \frac{1}{N} \sum C_{1,i} \cdot C_{2,i} = \frac{1}{4} [(1 \cdot 1) + (-1 \cdot -1) + (1 \cdot -1) + (-1 \cdot 1)] = 0$$

$$\langle C_1, C_1 \rangle = \frac{1}{4} [(1 \cdot 1) + (-1 \cdot -1) + (1 \cdot -1) + (-1 \cdot 1)] = \frac{4}{4} = 1$$

Il codice con le sue lunghezze delimita il n. max di utenti che si possono collegare al canale \Rightarrow codice lungo 4 \Rightarrow 4 permutations \Rightarrow 4 utenti

LUNGHEZZA CODICE = # utenti

Ogni elemento del codice si chiama chip
(ogni codice ne ha 256).

Si trasmettono + velocemente

T_b

—

—

—

—

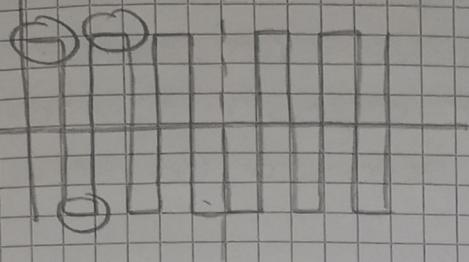
—

il codice C_1 quando moltiplicato per il bit dello trasmissione è pari a se stessa, ossia ha bit-1 se trasmette il codice

la durata di un bit di un codice è minore di un bit.

$$T_b = n T_{chip}$$

G.U.



Spreading il singolo bit viene ripetuto n volte con n pari olio lunghe del codice
DIMENSIONE minore m T , si allunga m frequenza
(f. onde)

R₁ - G₁

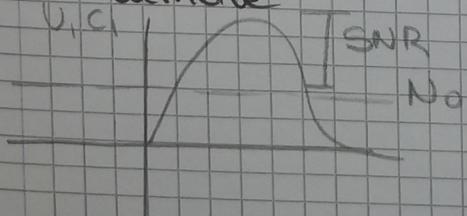
DESPREADING

Banda richiesta è Nb dove b
è la banda dei segnali senza spreading.
5 kHz

5 kHz

Potenza limone uguale

=> Abbassandosi il profilo di potere ollare
il segnale diventa confondibile con il
rumore



In ricezione quando moltiplico il segnale per il codice ottengo un all'segmento delle forme d'onda. Ma poiché ho anche il rumore che si somma al segnale quando lo moltiplico per c questo subisce lo spreading abbassandosi.

$$\text{SNR}^s = N \text{ SNR}$$

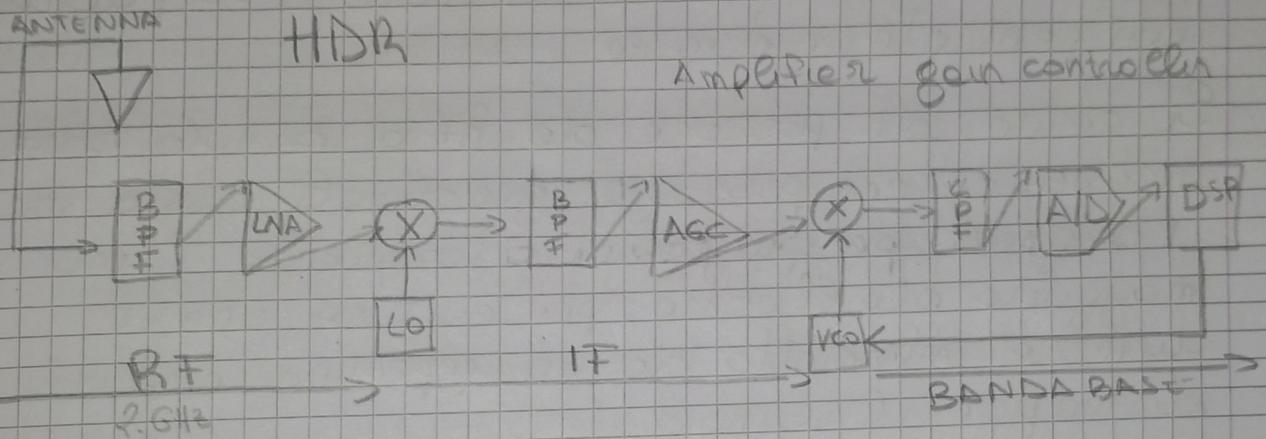
No

SNR in despreading è maggiore di SNR rumore di spreading poiché il rumore si abbassa in potenza di N volte

WIFI - BLUETOOTH - GPS sono CDMA

14/10/1 / 2022

Da Hardware defined Receiver (HdR) a Software defined Receiver (SdR)



- I segnali trasmessi via radio sono pure modulati (moltiplicati per coseno) ossia sposta il segnale creato interamente in banda base ad alte frequenze ossia offe RADIOFREQUENZE
- Noi siamo in grado di comporre un segnale se e solo se questo è limitato in banda (oltre off l'origine).
- La radiofrequente non è sicuramente un segnale con frequenze vicine off l'origine (tavole intorno ai 2 GHz) poiché ha subito modulazione. Noi possiamo comporre in kHz o MHz.
- Dunque necessitiamo di demodulatore per passare oltre banda base in cui possiamo comporre.
- If: frequenza intermedia. Puro ho il demodulatore composto da un blocco che moltiplica le segnali ricevuti per lo stesso coseno usato

in modulazione con l'effetto di replicare
3 volte il segnale 1 in 0 e 2 di doppio
della portante. Poi ho un filtro posse banda
che ha il compito di scartare le repliche
non centrate in frequenza

- Si cerca di possoire da RF a banda base e
viceversa con passaggi geschwedi ossia moltiplicando
il segnale per più coseni a frequenze l'uno
maggiori che l'altro per evitare che il
rumore introdotto da un repentino complemento
distrugga il segnale. (voti anche per amplificatori)

=> + stecca IF

Solo DSP è digitale il resto è analogico
ossia è comandato dal solo hardware.
Quindi per possoire su frequenze diverse deve
essere uscita.

Il filtro posse banda stop l'antenna
serve per togliere i 5 GHz che escono solo
le bande di interesse poiché l'antenna per
esser sicura di ricevere coda ad una banda
maggiore

- LNA: Amplificatore => low noise così da introdurre
poco rumore poiché è un analogico
introduce rumore

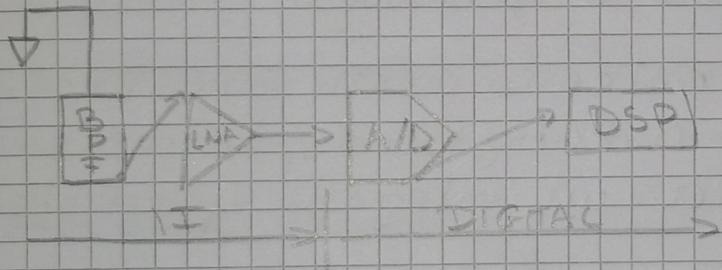
Tutti i blocchi prima di AD sono studiati
opposta per ogni singolo canale

SDR

→ DIVERSO X DIVERSI STANDARD
→ ANCHE DSP CAMBIA

Tecnica accesso al mezzo molto simile
⇒ stesso algoritmo con codice diversi pk
il rumore che offriggi un segnale di un altro
sistema è diverso dal segnale di un altro
sistema (GPS [satelliti] e wifi)

Per avere un sistema SDR deve lavorare
in digitale per cui deve necessariamente
spostare A/D più vicino possibile all'antenna
(3G - WIFI - BLUETOOTH - GPS ⇒ 2-2.5 GHz)
HD FLESSIBILE meno PERFORMANTE



voglio comprendere
a radiofrequenza
che contengono
tutte le info degli
standard su
quelle bande

Antenna: deve avere un'antenna a banda
larga così da ricevere tutte le
comunicazioni

⇒ + aumento la banda dell'antenna
maggiore sarà il rumore acquisito.

Antenne tradizionali multizizzabili
in questo schema

⇒ Smart antenne: antenne dotate di software
uprogrammabile, in modo tale che
migliorano da sole la copertura di

oscolto per una trasmissione

- Hd. digitale performante e riprogrammabile:
=> abb. in contrasto tra canali
↳ + hd. digitali insieme
- Si dimostra (Nyquist e Shannon) che si può comporre in radio frequenze
=> $\approx 5 \text{ GHz}$ per campionamento
↳ 5 giga campioni/sec
↳ uso 10 bit campioni
↳ $5 \text{ giga} \cdot 10^{\frac{\text{bit}}{\text{camp}}} \approx 50 \text{ gigabit/sec}$
=> Il convertitore A/D deve poter gestire un flusso dati di 50 Gb/s

Queste sono strutture ideali

- Risolvo problema antenna con smart-antenna
- Risolvo problema DSP (digital signal processor) inserendo circuiti integrati specifici
- Risolvo problema A/D dimensionando prestazioni
=> Campiona e freq. intermedia => - numero di standard rececibili ma solo quelli che si somigliano in software e freq. (meno larghe bande gestite)

pensando a
l'elosing.

Compiono o Radiofrequente sfruttando il segnale centrale replicato ovvero una replica del singolo compone con LPF in modo tale che resti la sola replica in O. \Rightarrow Equivalenti posso bussare con il vantaggio che un compone in RF contiene in sé tutta l'info sullo banda lunghezza Banda - velocità

In trasmissione ho un POWER AMPLIFIER (HDR) + D'A si sposta ^{verso} l'antenna + è SDR

Sconsigli Spettro

- Bandi giri per usare lo spettro radio \Rightarrow bene pubblico dobbiamo gestire molto stato agli operatori di tlc

Accesso dinamico allo spettro tramite radio cognitive. In modo tale da poter usufruire di uno banda al momento in cui quelle bandi non si usa più