

Prontuario per l'uso della nuova acquisizione ed analisi di Auriga

November 27, 2003

Contents

1	Come far partire un'acquisizione	1
2	Locazione dei dati	2
3	Come avere una prima impressione della qualità dei dati: Baud-line	3
3.1	Off-line	3
3.2	On-line	3
4	Come fare un lock-in digitale	3
4.1	Il processo lock-in con <i>ada</i>	4
4.2	Visualizzazione dei dati utilizzando Paw++	6
4.3	Utilizzazione dei dati con windows	7
5	Visualizzazione dei dati con Matlab	8
5.1	visualizzazione dello spettro di potenza (Matlab)	8
6	Visualizzazione dei dati con ada - Vega - Root	9
6.1	Acronimi di ada	9
7	Altre Operazioni utilizzando Linux	9

1 Come far partire un'acquisizione

Appena si entra nella stanza dove ci sono gli acquisitori ci si trova di fronte un terminale della Philips. Con questo terminale si possono visualizzare varie macchine che si selezionano con "KVM-switch" (Keyboard, Video and Mouse switch), uno scatolino che si trova sulla scaffalatura dove sono alloggiati i PC. Il metodo per selezionare consiste nello schiacciare un pulsante di destra sullo scatolino, operazione che farà comparire un numero che corrisponde al PC selezionato che si potrà comandare da tastiera e mouse davanti al terminale della Philips.

La corrispondenza tra il numero a il PC e' molto semplice: 1 = *auriga1* (disk server), 2 = ... (non utilizzato), 3 = *auriga3* (nis e samba server), 4 = *auriga4* (primo daq lato porta) 5 = *auriga5* (secono daq fondo stanza), 6 = *auriga6* (pc su rack di analisi) 7 = *auriga7* (pc su rack di analisi), 8 = ... (non utilizzato). Ricapitolando le macchine di acquisizione sono

Led	Nome macchina	descrizione
A	auriga4	Acquisitore sul rack vicino alla porta di ingresso
B	auriga5	Acquisitore sul rack sul fondo della stanza

Selezionata la macchina di acquisizione (4 o 5 sul "KVM-switch") si deve fare partire l'acquisizione. Nel caso si debbano fornire username e password queste si trovano scritte in una busta sotto il terminale Philips. Una volta entrati nell'ambiente grafico di Linux (KDE) bisogna far partire i processi per l'acquisizione (comando: "**upAll**") e la finestra grafica che la gestisce (comando "**upGui**") da due qualunque finestre di shell. L'interfaccia grafica parte in modalit  "locked" per cui per dare i comandi all'acquisitore bisogna andare nel men  a tendina in alto, clickare su **Unlock** e selezionare **Unlock buttons**.

La finestra grafica che si apre (se gi  non la si trova aperta) contiene un elenco di voci a cui dovrebbe seguire la dicitura "*INITIAL*". Se cos  non fosse clickare "**Reset**". A questo punto cliccando prima **Boot** e poi due volte sulle varie voci CHE INTERESSANO si possono selezionare i parametri dell'acquisizione voce per voce. Importante: quando si settano i parametri dell'acquisitore ricordarsi di tirare gi  il numero di run di acquisizione (che, se non settato diversamente, viene automaticamente incrementato di uno rispetto a quello del run precedente). Quando tutto   stato settato a fianco di ogni voce deve comparire la scritta *Loaded*. A questo punto per confermare il setting e renderlo operativo si deve andare con il mouse alla prima riga (*DAQ*) e schiacciare il tasto **Config**. Su ogni voce compare allora la scritta *Configured*. Per fare partire l'acquisizione si fa **Start**, per farla finire **Stop**, **Abort**, **Shutdown** in serie.   possibile che ci siano dei problemi dovuti al driver vxi che gestisce l'interfaccia tra il PC e le schede di acquisizione. Non esiste una procedura standard per risolvere i problemi: chiedere agli analisti dati.

2 Localione dei dati

I dati dei run della Test Facility fisicamente si trovano sul disk server auriga1. Tutte le macchine Linux di auriga (aurigapc8, aurigapc9, aurigapc11, aurigapc12, aurigapc19, aurigapc21, aurigapc24) montano i dischi del disk server *auriga1* (/aufs/disk1, /aufs/disk2, /aufs/disk3 si usano per i dati raw mentre /aufs/disk4 conterra' i dati analizzati): pertanto l'analisi dei dati si pu  fare da una qualsiasi di queste macchine. Inoltre si possono usare per l'analisi i PC "server" auriga6 e auriga7 e la PC-FARM dei Laboratori (plcfe1.lnl.infn.it). I

dati raw si trovano tipicamente nella directory /aufs/data1. Se non lo si vede dalla propria account su un qualsiasi dei PC Linux, chiedere ai softwaristi. Per analizzare i dati occorre avere settato alcune variabili ambiente. Il comando per sapere se l'ambiente e' settato e' "**which ada**". Nel caso non venga riconosciuto il comando ada per settare le variabili d'ambiente basta dare il comando **source /users/auriga/.ada/rh72-#versione** (ad esempio #versione = 0.60.00). In caso di problemi chiedere ai softwaristi: è cosa che si fa una volta per tutte modificando il file .cshrc della propria home directory.

3 Come avere una prima impressione della qualità dei dati: Baudline

Per dare una prima occhiata ai dati si usa un programma che calcola gli spettrogrammi utilizzando la tecnica delle Short Fourier Transform (SFT). Il programma è **baudline**: ha un help online alla pagina <http://www.baudline.com/manual/index.html>. Il fondoscala del programma bf baudline è 32768 ($= 2^{15}$), questo numero corrisponde al fondoscala dell'acquisitore settato come descritto nel precedente paragrafo.

Il programma può essere usato sia *On-line* che *Off-line*.

3.1 Off-line

Per farlo partire bisogna dare il comando

```
ada -r NumeroRun -t daq -f FileConf.cfg --baudline --verbose
```

Per sapere cosa è il file di configurazione di ada *FileConf.cfg* si veda il prossimo paragrafo.

3.2 On-line

Si tratta di utilizzare lo script **aom** (AURIGA on-line monitor) che chiama sempre *ada* (e' equivalente a settare la variabile d'ambiente *OMNIORB_CONFIG* ("*setenv OMNIORB_CONFIG \$HOME_DAQ/omni/omniORB-auriga5.cfg*" e a lanciare il comando **ada --online --baudline --verbose**) che fa da "spia" i dati presenti nei processi di acquisizione. Il comando è semplicemente **aom auriga#**

dove # e' il numero del PC dove avviene l'acquisizione dei dati che ci interessano.

4 Come fare un lock-in digitale

Il programma sviluppato per l'analisi dati di AURIGA che si chiama *ada* (AURIGA Data Analysis). Questo programma ha molte funzioni tra cui anche la

possibilità di applicare dei lock-in digitali ai dati raw. Per confrontare i lock-in digitali di *ada* con quelli del lock-in analogico bisogna tenere conto delle diverse normalizzazioni e del fatto che l'acquisitore ha un'impedenza di ingresso di 50 Ω . Da una calibrazione sperimentale abbiamo appurato che se si misura l'uscita a 50 Ω dell'amplificatore SR560 **per un Volt al lock-in digitale si leggono 1.47 Volt al lock-in analogico** (cioè circa $\sqrt{2} \sim 1.41$). Il lock-in digitale è normalizzato per dare come risposta per un segnale $A \cos(\omega_0 t)$ all'ADC la grandezza A . Si possono utilizzare al massimo 3 Lockin contemporaneamente. Si possono avere comunque tutti i lockin necessari facendo run diversi di *ada*. I dati dei lockin vengono salvati in directory diverse e possono essere consultati, trasformati in file ascii o visualizzati in seguito.

4.1 Il processo lock-in con *ada*

Il processo lock-in è fatto in due fasi:

- Si converte a bassa frequenza il segnale da analizzare utilizzando l'usuale procedura di modulazione in ampiezza con una frequenza selezionabile
- Si filtra il segnale a bassa frequenza con un filtro passabasso di Butterworth di ordine selezionabile con un numero di poli $N_P = 1 \dots 8$. La funzione di trasferimento di questo filtro è $|H(\omega)|^2 = 1/(1 + (\omega/\omega_c)^{2N_P})$, dove ω_c è la frequenza di taglio a -3dB (cutoff frequency). Il normale passa-basso utilizzato dal lock-in analogico si recupera come caso particolare $N_P = 1$ e $\tau_{ck} = 1/\omega_c$. La risposta in frequenza e fase di questo filtro si trova sul sito <http://www-users.cs.york.ac.uk/fisher/mkfilter/>

Un'altra possibilità per studiare ampiezza e fase del filtro è l'utilizzo del package *Signal and Systems* di Mathematica. Il setting dei lock-in si trova nei files con estensione *.cfg*. Se non specificato esplicitamente (vedi sotto) il programma di analisi (*ada*) va a cercare il file di configurazione *ada.cfg* nella directory di lavoro. Per "settare" i lock-in bisogna editare il proprio file di setup *FileConf.cfg* utilizzando un editor ¹.

Il file di configurazione contiene tutti i parametri dell'analisi. Un esempio di questo file si trova nella directory `"/users/vedovato/rh72/ada-#versione/cfg"`. Per settare i lock-in bisogna cercare la riga "LOCKIN setup" ed agire sulle seguenti righe

¹Consiglio di non usare il diabolico editor "vi" ma di usare l' *Advanced Editor* che si avvia selezionando l'icona raffigurante un pennino posizionata in basso sulla sinistra della schermata. In alternativa, se non si trova l'icona, usare i comandi **kwrite** oppure **kedit** che lanciano entrambi editors con interfaccia grafica e user-friendly

Lockin_size		Numero di Lock-in usati (max 3).
Lck_block_size		dimensione del blocco dati in uscita al lockin (di solito non si cambia)
Lck_dcm_factor	N_{dcm}	La frequenza con cui si campiona l'uscita del lock-in è $4882.8125/N_{dcm}$. Nota che N_{dcm} deve essere una potenza di due e vale al max Lck_block_size.
Lck_real_1	ν_{LI}	Frequenza del primo lock-in
Lck_imag_1	$\Delta\nu$	Mezza larghezza di banda del filtro Butterworth. Cioè si selezionano le componenti in frequenza tra $\nu_{LI} - \Delta\nu$ e $\nu_{LI} + \Delta\nu$
Lck_lp_bu_1		Ordine del filtro Butterworth (massimo 8)
Lck_hold_samples_1	N_λ	Volendo il programma cerca il massimo di N_λ campioni contigui e lo registra

Attenzione alla varianza σ_{lck}^2 del rumore bianco dopo i lockin digitali. Nel caso classico si aveva $\sigma_{lck}^2 = S_0/\tau_{lck}$, dove S_0 è il livello dello spettro bianco bilatero e $1/\tau_{lck} \equiv \pi(2\Delta\nu)$ corrisponde al tempo di integrazione del lock-in. Nel caso del filtro Butterworth abbiamo $\sigma_{lck}^2 = (2S_0)(2\Delta\nu)/((2N_P/\pi)\text{Sin}(\pi/2/N_P))$, che si riduce al solito risultato del passa-basso per $N_P = 1$. Da ricordare anche che la pendenza (slope) dell'istogramma ottenuto con la variabile $\rho = x^2 + y^2$ è $1/(2\sigma_{lck}^2)$.

Infine il file *NomeFile.cfg* sotto la voce "DATA PATHS" contiene

root_read_data_path	Che è la directory dove il programma si va a prendere i dati raw da analizzare con i lock-in digitali. Il path deve essere scritto tra virgolette.
root_write_data_path	Che è la directory dove viene scritto il file di uscita del lock-in

Una volta settato il file di configurazione si fa partire il lock-in con il seguente comando:

ada -f FileConf.cfg -t TipoAnalisi -r NumeroRun -b TempoIniziale -d Durata

dove

- *FileConfig.*: è il nome del file di configurazione
- *TipoAnalisi*: è il tipo di analisi che si vuole fare. Nel nostro caso si vuole fare il lock-in e quindi si deve scrivere *lck*
- *NumeroRun* : è il numero di run che si vuole analizzare. Per esempio se voglio il run 43 scrivo *43*

- *TempoIniziale*: è il GPS time ² al quale si vuole fare iniziare l'analisi dei lock-in. Se non uso l'opzione "-b" il programma fa l'analisi iniziando dal primo istante dell'acquisizione.
- *Durata*: è il tempo in secondi che si vuole fare durare l'analisi. In alternativa a questo comando si può usare -e *tempo di fine* dove *tempo di fine* rappresenta il GPS time a cui voglio fermare l'analisi. Se non uso l'opzione "-d" il programma **ada** continua l'analisi fino alla fine dell'acquisizione.

Per avere più informazioni sul programma **ada** usare il comando

man ada

4.2 Visualizzazione dei dati utilizzando Paw++

Il programma *ada* si trova in ambiente Linux quindi per visualizzare i dati velocemente si deve usare un programma che funzioni su Linux. Come per esempio *Root*, *Matlab* o *Paw++*. In questa sezione descriviamo alcuni comandi di elementari con *Paw++*. Per fare partire il programma si scrive **paw++**. Come prima istruzione

filecase keep

che serve a fare in modo che le maiuscole e le minuscole vengano viste come caratteri diversi. A questo punto si possono fare molte cose:

- Creare per esempio i vettori *Tempo*, *Ampsq* e *Fase* contenenti il tempo il quadrato dell'ampiezza e la fase del lock-in i cui dati sono registrati nel file ascii *NomeFile.ascii* (Nota che *Paw++* ha bisogno di un file ascii e quindi devi prima trasformare il file raw dei lock-in in file ascii)
ve/re *Tempo,Ampsq,Fase NomeFile.ascii*
- Manipolare i vettori così creati con operazioni matematiche mediante il package *sigma*. Per esempio se si vuole creare il vettore *Amp* ampiezza del lock-in
sigma *Amp=sqrt(Ampsq)*. Se invece si vuole creare un vettore *V* lungo 1000 con $V(i) = i$
sigma *V=array(1000,1#1000)*
- Plottare un vettore verso il numero di campioni. Per esempio per plottare il vettore ampiezza *Amp* del lock-in
ve/draw *Amp*
se avessi voluto plottarlo dal campione *ini* al campione *end* avrei dovuto scrivere **ve/draw** *Amp(ini:end)* (se omettevo *end* plottava tutti i campioni a partire da *ini*). Per avere una scala per esempio logaritmica in "y" devo scrivere **opt logy** per tornare a quella lineare **opt liny**

²Il GPS time è il tempo in secondi dal 6 gennaio 1980 h00:m00:s00. A differenza dell'UTC, il GPS time non ha "leap seconds". In realtà si può usare un formato "umano" per immettere le date in *ada* (dare il comando *man ada* e vedere le opzioni -B e -E). Il GPS time ha il vantaggio che i files in formato *frames* dei dati hanno come nome il GPS time del dato iniziale.

- Fare l'istogramma di un vettore. Per esempio per farmi la Boltzmaniana delle ampiezze al quadrato del lock-in avrei dovuto scrivere
ve/plot *Amps²*
Se di questo istogramma voglio il valore medio e la deviazione standard devo scrivere (prima di *ve/plot*) **opt stat**
- Plottare un vettore verso l'altro. Per esempio per plottare l'ampiezza del lock-in *Amp* verso il tempo *Time*
gra *NPunti Amp Time*
dove *NPunti* è il numero di campioni che si vogliono plottare (per plottare tutti i dati mettere un *NPunti* grandissimo)
- Fittare un istogramma
hi/fit *NUM Funzione*
dove *NUM* e' il numero intero che rappresente l'istogramma e *Funzione* e' la funzione di fit. In modo automatico Paw riconosce le funzioni *g* = *gaussiana*; *e*=*esponenziale*; *p0...p8*=*polinomi di grado 0...8*. Altre funzioni devono essere scritte in Fortran su file esterno.
- Fittare un vettore come ordinata rispetto ad un'altro come ascissa. Per esempio se voglio fittare il vettore fase *Fase* del lock-in rispetto al vettore tempo *Time*
ve/fit *Buzzi*
dove la funzione di fit *Funzione* può essere come per l'istogramma: gaussiana **g**, esponenziale **e** oppure polinomiale **p0...p8**
- Personalizzare un istogramma: Per prime cosa si scelgono il binning, il valore massimo ed il numero identificativo dell'istogramma (il numero *NUM*) con il comando **1d**, poi si riempie l'istogramma personalizzato con il vettore *V* **ve/hfill** *NUMV*, poi si plotta l'istogramma **hi/pl** *NUM* ed infine si fitta per esempio esponenziale **hi/fit** *NUMe*
- Se si vuole fare un grafico con una scala predefinita
null *x_{min}, x_{max}, y_{min}, y_{max}*
e poi si riempia il frame con il grafico desiderato.

Qualsiasi sia il comando che si usa se si hanno dubbi sulla sintassi si usi l'help nella seguente maniera ? **Comando**

4.3 Utilizzazione dei dati con windows

Per prima cosa si devono trasformare le uscite dei lock-in scritti in formato FRAME in files scritti in foermato *ASCII*:

ada -f *FileConfig*. **-t** *TipoAnalisi* **-r** *NumeroRun* **-c** *ascii*

fate attenzione di mettere nel file di configurazione l'indirizzo dove andare a prendere i files lock-in.

Per poi usare i files di *ASCII* in windows si deve andare su un PC ed aprire *My Network Places*, poi aprire *AURIGA3* e infine aprire la propria cartella personale. A questo punto si importano i dati con *Origin* andando a cercarli in *My Network Places*, *AURIGA3*, cartella personale. Il programma **ada** quando converte un lock-in genera un file a tre colonne: **Tempo**[secondi], **Ampiezza**²[V²], **Fase**[rad.].

5 Visualizzazione dei dati con Matlab

Per prima cosa si fa partire il Matlab con il comando **matlab**, poi si devono estrarre i dati dai frame usando **frgetvect** oppure **frextract**.

Questi due comandi scritti dai "framisti" di Virgo estraggono rispettivamente il vettore dati i tempi come indice o il frame. La documentazione completa di questi comandi, così come di qualsiasi altro comando di Matlab, si ha scrivendo **help comando**. Come esempio si consideri:

```
dati=frgetvect('NomeFile','Richiesta',Inizio(GPS),Nr di Frame da processare);
```

che scrive nel vettore "dati" ³ la *Richiesta* usando i dati presenti in *NomeFile*. Ricordarsi di metter in *NomeFile* tutto il percorso completo della dir: i dati si trovano in /aufs/data#. Opzioni possibili:

Richiesta=Antenna, per importare i dati raw

Richiesta=LockinRho-n, per importare l'ampiezza del n-esimo lock-in

Richiesta=LockinPhs-n, per importare la fase del n-esimo lock-in

Per sapere cosa contiene un file in formato FRAME (e quindi per sapere cosa usare come *Richiesta*) basta rientrare nella shell Linux e dare il comando **Fr-Dump -i Nome file**. Nota che in matlab per fare un comando shell basta mettere davanti al comando il carattere "!" seguito dal comando

5.1 visualizzazione dello spettro di potenza (Matlab)

Se con il comando **frgetvect** ho scelto di importare i dati raw posso farne lo spettro di potenza. Il comando è:

```
[sp,fr]=psd(dati,NFFT,Fs>window)
```

per vedere cosa è *dati* si veda il paragrafo precedente. Per la corretta sintassi si veda l'help in linea. Si tenga conto che per noi la frequenza di campionamento è $Fs=4882.8125$ mentre il numero di punti dello spettro è una potenza di due $NFFT=2^N$: in MatLab si può direttamente scrivere 2^N . Per avere lo spettro in V/\sqrt{Hz} devo dividere *NomeSpettro* per la frequenza di campionamento Fs cioè si

³Questo nome si puo' scegliere a piacere: è il nome del file di uscita di Matlab. I dati di uscita di MatLab sono tra parentesi quadre, i dati di ingresso tra tonde.

fa $NuovoNome=NomeSpettro/4882.8125$. A questo punto posso plottare lo spettro per esempio in asse logaritmico con il comando **semilogy(f,NuovoNome)**. Lo spettro plottato è la media di $Size/NFFT$ spettri dove $Size$ è il numero di dati che sono stati importati in matlab

6 Visualizzazione dei dati con ada - Vega - Root

In generale (ma esistono alcune eccezioni) con **ada** per visualizzare dati in ingresso al processo di analisi bisogna utilizzare il flag **-g10** (g sta per output grafico), per visualizzare dati in uscita (dopo l'analisi) bisogna esplicitare il flag **-g1**. Il flag **-i** in **ada** indica il modo interattivo: il programma aspetta sempre per una risposta/prompt (premere la barra spaziatrice) prima di analizzare al frame successivo.

Per trovare una prima stima per i valori degli zeri, dei poli e del livello del bianco:

ada -r NumeroRun -t fme -f FileConf.cfg --phase 1 -g1 -i --verbose

Una volta che si è soddisfatti del fit visualizzato da **ada** e dal valore del χ^2 del fit si può utilizzare l'output alfanumerico di questo comando per inserirlo nel file di configurazione *FileConf.cfg*: il formato è lo stesso sicchè basta fare un copia-incolla con il mouse.

6.1 Acronimi di ada

Nella tabella.1 si trovano le spiegazioni di alcuni acronimi e abbreviazioni usati in ada.

fme	<i>full mode estimation</i>
-----	-----------------------------

Table 1: Acronimi e abbreviazioni in ada.

7 Altre Operazioni utilizzando Linux

Se sono connesso come utente "Buzzi" e voglio connettermi come utente "Utente" senza uscire da "Buzzi" basta fare

su Utente

se invece voglio aprire un'altra finestra grafica indipendente devo schiacciare uno tra i tasti $Ctrl+Alt+"F1 \div F6"$ e logarmi nella shell alfanumerica con il nome che mi pare. Poi faccio

startx -1:0

(l'1" sta ad indicare la finestra grafica da usare, normalmente uso "0", per altre finestre posso scegliere 2,3,ecc.). Poi per passare da una finestra grafica ad un'altra uso i tasti Ctrl+Alt+F7 (per la "0",) oppure F8 per la "1" ecc..

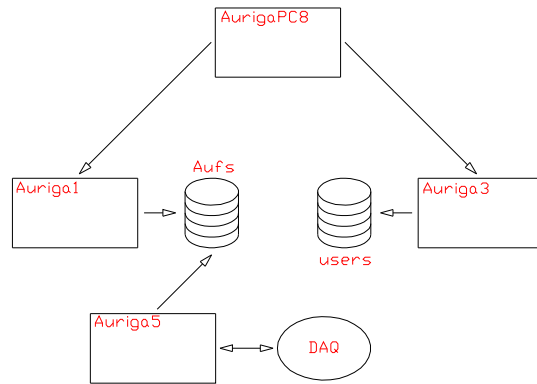


Figure 1: Architettura del sistema di acquisizione-analisi. Quando entro in un PC Linux (ad es. *aurigapc8*), questo ha montato i dischi di *auriga1* (file server) e *auriga3* (nis e samba server) che gli permettono di accedere rispettivamente ai dati in("/*aufs/data#*" *#=1,2,3*) dove vengono scritti in formato frame i dati raw acquisiti e ai dati in ("/*users*"), dove ogni utente ha la sua "home page" e potrebbe scrivere i risultati delle sue analisi. Un posto conveniente per scrivere i risultati delle analisi è anche */aufs/data4* specie se si hanno grosse moli di dati