



ARI

ASSOCIAZIONE RADIOAMATORI ITALIANI

Sezione " Brianza " di Lissone

Via Settembrini, 26 - 20035 Lissone (MI)

Tel : 333.23 30 269

Software Defined Radio

Nell'ambito della manifestazione :

**CONOSCIAMO E PROVIAMO
la radio definita dal software**



Alberto di Bene I2PHD

SDR, se la conosci non la eviti, se non la conosci ti spaventa

SDR, se la conosci non la eviti, se non la conosci ti spaventa

Alberto di Bene I2PHD

SDR : una parola che sempre piu' spesso sentiamo pronunciare, leggiamo sulle riviste tecniche, e anche se abbiamo una vaga idea di quello che possa essere, non e' facile trovare informazioni concrete, che vadano al di la' del sentito dire, circa cosa realmente sia e come si inserisca nel contesto della nostra attività radioamatoriale. Questo mio intervento al Convegno di Lissone 2008 e' un tentativo di dissipare, sia pur parzialmente e con molte omissioni, le nebbie e i miti che circondano questa parola e le tecnologie che ad essa fanno riferimento.

Partiamo dal significato stesso della parola : SDR, ovvero un acronimo che in Inglese nasce dalle parole Software Defined Radio, ovvero Radio Definita dal Software. Cosa vuol dire ? Le radio, come le abbiamo conosciute sino ad oggi, sono definite dall'hardware. Se uso un filtro a quarzo da 2.4 kHz di banda passante, quella sara' la selettività della mia radio. Se uso un rivelatore a prodotto, tendenzialmente sara' una radio adatta per la SSB, e cosi' via. Se voglio cambiare la selettività devo usare un filtro a quarzo differente, e lo stesso, se voglio ricevere la AM, il rivelatore dovra' essere di un altro tipo. Pensiamo a come si comporta invece un programma su un PC, ad esempio Winamp (per chi lo conosce). Winamp e' essenzialmente un riproduttore di file musicali che ha una caratteristica, quella di poter cambiare skin. Cioe', se non mi piace la sua veste grafica, posso cambiarla con un semplice click del mouse, cosi' come i colori ed altri aspetti del programma. Un livello di flessibilità e di adattabilità molto diverso dall'esempio fatto sopra relativo alla radio, dove per cambiare qualcosa devo accendere il saldatore... ecco, una radio definita dal software e' una radio dove per cambiare le caratteristiche fondamentali di funzionamento devo solo agire sul software che, se scritto bene, permettera' una flessibilità molto elevata, tale da poter cambiare la banda passante, il modo di demodulazione ed altri parametri del funzionamento semplicemente con un click del mouse o la pressione di un tasto sulla tastiera.

E dove entra il software parlando di radio ? Lo vedremo tra un attimo. In una radio tradizionale, il segnale ha un formato analogico, cosi' detto perche' viene rappresentato mediante analogie tra grandezze fisiche, e tendenzialmente con una gamma continua di valori. Se vogliamo usare un computer per trattare, elaborare, gestire i segnali radio, bisognera' trasformare questo segnale continuo in una serie di numeri, perche' i computer capiscono solo i numeri... questo lo si fa tramite un processo cosiddetto di 'digitalizzazione' con l'uso di un convertitore ADC che converte il segnale analogico, continuo, in una serie di numeri presi ad intervalli prefissati, periodici, che rappresentano una delle grandezze del segnale, tendenzialmente l'ampiezza ovvero il voltaggio. Abbiamo trasformato il segnale da analogico a digitale. Ora e' possibile dare questa serie di numeri in pasto ad un programma software che possa fare di esso quello che noi vogliamo che lui faccia. La domanda spontanea che viene subito in mente e' :

PERCHE' ? A che scopo questa complicazione ?

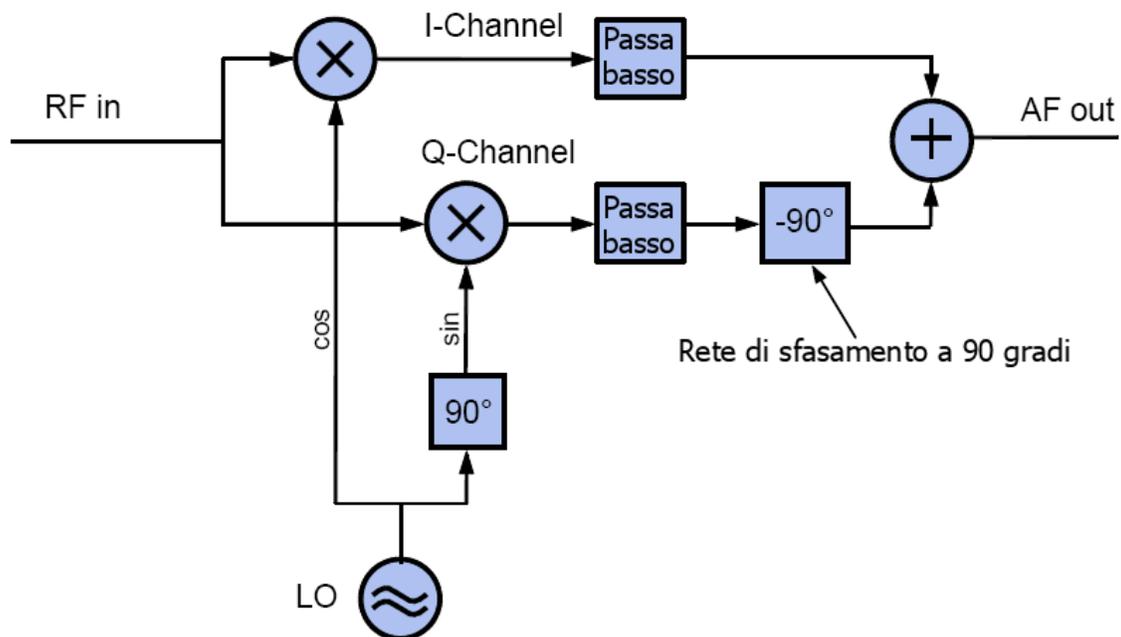
Alcune delle possibili risposte potrebbero essere :

- **Flessibilità** – non occorre piu' usare il saldatore per cambiare il funzionamento della radio, basta cambiare il software.
- **Economicita'** – il software non costa nulla (almeno finche' c'e' qualcuno che lo scrive gratis...)
- **Prestazioni** – Nessun filtro hardware puo' stare alla pari con un buon filtro software. E lo stesso si puo' dire per i demodulatori SSB, AM, FM, etc.
- **Ripetibilità** – Nessuna dipendenza da tolleranze di componenti o dal loro invecchiamento nel tempo.

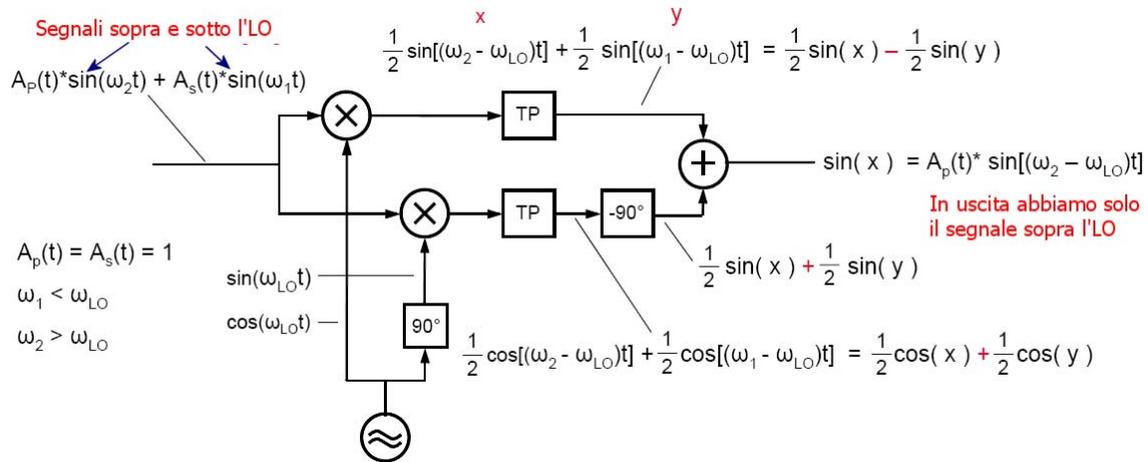
- **Nuove funzioni – Possibilita' di display dello spettro, waterfall, registrazione di una fetta di banda larga anche alcuni MHz, per poi riascoltare con comodo le stazioni presenti (es. DX, Contest, etc.), controllo remoto della radio, etc.**

Il gioco vale sicuramente la candela. A mio modesto avviso (ma non solo mio) l'introduzione della filosofia SDR e delle sue tecnologie e' comparabile all'avvento della SSB nel mondo AM, con la rivoluzione che cio' comporto' all'epoca.

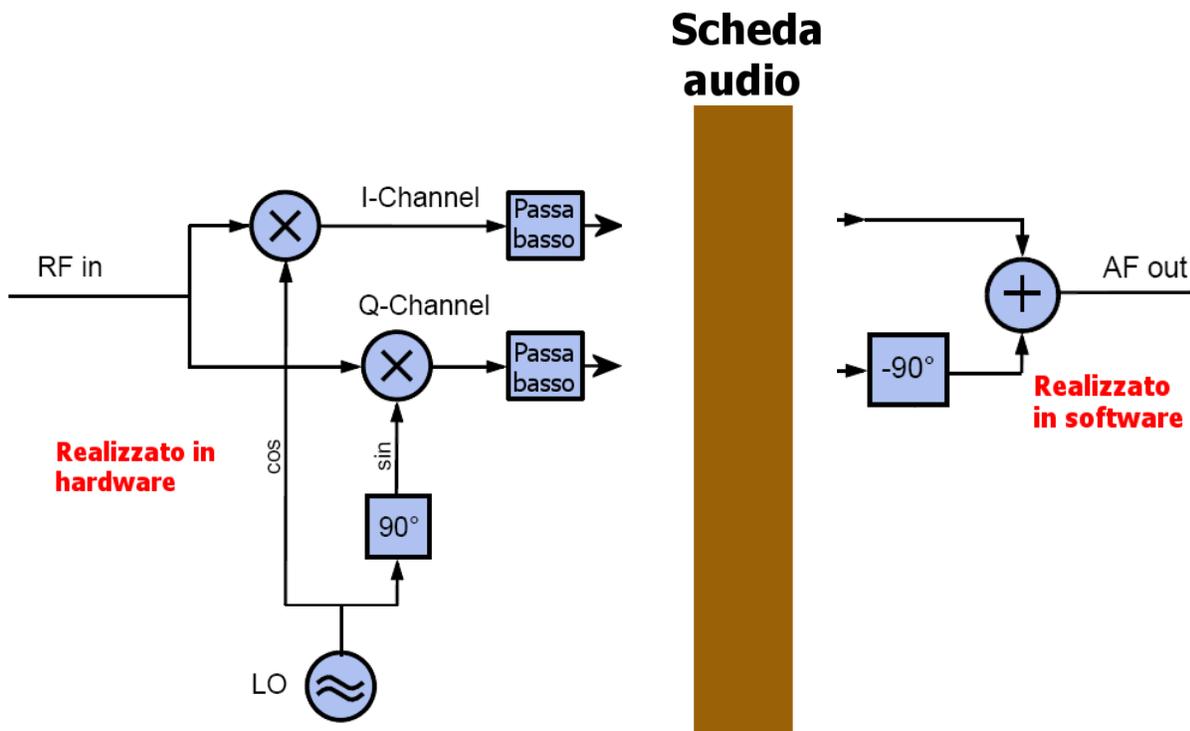
Vediamo ora come tutto questo si puo' realizzare in pratica. Diciamo subito che esistono due principali scuole di pensiero, ognuna con i suoi pro e i suoi contro, con costi e prestazioni differenti. La prima di queste e' in pratica la realizzazione dell'antico sogno della conversione diretta seguita da una rete di 'sfasamento', che come implementazione hardware circola gia' da alcuni decenni, ma con la palla al piede della necessita' di una rete di sfasamento molto precisa e soprattutto costante nel tempo. In pratica il segnale da ricevere viene convertito a 'zero IF', cioe' si usa un oscillatore locale per il mixer con la stessa frequenza del segnale da ricevere. In altre parole, il ricevitore a conversione diretta. Pero', come molti sanno, il ricevitore a conversione diretta ha lo svantaggio grave di mescolare insieme il segmento di frequenze superiore alla frequenza dell'oscillatore locale con quello inferiore. Ad esempio, se il mio LO ha una frequenza di 7050 kHz, un segnale CW a 7049 kHz e uno a 7051 kHz producono entrambi una nota a 1 kHz, e non c'e' verso di poterli separare dopo il mixer. Questo e' inaccettabile in un ricevitore che voglia avere anche una minima patina di serietà. Allora, sempre rimanendo a livello di implementazione hardware, quello che era stato pensato di fare e' di avere l'oscillatore locale in grado di generare due segnali alla stessa frequenza, ma sfasati tra di loro di 90 gradi. I due segnali vanno a due mixer separati, e i prodotti di conversione vengono inviati ad una rete di sfasamento realizzata con resistenze e condensatori di precisione (e stabili nel tempo...) in modo tale da poter ricombinare i due segnali in uno solo, ma con la possibilita' di poter scegliere o la banda superiore o quella inferiore rispetto alla frequenza dell'oscillatore locale. Schematicamente la cosa e' in questi termini :



Matematicamente (per chi fosse interessato), la cosa e' descrivibile in questi termini :



Questi sono concetti conosciuti gia' dagli anni '30, ma che sono stati ostacolati nella loro implementazione effettiva dalla difficolta' di realizzare una rete di sfasamento hardware precisa e stabile, che consentisse reiezioni della banda non voluta di almeno 50 - 60 dB. Con l'avvento di PC veloci e di schede audio in grado di campionare segnali ad almeno 48 kHz di frequenza di campionamento, si e' pensato allora di operare questo cambiamento nell'architettura :



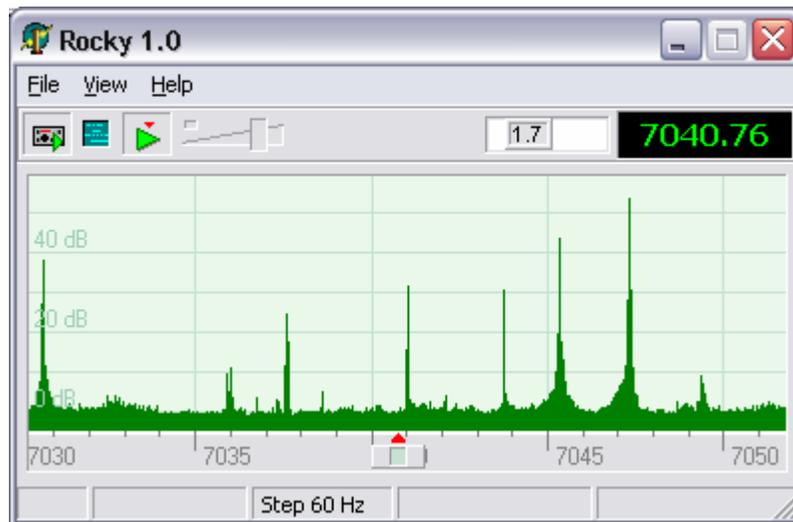
La scheda audio viene usata solo per il suo convertitore ADC che, essendo stereo, e' esattamente quello che ci serve per trasformare in digitale i due segnali I e Q (come vengono detti) in uscita dal mixer doppio. Nessun altro componente della scheda audio, ne' processore di segnale, ammesso che ne abbia, viene usato in questo tipo di applicazione.

Lo sfasamento finale di 90 gradi e' ora operato nella parte software, ed e' possibile usare algoritmi, tipo il trasformatore di Hilbert, anche questi noti da almeno 50 anni, ma che era difficile

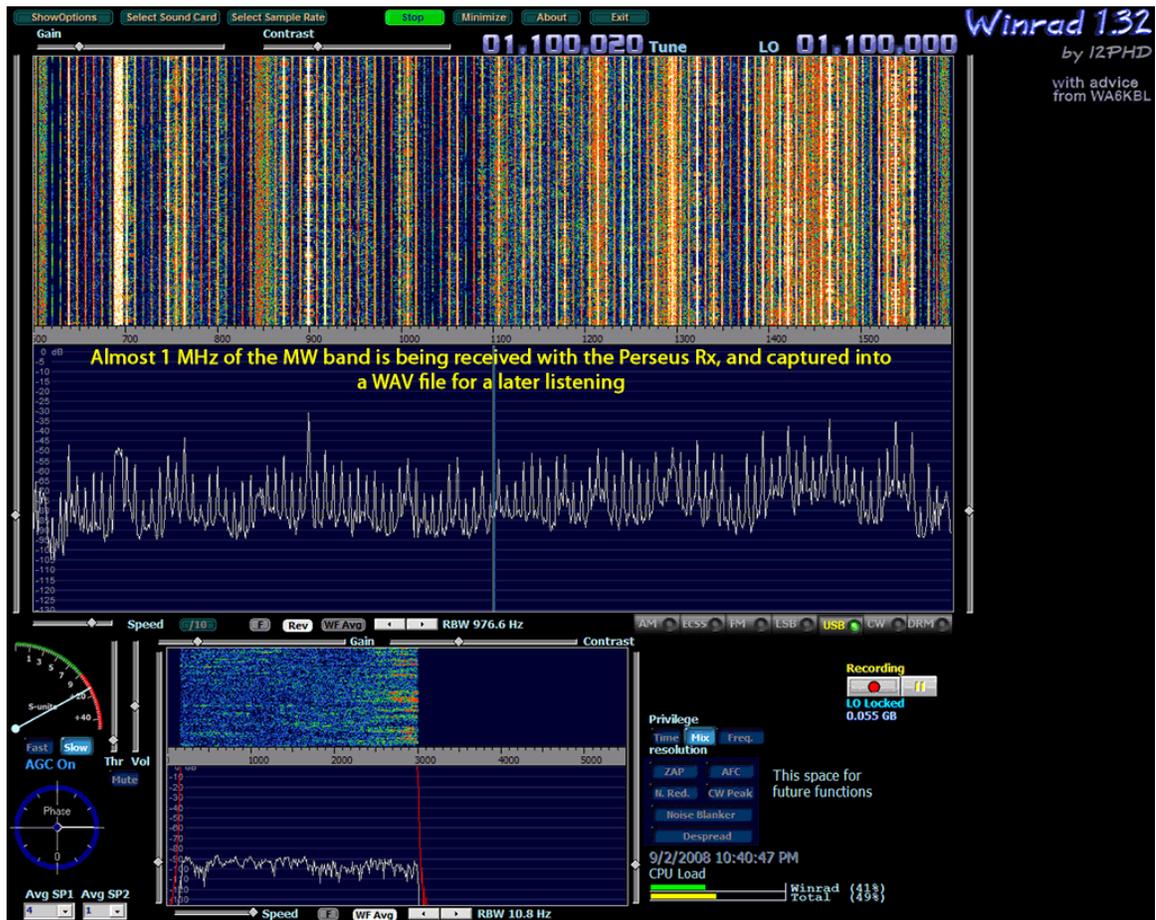
utilizzare per la lentezza dei computer fino a non molto tempo fa. Una implementazione software dello sfasatore non ha più i difetti di quella hardware. Non ci sono tolleranze di componenti con cui avere a che fare, né modifiche di valori nel tempo. Con qualche precauzione si riesce ad arrivare anche a 80 dB di reiezione della banda non voluta.

Naturalmente l'appetito viene mangiando, ed allora, visto che costa poco farlo, non occorre accendere il saldatore, basta qualche riga di codice, già che ci siamo viene naturale aggiungere alla parte software magari una filtro passa banda con frequenze di taglia basso e taglia alto regolabili, un filtro notch, un denoiser, un noise blanker ed altre funzioni accessorie, come un display dello spettro del segnale ricevuto, ed anche un waterfall ed altre cose dettate dalla fantasia dell'implementatore. Un paio di esempi di programmi di questo tipo possono essere i seguenti :

- Rocky, scritto da Alex Shovkoplyas, VE3NEA, scaricabile dal sito Web <http://www.dxatlas.com/Rocky/>
- Winrad, scritto da Alberto di Bene I2PHD, scaricabile dal sito Web <http://www.weaksignals.com>

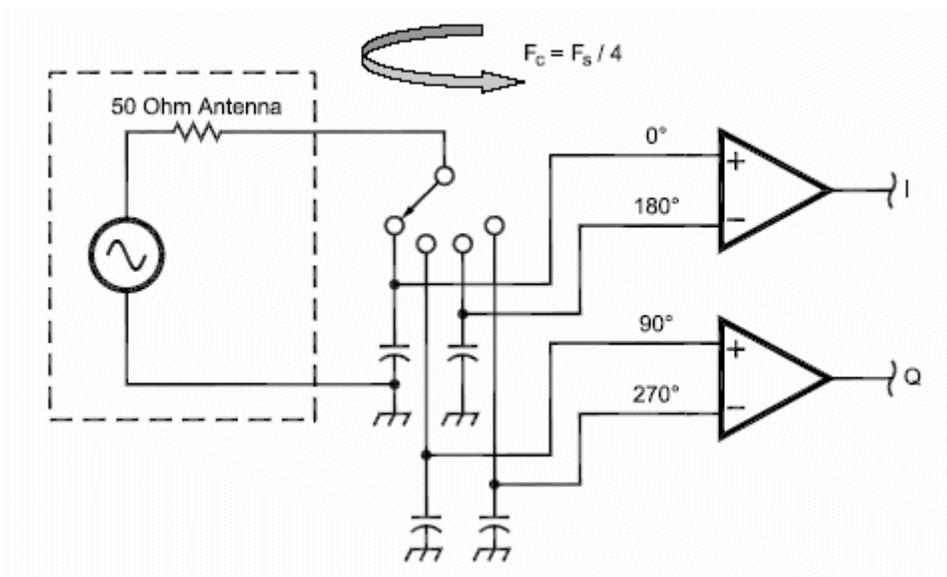


La schermata principale di Rocky



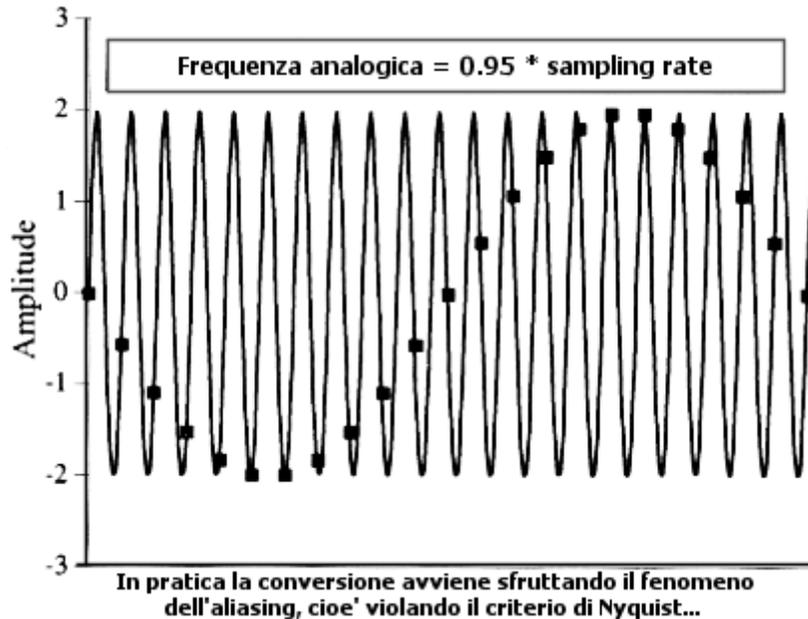
La schermata principale di Winrad

Ritornando alla parte hardware, molto spesso il doppio mixer viene sostituito da un circuito noto anche esso già da molti anni, ma recentemente riproposto da Dan Tayloe N7VE. Viene impropriamente definito come "Tayloe mixer", anche se lui non è l'inventore. Una dizione più corretta potrebbe essere QSD, cioè Quadrature Sampling Detector.. Schematicamente lo si può rappresentare in questo modo :



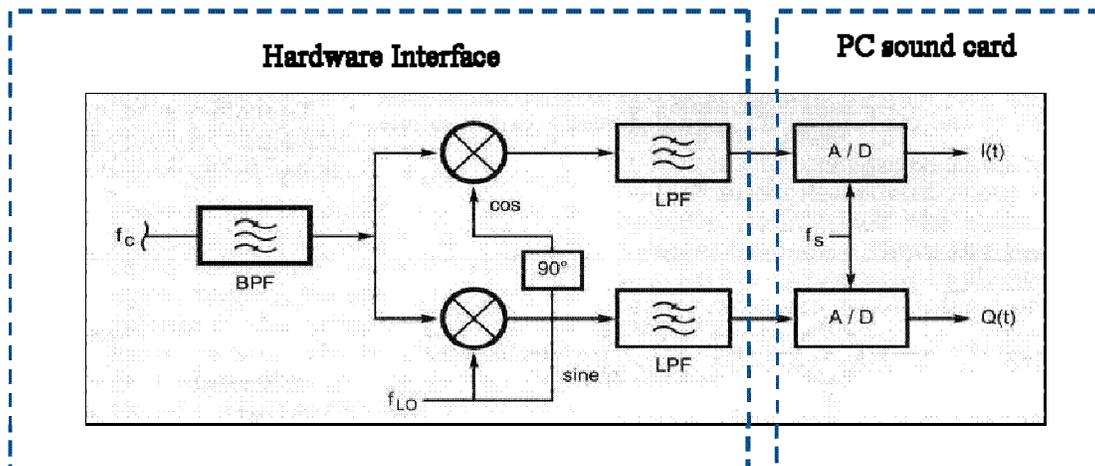
Ovviamente il commutatore non e' un commutatore meccanico, ma un demultiplexer analogico, ad esempio l'FST3125 della Fairchild. Se viene pilotato con una frequenza pari a quattro volte quella che si vuole ricevere, l'effetto e' quello di avere i quattro condensatori che lavorano in sample-and-hold, due a due con fasi invertite. I due amplificatori operazionale a basso rumore che li seguono ricostruiscono le componenti I e Q del segnale, come nel caso dei due mixer separati. Questo circuito e' molto valido, il suo unico difetto e' quello di essere sensibile alle armoniche dispari del segnale desiderato. Questo fa si' che in genere venga preceduto da un filtro passa basso con frequenza di taglio poco superiore a quella del segnale ricevuto.

Un modo di vedere come funziona il QSD e' di considerare che in pratica il mixer si comporta come un campionario che viola il criterio di Nyquist, perche' la frequenza di campionamento e' in pratica uguale alla frequenza del segnale campionato :



Ricapitolando, questo e' lo schema di base della parte hardware di un ricevitore SDR che si basi sulla conversione a zero IF tramite un QSD :

Quadrature Sampling Architecture



Questo tipo di architettura viene usato sia in SDR di alte prestazioni, come il FlexRadio 5000, come anche in kit veramente economici, come la serie Softrock, dal costo di una ventina di dollari USA. E, accettando il fatto che questi Softrock sono a frequenza centrale fissa, determinata da un quarzo, le loro prestazioni sono decisamente molto buone, se la scheda audio che li segue sul PC e' una scheda ad alta dinamica, come la Delta 44, oppure la ancora migliore E-MU 1212M.



Uno dei kit della famiglia Softrock, costo 12 dollari USA, compresa la spedizione...

Quello che abbiamo visto sinora rappresenta una delle due principali architetture utilizzate per la SDR. Il segnale viene convertito a "zero IF", con un mixer in quadratura di fase. Le

due componenti I e Q così ottenute vengono poi trasformate in rappresentazione numerica (cioè digitale) usando solitamente il convertitore stereo ADC di una scheda audio di buona qualità, ad alta dinamica (le migliori, e più costose, arrivano anche a 120 dB), con risoluzione di ampiezza che nel caso delle schede migliori è di 24 bit, e frequenza di campionamento che può arrivare, sempre nel caso delle schede migliori, fino a 192 kHz.

Se campioniamo a 192 kHz il segnale in uscita da un mixer in quadratura di fase che abbia un LO con frequenza pari a 7100 kHz, quello che si ottiene è uno spettro di frequenze che va da $7100 - 96$ kHz (metà di 192 kHz) = 7004 KHz fino a $7100 + 96 = 7196$ kHz, in pratica tutta la banda dei 40 metri, tenendo conto dell'ultimo allargamento ottenuto fino a 7200 kHz. La sintonia fine da 7004 a 7196 kHz per sintonizzare il segnale voluto la si fa con il software, il quale permette anche di scegliere il modo di demodulazione (LSB, USB, CW, AM, FM, etc.) e il filtro passabanda desiderato. Questo è solo un esempio. Cambiando la frequenza dell'LO si sposta il segmento ricevibile di 192 kHz da qualche altra parte, centrata appunto sulla frequenza dell'oscillatore locale.

L'architettura a campionamento diretto RF

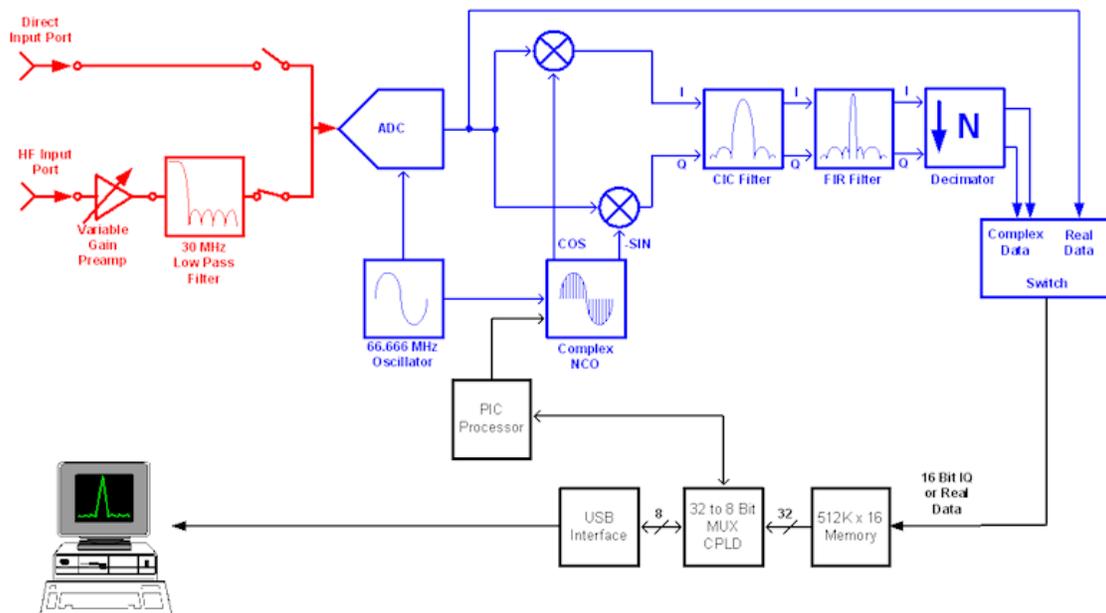
Vediamo ora il secondo tipo di architettura possibile per una radio SDR, e cioè il campionamento diretto RF. Anche questo è sempre stato l'antico sogno di chi si è occupato di questi argomenti già da decenni. Cioè collegare direttamente l'antenna al convertitore ADC, eliminare completamente qualsivoglia traccia di stadio analogico, trasformare immediatamente il segnale ricevuto in una serie di numeri e con quelli fare tutto quello che la disciplina del Digital Signal Processing insegna che si può fare. Un sogno che però si è sempre scontrato con la inadeguatezza dell'hardware a disposizione. Usando la formulazione più semplice del criterio di Nyquist, per poter adeguatamente campionare un segnale con frequenza massima F Hz, occorre che la frequenza di campionamento sia almeno $2 \cdot F$, cioè il doppio. Meglio se è un po' più alta, perché usando esattamente $2 \cdot F$, si richiede poi un filtro di ricostruzione di lunghezza infinita, irrealizzabile in pratica. Questo implica che, se ci limitiamo alle HF fino a 30 MHz, la frequenza di campionamento non può essere inferiore a 60 MHz. Solo recentemente sono comparsi sul mercato dei convertitori ADC che potessero lavorare a questa frequenza e con un numero adeguato di bit.

Vediamo di analizzare questo tipo di architettura. Il segnale viene campionato, diciamo, a 66 MHz con una risoluzione di ampiezza di 14 o 16 bit. Ora, 66 milioni di campioni al secondo moltiplicato due bytes per campione, produce un data rate enorme, molto più ampio di quanto il processore di un PC sia in grado di elaborare in tempo reale. Senza contare i problemi per portare quella mole di dati all'interno del PC... se lo si potesse ipoteticamente fare, poi sarebbe possibile sintonizzare una frequenza qualunque da 0 a 30 MHz solo con il software. Ma bisogna essere realistici, non è possibile farlo, e allora occorre in qualche modo ridurre questo enorme data rate. Il segnale è già a rappresentazione numerica, e allora viene spontaneo pensare di usare dei gate array veloci, del tipo programmabile dall'utente finale, cioè un **Field Programmable Gate Array**, in altre parole una FPGA. I principali produttori sono Cypress, Altera e Xilinx, ed tutti hanno dei chip adatti alla bisogna. In pratica è come scrivere un programma per il PC, solo che il programma si traduce poi in una riconfigurazione delle porte interne del chip (che può averne decine o anche centinaia di migliaia) in modo da fargli 'eseguire' quel programma. Solo che ora la esecuzione è puramente hardware, senza la interpretazione del firmware di basso livello del Pentium o dell'Athlon, quindi con una velocità nettamente superiore.

E quale programma andremo a scrivere per la FPGA? Sorpresa... un programma che la faccia comportare più o meno come il ricevitore a conversione diretta che abbiamo visto prima... solo che ora abbiamo già dei numeri con i quali lavorare e non dei segnali analogici. E con i numeri le cose cambiano (spesso in meglio, ma qualche volta anche in peggio). Ad esempio, nel campo analogico un mixer è un componente delicato, che richiede cura nella progettazione nella costruzione. Nel campo numerico un mixer è... una moltiplicazione, niente più! E infatti, pensandoci bene, abbiamo sempre sentito parlare di rivelatore a prodotto (non a rapporto, come qualche ignorante li chiama... il rivelatore a rapporto non è un mixer, ma solo un demodulatore per la FM, quindi una cosa completamente diversa). Magari ci siamo in passato chiesto il perché di questa dizione, ma ora il perché è chiaro. E un mixer, pardon, una moltiplicazione, non ha i difetti di non linearità, intermodulazione, generazione di spurie che affliggono un mixer analogico. Semplicemente è il 'mixer' perfetto. L'oscillatore locale per la conversione dovrà semplicemente generare una sinusoide campionata alla stessa frequenza di campionamento del segnale, e questa serie di numeri saranno quelli per i quali verranno moltiplicati i valori numerici che rappresentano il segnale, per ottenere la IF. E, ricordando che occorre un doppio segnale di LO, in quadratura di fase, vediamo un altro vantaggio. Numericamente parlando, avere un secondo segnale LO sfasato di 90 gradi è semplicissimo. Anziché il seno di ωt basta prenderne il coseno. In campo analogico non è così semplice, e anche un piccolissimo errore di qualche centesimo di grado porta ad un rapido e tragico deterioramento della ricezione della banda non voluta. Un altro punto a vantaggio della soluzione numerica. Quindi, tirando le somme, quello che facciamo nel programma della FPGA è di generare un LO in quadratura di fase, lo usiamo per portare il segnale (numerico) ricevuto a zero IF, e poi lo ricampioniamo con una frequenza più adatta ad

essere elaborabile dal PC. Ovviamente questo ricampionamento ridurrà lo spettro di frequenze rappresentabile dal nostro flusso di numeri, e tendenzialmente con le radio SDR oggi presenti sul mercato si può andare da 100 kHz a 2 o anche 4 MHz. All'interno di questa fetta poi ancora una volta sarà il software del PC a fare la sintonia fine. Ora la scheda audio non serve più, per due motivi. Il principale è che non c'è più bisogno di convertire da analogico a digitale, abbiamo già dei numeri. E poi siamo ben oltre le velocità che sono gestibili da una scheda audio, nata per altri scopi. Il mezzo per portare i dati all'interno del PC è la porta USB, anche se stanno uscendo dei prodotti che usano la porta Ethernet. Ovviamente anche in questo tipo di architettura il mixer numerico in quadratura di fase produce le due componenti I e Q, che sono quelle che realizzano una perfetta descrizione analitica del segnale ricevuto, permettendo poi al software di operare qualunque tipo di demodulazione su di esso.

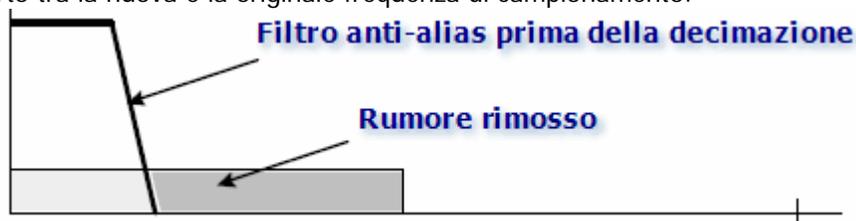
Forse una figura aiuta più delle parole :



Questo è lo schema di massima dell'SDR-14 della RFspace. La parte in rosso rappresenta la zona dove il segnale è ancora analogico (essenzialmente un amplificatore a guadagno variabile e un filtro passabasso, entrambi opzionalmente escludibili), mentre le parti in blu e nero indicano una rappresentazione numerica del segnale. Possiamo vedere chiaramente il doppio LO, i due 'mixer', i filtri anti-alias che liseguono e il decimatore, cioè il componente che riduce la frequenza di campionamento. Forse qui occorre fare una parentesi per chiarificare un punto. Nell'SDR-14 (come anche nell'SDR-IQ e nel Perseus) la campionatura viene fatta con una risoluzione in ampiezza di 14 bit. Questo può far storcere il naso allo sprovveduto.. come, solo 14 bit ? Ma applicando la nota relazione

$$\text{SNR} = (6.02N + 1.76) \text{ dB} \text{ dove } N \text{ è il numero di bit}$$

si ottengono solo $6.02 * 14 + 1.76 =$ circa 86 dB di range dinamico... non bastano mica... In questo ragionamento non si è però tenuto conto del cosiddetto 'processing gain' che si ottiene quando si effettua la decimazione, ovvero il downsampling del segnale. Infatti, nel processo di downsampling si preserva la parte del segnale che interessa, mentre il rumore, che è a banda larga e uniforme, viene drasticamente ridotto, ed esattamente di un fattore pari al rapporto tra la nuova e la originale frequenza di campionamento.



Tenendo conto di questo, la formuletta di prima va modificata come segue :

$$\text{SNR} = (6.02N + 1.76) + 10 \cdot \log_{10}(F_s/F_d) \text{ dB}$$

Dove F_s è la frequenza di campionamento originaria, e F_d quella dopo la decimazione. Se facciamo un esempio, in cui da 66 MHz si decima fino a 100 kHz, si ottiene un processing gain, cioè un aumento della dinamica di circa 28 dB che, sommati agli 86 trovati prima, fanno 114 dB, cioè una cifra di tutto rispetto. E questo con un convertitore da 14 bit. Ultimamente sono comparsi ADC da 160 MHz e 16 bit, quindi rifacendo i conti si arriva a cifre di dinamica ben superiori. Ovviamente questi sono i valori che da' la teoria. Nella pratica le cose vanno un po' peggio, perché gli ADC non sono componenti perfetti, e hanno i loro problemi che però non è opportuno trattare in questa sede.

La via del campionamento diretto RF sembra ormai essere quella definitiva, con maggiori promesse, appunto grazie alla comparsa di componenti con specifiche adeguate alle necessità. La eliminazione quasi completa degli stadi analogici è garanzia di affidabilità, ripetibilità, assenza di tarature, semplicità e basso costo. Già' attualmente i ricetrasmittitori giapponesi adottano una via intermedia, e cioè campionano a RF non il segnale dell'antenna, ma il segnale della ultima IF, ma sicuramente sui loro tavoli da disegno ci sono i progetti per le radio di domani che vedranno il convertitore ADC spostato più vicino all'antenna. Questo perché avere una rappresentazione numerica offre molta più flessibilità, e nel caso della reiezione di immagine elimina tutte le problematiche di tolleranza dei componenti e della loro stabilità nel tempo.

Inoltre esiste la possibilità di sfruttare la formulazione corretta del teorema di Nyquist, che lega la frequenza di campionamento alla larghezza di banda del segnale da digitalizzare. Questo vuol dire che posso sfruttare a mio vantaggio il fenomeno dell'aliasing, a patto che prenda le opportune precauzioni. Facciamo un esempio. Il Perseus, attualmente campione di dinamica nel campo delle SDR a campionamento diretto, usa un ADC con clock da 80 MHz. Questo vuol dire che la prima zona di Nyquist va da 0 a 40 MHz, metà della frequenza di campionamento. Da 40 a 80 MHz, la seconda zona, abbiamo dei segnali, a banda invertita, che vanno in alias con quelli della prima zona. Da 80 a 120 MHz, la terza zona, i segnali non sono più invertiti e anche loro sono in alias. Ad esempio, un segnale a 83 MHz sarebbe indistinguibile da un segnale a 3 MHz, come pure da uno a 77 MHz. Continuando, da 120 a 160 MHz si ha la quarta zona, a spettro invertito. In questa fetta ricade la banda dei 2 metri, da 144 a 148 MHz, in alias, rispettivamente, con la banda da 16 a 12 MHz (lo spettro è invertito) più le altre nelle altre zone di Nyquist. Quindi, se metto in fronte all'ADC un bel filtro (analogico) passa banda che elimini tutti i segnali che non siano compresi tra 144 e 148 MHz, ottengo la digitalizzazione della banda dei 2 metri con un ADC con frequenza di clock relativamente bassa, senza pericoli di alias, grazie al filtro passabanda. Occorre che l'ADC abbia al suo interno una circuiteria di sample-and-hold sufficientemente veloce, adatta per la frequenza dei 144 MHz, ma questo generalmente è un requisito rispettato dalle specifiche degli ADC di ultima generazione.

Quanto sopra fa definitivamente ritenere la strada del campionamento RF quella giusta da seguire, ed infatti nuovi prodotti, oltre ai già citati SDR-14, SDR-IQ e Perseus, sono in fase di completamento, tipo il QS1r di Phil Covington N8VB e la board Mercury del progetto HPSDR.

Il software per la SDR

Abbiamo citato in precedenza due programmi usati per la parte piu' importante di una SDR, cioe' il software, che poi e' quello che anche da' il nome a questa filosofia di radio. Vanno attribuiti ai meriti o ai demeriti del software le caratteristiche di filtraggio, di bonta' e fedelta' di demodulazione, denoising etc. di una radio realizzata come SDR.

Posiamo ora vedere in un dettaglio maggiore il programma Winrad, con qualche nota sulla sua architettura.

Il suo uso principale è quello di essere messo dopo un mixer NZIF (Near Zero IF) in quadratura di fase, cioè con uscite I e Q, ma, seppure con qualche limitazione, accetta anche segnali in banda base e reali, cioè con solo la componente I. Concettualmente lo si deve vedere come una ultima IF sintonizzabile, con larghezza complessiva di banda pari alla frequenza di campionamento impostata, che, nel caso di scheda audio input, può arrivare a 192 kHz. Quindi il programma presenterà tutto lo spettro dei 192 kHz, con possibilità per l'utente di fare una ricezione point-and-click, cioè di sintonizzare il segnale desiderato con il mouse, impostando quindi i limiti del filtro passabanda, il modo di ricezione (AM/ECSS/FM/USB/LSB/CW) ed eventuali filtri per la riduzione del rumore. Ovviamente nel caso di collegamento ad un hardware campionatore RF tramite la porta USB, sarà quest'ultimo a dettare la frequenza di campionamento da impostare. Questa e' l'architettura di massima che sarà brevemente discussa nel seguito.

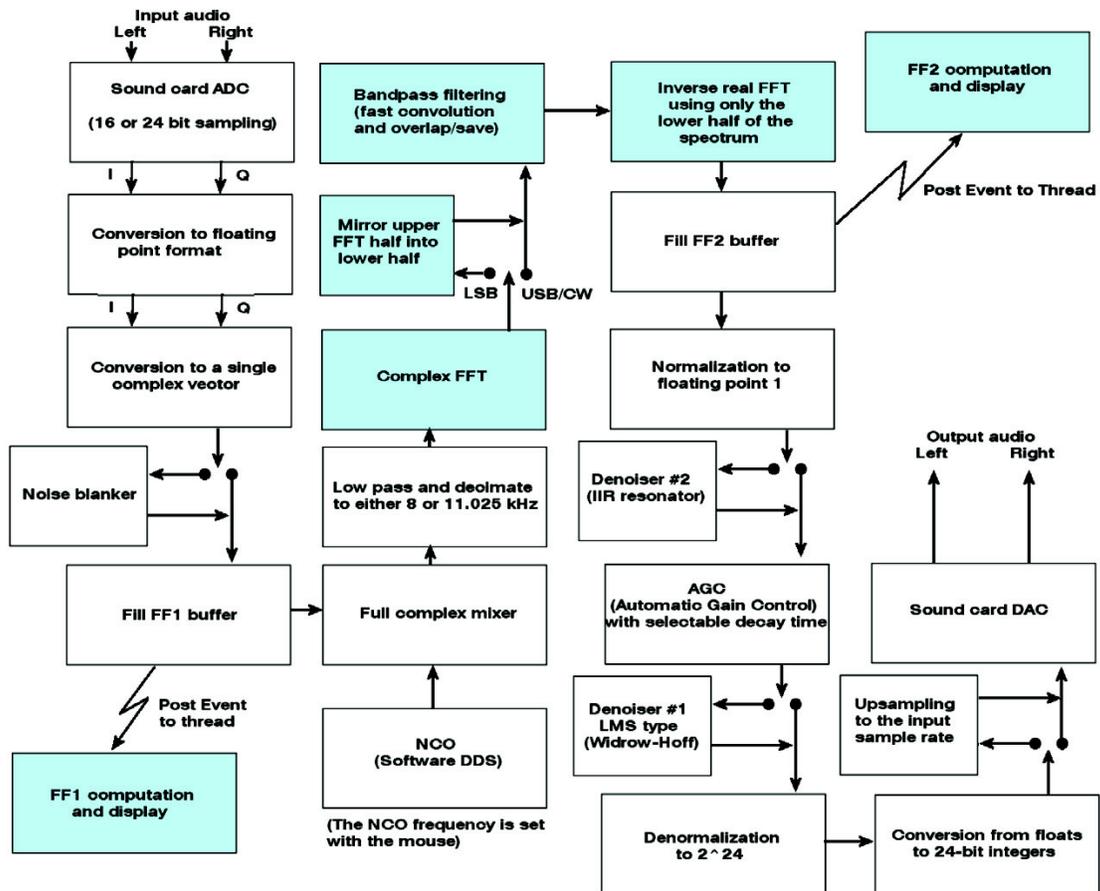


Figure 1 - Architettura di massima

Nella **fig. 1** è rappresentata la architettura del thread di processing audio, che è separato dal thread di gestione interfaccia utente e dai due threads di calcolo e display delle due FFT (Fast Fourier Transform) con relativi waterfall. Nel seguito darò qualche breve nota sul funzionamento dei blocchi di questo thread, gli altri threads non sono particolarmente interessanti da un punto di vista concettuale.

La digitalizzazione del segnale viene effettuata dalla scheda audio, con la scelta del subsystem da usare, WMME a 16 bit oppure ASIO a 24 bit, sempreché la scheda audio presente nel PC abbia installati dei drivers ASIO (non tutte ne sono fornite). Subito dopo la digitalizzazione si converte immediatamente ad un formato floating point, che sarà usato nel resto del thread, fino al momento di restituire il segnale alla scheda audio per la riproduzione.

I segnali I e Q rispettivamente presenti sul canale sinistro e destro vengono conglobati in un unico vettore complesso. A questo punto si effettua opzionalmente una fase di noise blanking, in cui si individuano e si eliminano gli spikes nel segnale che con grossa probabilità rappresentano solo dei disturbi. La soglia di intervento del noise blanker è regolabile. Il segnale così ripulito viene accodato in un buffer per il calcolo e il display della prima FFT nel pannello superiore dell'interfaccia utente. Questo permette non solo di visualizzare lo spettro del segnale in arrivo, ma anche di fare un point-and-click per impostare l'oscillatore locale software (NCO) di conversione in banda base.

Proseguendo con la analisi del percorso del segnale nel thread di processing audio, siamo ora arrivati al full complex mixer. Il nome "full complex" indica che accetta un segnale complesso sia nella porta RF che nella porta LO. Il risultato è che si ha una conversione senza produzione di frequenza immagine, in quanto entrambi i segnali sono di tipo analitico, quindi senza componente specchiata sull'asse delle frequenze. Come detto precedentemente, la frequenza dell'oscillatore locale viene impostata con il mouse, in funzione del segmento che si vuole ricevere. Il segnale LO è generato da un NCO (Numerically Controlled Oscillator), cioè dall'equivalente software di un DDS. Visto che qui non si hanno i limiti imposti dall'hardware, questo NCO ha una phase table piuttosto lunga, esattamente 262144 entries, in modo da minimizzare i prodotti spurii. Il risultato della conversione è il segnale desiderato in banda audio. A questo punto è conveniente, nel caso che la campionatura iniziale fatta con la scheda audio fosse piuttosto alta, fare una decimazione in modo da portare il sampling rate ad un valore inferiore, compatibile con la banda audio da riprodurre. Considerando che l'audio finale non avrà componenti superiori a circa 4.5 kHz (eccetto quando si ricevono segnali digitali nella modalità DRM), la frequenza di downsampling è stata scelta pari a 11025 Hz.

Ovviamente la decimazione è preceduta da un filtro FIR passabasso per rispettare Nyquist, che si arrabbia se un segnale campionato a frequenza F contiene delle componenti spettrali di frequenza superiore a F/2. Arrivati a questo punto abbiamo il segnale desiderato, che, nel caso della SSB, ancora contiene la banda superiore e quella inferiore rispetto alla frequenza zero, che è quella di sintonia. Potremmo usare l'arcinoto e vecchio metodo del trasformatore di Hilbert per selezionare la banda desiderata, però la considerazione che dobbiamo ancora applicare un filtro passabanda con fianchi variabili e regolabili con il mouse, e che questo filtro per motivi di performance opererà nel dominio della frequenza, ci fa propendere per un metodo alternativo ad Hilbert.

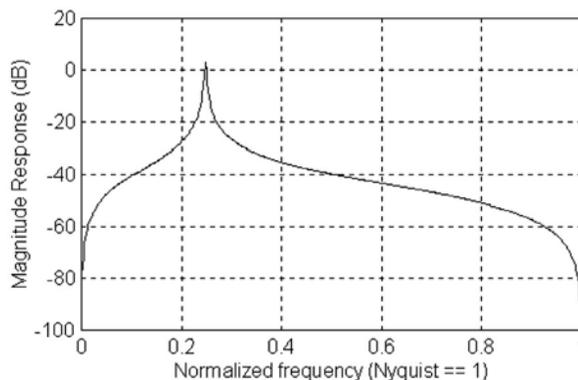
Quello che facciamo è calcolare una FFT complessa del segnale. Il risultato di questa FFT è uno spettro che, a differenza di quello ottenuto da una FFT di un segnale reale, non è coniugato simmetrico nella sua seconda metà rispetto alla prima metà. Invece quello che abbiamo è che la prima metà è lo spettro della parte positiva delle frequenze (ricordiamoci che ora stiamo lavorando a IF zero), mentre la seconda metà è lo spettro della parte negativa delle frequenze, e cioè, in altre parole, della banda LSB rispetto al carrier virtuale a zero Hz. Capito questo, il resto viene di conseguenza. Se vogliamo la LSB, altro non facciamo che prendere la metà superiore dello spettro FFT, ne calcoliamo il coniugato simmetrico, lo specchiamo, e

rimpiazziamo la prima metà dello spettro con il risultato. Se invece vogliamo la USB, semplicemente usiamo la prima metà della FFT. Ora compiamo un piccolo imbroglio. Facciamo finta che lo spettro così ottenuto sia il risultato di una FFT reale, non complessa, e ne calcoliamo la FFT inversa reale. Per magia, otteniamo un segnale audio reale, che è quello che volevamo. Abbiamo quindi demodolato la USB oppure la LSB senza scomodare Hilbert. Provare per credere.

Nella descrizione del paragrafo precedente è stato anticipato un passaggio, e cioè il filtraggio passabanda. Visto che abbiamo il segnale nel dominio della frequenza, viene spontanea l'idea di usare il metodo della fast convolution, suggerito, come detto, anche da considerazioni di performance. In pratica si calcola il kernel del FIR che rappresenta il filtro voluto, lo si smussa agli estremi con una finestra adeguata (in Winrad è stata usata una finestra di Blackman ottimizzata), se ne calcola la FFT, si moltiplica questa FFT per la FFT del segnale, e poi si applica l'algoritmo overlap/save per trasformare la convoluzione circolare in una lineare. Nel caso specifico il kernel del FIR viene calcolato on-the-fly in funzione dell'impostazione fatta dall'utente dei limiti del passabanda, simulando un numero di prese (taps) del FIR molto alto, e cioè 1537. Questo numero sarebbe assolutamente inaccettabile per una implementazione del filtro nel dominio del tempo, ci metterebbe una eternità come tempo di esecuzione. Usando invece la fast convolution il tempo derivante dalla lunghezza lo si paga una sola volta, cioè al momento del calcolo del kernel. Questo numero equivalente di prese così alto permette di avere una reiezione fuori banda ampiamente maggiore di 120 dB, assolutamente sufficiente per ogni esigenza. Abbiamo quindi riottenuto un segnale audio nel dominio del tempo, di formato reale. Potremmo già inviarlo alla scheda audio per la riproduzione. Ma abbiamo ancora qualcosetta da fare. Innanzi tutto prepariamo un buffer per seconda FFT, quella che usiamo per il display nel pannello inferiore. Ogni volta che il buffer è pieno, segnaliamo un Event al thread della FF2. che asincronicamente farà il suo lavoro. Una piccola nota: sia il thread della FF1 che quello della FF2 hanno una priorità di esecuzione inferiore al thread audio. In questa applicazione, ascoltare è più importante che vedere!

Ora abbiamo un segnale audio ma dobbiamo ancora applicare un paio di filtri opzionali. Il primo si basa sul fatto che il pannello della FF2 permette all'utente di specificare qual è il pitch CW gradito all'utente stesso. Con questa informazione calcoliamo i coefficienti di un filtro IIR a due zeri e due poli. I due zeri vengono semplicemente piazzati nell'origine, mentre i due poli (uno il coniugato simmetrico dell'altro) vengono posti all'interno del circolo unitario sul piano di Argand-Gauss della trasformata Z. Se proprio vogliamo fare un paragone con il campo analogico, possiamo dire che questo filtro si comporta come un risonatore con un Q variabile. La frequenza di accordo del risonatore è quella impostata mediante la scelta del CW pitch, mentre il fattore Q lo si può scegliere facendo comparire, mediante il tasto destro del mouse sul bottone corrispondente, uno slider di regolazione. L'effetto auditivo è esattamente quello che si avrebbe con un circuito risonatore hardware. Se si sceglie un Q troppo alto (che in questo caso significa che i due poli si avvicinano molto al circolo unitario) il filtro è più selettivo, con però il pericolo di ringing, che può arrivare fino al punto di rendere i punti e linee del CW indistinguibili tra di loro, perché la stessa frequenza di manipolazione CW è al di fuori della banda passante del filtro!

Nella figura a lato e' possibile vedere la risposta in frequenza del risonatore, simulate mediante Matlab, per un fattore Q medio. Aumentando questo fattore, il risultato è che il picco diviene più stretto, più aguzzo.



Dopo questo filtro di picco incontriamo la routine AGC, che si preoccupa di rendere più o meno costante il livello del segnale. Il tempo di attacco è molto basso, mentre il tempo di rilascio ha un hang di poche centinaia di millisecondi, dopodiché ha una velocità di decadimento selezionabile tra due valori, che nel pannello principale di Winrad sono indicati con Fast e Slow.

Dopo l'AGC troviamo il denoiser di tipo LMS, con aggiustamento dinamico dei coefficienti h secondo Widrow-Hoff. In pratica l'algoritmo privilegia i segnali che hanno una correlazione temporale alta, mentre attenua gli altri. In termini tecnici si tratta di un FIR di lunghezza 64 prese, con una linea di ritardo (quella che determina la correlazione temporale) di lunghezza variabile a seconda che il modo di ricezione sia il CW oppure la SSB. Infatti nel caso della voce l'intervallo di correlazione è parecchio inferiore a quello del CW.

Bene, il segnale è arrivato quasi al termine del suo lungo viaggio all'interno del programma. Non resta che denormalizzarlo, cioè lo riportiamo ad un fondo scala di 2^{24} , e poi lo convertiamo da formato floating point a formato intero. Ora abbiamo veramente finito. Il segnale viene inviato contemporaneamente al canale destro e sinistro della scheda audio per la riproduzione.

Conclusione

Bene, siamo arrivati alla fine di questo cammino descrittivo delle architetture hardware e software di una SDR, cercando di descriverne i pregi e i difetti, nonché i punti a cui fare attenzione. Spero di non avere confuso ulteriormente le idee, ma di avere invece dissipato qualche nebbia e dubbio sull'argomento. Chi volesse approfondire queste tematiche potrebbe leggersi questi articoli (in lingua Inglese), molto descrittivi e di grande aiuto.

<http://www.arrl.org/tis/info/pdf/98qex003.pdf>

<http://www.arrl.org/tis/info/pdf/98qex022.pdf>

<http://www.arrl.org/tis/info/pdf/98qex013.pdf>

<http://www.arrl.org/tis/info/pdf/98qex019.pdf>

<http://www.arrl.org/tis/info/pdf/020708qex013.pdf>

<http://www.arrl.org/tis/info/pdf/020910qex010.pdf>

<http://www.arrl.org/tis/info/pdf/021112qex027.pdf>

<http://www.arrl.org/tis/info/pdf/030304qex020.pdf>

<http://www.arrl.org/tis/info/pdf/020810qex041.pdf>

Grazie per aver avuto la pazienza di leggere sino a qui e buon divertimento con la SDR !

73 Alberto I2PHD