

Inf. 81

SISTEMA DE ADQUISICIÓN Y CONTROL
PARA EL CONTINUO DE RADIO EN 1420,
1660 Y 3300 MHz

Federico A. Bareilles

ftp://mafalda.iar.unlp.edu.ar/pub/fede/doc/informe01.ps

ftp://ftp.iar.unlp.edu.ar/pub/fede/docs/continuo/

9 de febrero de 1999

informe01.pdf

1 Introducción

Las Radiofuentes pueden ser clasificadas en dos categorías: fuentes térmicas, que radían porque el medio está caliente, y fuentes no térmicas. En principio, varios mecanismos de emisión son responsables de estas emisiones, pero en la práctica un mecanismo de emisión es el que domina sobre los otros: la emisión de sincrotrón o bremsstrahlung magnético.

Otra división de las radiofuentes discretas es en: galácticas o extragalácticas, esto es totalmente independiente de los mecanismos de emisión, pero se encuentran predominantemente fuentes no térmicas entre las extragalácticas.

Con excepción de las líneas de emisiones térmicas de átomos, moléculas y la emisión de cuerpos sólidos, las emisiones en radio provienen de electrones libres. Estos electrones libres experimentan cambios de energía en cantidades arbitrarias (saltos de energía no definidos); el resultado son emisiones o absorciones con un espectro continuo.

Un electrón libre, sólo emite radiación si es acelerado. Los mecanismos de aceleración pueden ser estudiados por el espectro en el continuo de radio. Sólo dos mecanismos producen aceleraciones importantes: la radiación de frenado emitida por el electrón en un campo eléctrico y la radiación de frenado emitida por un electrón relativista en un campo magnético. Este último produce la radiación de sincrotrón, que es la fuente de radiación de la emisión no térmica.

2 Modos de Observación

En principio, son necesarios dos modos de observación: para fuentes puntuales y para fuentes (o regiones) extendidas.

2.1 Observación de fuentes puntuales

Consiste en realizar barridos en ascensión recta (AR) o declinación (D) de determinada longitud (en grados). Actualmente existen dos velocidades de barrido: rápida ($10^\circ/min$) y lenta ($15^\circ/hora$). Queda por determinar la posibilidad de variar las velocidades, a fin de obtener resultados aceptables para fuentes de 0.5 Jy (Ver Tabla 1)

La observación se podrá realizar con un file de observación (Meta-Lenguaje) en el que se especificarán las fuentes a observar y las calibradoras, debiendo el programa ir realizando los scans (barridos) sobre las fuentes en forma automática.

2.2 Observación de fuentes extendidas

Se observarán grillas que serán ingresadas en un file de observación (Meta-Lenguaje). Los barridos se harán en declinación (D) y con ascensión recta (AR) fija (seguimien-

Tabla 1: Los tres receptores de la Antena 1

Frec. [MHz]	BW [MHz]	HPBW [°]	T_{sis} [K]	$\frac{\Delta T_{rms}}{1SegInt.}$ [K]	$t_{integ.} / 0.5 \text{ Jy}$ [Seg]
1420	22	30.0	35 ± 2	$1.0 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$
1660	110	26.5	35 ± 2	$4.7 \cdot 10^{-3}$	$3.5 \cdot 10^{-3}$
3300	200	13.3	60 ± 2	$6.0 \cdot 10^{-3}$	$5.6 \cdot 10^{-3}$

0.5 Jy \equiv 0.08 K

to). El intervalo de muestreo espacial se hará respetando el teorema de Nisquit, es decir: menor que la mitad del ancho de haz para la frecuencia a la que se observe (ver Tabla 1).

Una vez observada la grilla, la reducción off-line debe permitir: promediar barridos, trazar línea de base, ajustar gaussianas y construir el mapa.

3 Software o Sistema

El desarrollo se hará íntegramente sobre Linux, pero teniendo cuidado que la interface con el usuario sea portable a otros entornos como Windows NT (ésto no se persigue como objetivo, pero sí deberá quedar la puerta abierta para que sea viable).

Se desarrollará inicialmente sobre el kernel 2.2.x que debería estar listo a los momentos de iniciarse la construcción del software. Esta versión, y las sucesivas dentro de la serie (2.2.x), está orientada al procesamiento con múltiples procesadores (actualmente hasta 32) y al uso de tareas (thread); si bien ésto está soportado en los kernel's 2.0.x, su desarrollo se hizo sobre 2.1.x y luego portado hacia atrás sobre el 2.0.x por Alan Cox. Otro cambio importante sobre el nuevo kernel es la eliminación de las `ioctl`.

Si bien el sistema no va a requerir el uso de múltiples procesadores, se hará uso de ésto sólo para que se tenga el dominio de esta tecnología para desarrollos futuros.

Para la sincronización de los procesos se usará una modificación del RTC (Real Time Clock) del kernel: RT-Linux; éste es un kernel en sí, pasando el kernel Posix de Linux a ser una tarea que depende de él.

El lenguaje elegido es el C y C++ por ser el que mejor se domina y en el que está escrito el OS (Operating System) y la mayoría del software en Linux. También podrá incluirse código en FORTRAN para los cálculos de posiciones y tiempo, aunque puede que se porten a C.

El Sistema estará constituido por tres partes bien diferenciadas como muestra la figura 1: Módulos, Bibliotecas e Interface con el usuario (GUI).

- **Módulos (primer capa de software):** es la parte que se comunica con el Hardware y el kernel, sólo éstos tendrán acceso a las IRQ's y ports. Funcionan en forma dinámica, es decir, se pueden insertar y retirar sin

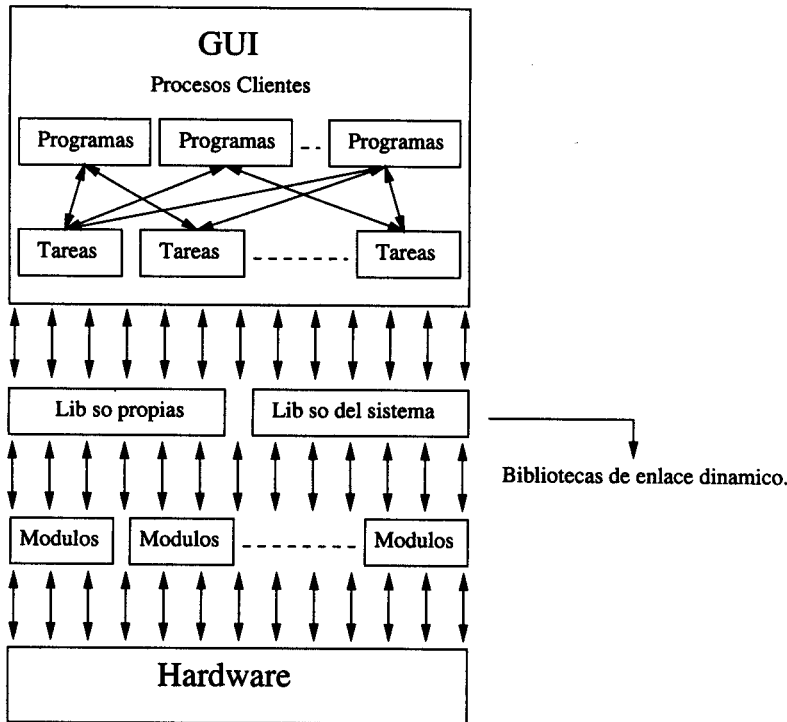


Figura 1: Capas del software

tener que apagar el sistema. Su nombre también podría ser drivers, pero en el mundo UNIX se conocen como módulos.

- **Bibliotecas:** son el nexo entre los módulos y la GUY. Sólo éstas podrán acceder a las funciones definidas en los módulos. Serán enlazadas dinámicamente (en tiempo de ejecución), por lo tanto podrán recibir mantenimiento o modificaciones sin tener que recompilar o alterar el software que se encuentra más arriba (GUI) o por debajo (módulos).
- **GUI:** es la parte más compleja de realizar. Sólo accederá las funciones definidas en las bibliotecas de enlace dinámico, las propias del sistema de adquisición y las del sistema operativo.

3.1 ¿ Por qué Linux ?

Es difícil responder esta pregunta sin entrar en fanatismos y pasiones que este OS genera.

Linux es un "UNIX" originalmente escrito para 80x86 (modo protegido); actualmente soporta otras plataformas como ALFA y SPARC. Soporta mutiprocésamiento, multitarea (real) y es multiusuario.

Un muy importante motivo es que los paquetes de reducción de datos usados en astronomía (IRAF, AIPS, STERLING, GREG, CLASS, etc) están basados en UNIX (o VMS); por su estructura no es posible que se los porte a otros entornos fácilmente y no está previsto que funcionen en otros OS que no estén relacionados con UNIX. Por lo tanto se puede hacer uso de las herramientas desarrolladas en estos paquetes para la manipulación de los datos (operaciones, despliegues etc.).

Otras características por las cuales elegir un sistema Linux son:

- Ofrece una variedad de proveedores.
- Es escalable.
- Hace uso de los recursos muy eficientemente.
- Permite la administración **remota** del sistema.
- Ofrece la capacidad de realizar cómputo remotamente.
- Posee capacidad de multiusuarios.
- Existe una gran variedad de software.
- Los estándares son independientes de los proveedores (POSIX).
- No le afectan los virus de hace 10 años diseñados para MS-DOS. Ni es posible actualmente infectarlo.
- Su código es de dominio público.
- Soporta hasta 32 sistemas de archivo diferentes.

3.2 Bibliotecas gráficas

También es importante al elegir un OS el disponer de bibliotecas gráficas que faciliten el uso de ventanas, botones, listados, etc. En este aspecto se evaluaron algunas bibliotecas existentes en Linux:

- MOTIF: No es de dominio público, es antigua y no tiene revisiones recientes. Su continuidad es dudosa.
- Qt: Actualmente son de dominio público (fines de 1998); el KDE está basado en ellas; son potentes pero su implementación es complicada y requieren de un gran esfuerzo. No posee una interface para generar formularios.

- **GTK:** Son similares a las Qt; nacieron como necesidad de tener bibliotecas gráficas de dominio público en Linux (las Qt no lo eran en un principio). El GNOME está basado en ellas. Se hicieron numerosos ejemplos y testeos de su uso, con buenos resultados pero con gran dificultad. No poseen una interfaz para generar formularios, pero los objetos son ubicados en forma automática muy eficientemente. Su port a otros OS está desarrollandose, pero aún no se a concluido. Si el desarrollo contara con más gente, éstas hubieran sido la elección adecuada.
- **XFORMS:** son de dominio público pero su código no está liberado. Su utilización es simple y posee una interface para la generación de formularios o ventanas. Se desarrolló incluso una aplicación con ellas a modo de test. Aunque aún no se han difundido mucho, viene creciendo su uso en el mundo UNIX. Existe un port de éstas a Windows NT, por lo que permitirían portar la GUI a ese OS. Pueden ser utilizadas desde varios lenguajes como C, C++, FORTRAN, PERL, Ada95 y python. Sólo hacen uso de las funciones de Xlib (X Window System Protocol V11).

Por todo ésto se las adoptó como las bibliotecas gráficas de la GUI; su contra es que el código no fue liberado. Pero es probable que suceda con el tiempo; de no ser así sería el único punto en el cual no se dispone del código fuente.

3.3 Almacenamiento de Datos

Los datos se almacenarán en formato FITS standard, por ser el más cómodo para su manejo con paquetes de reducción y su transporte; ya sea en: diskettes, zip drive , cintas o TCP/IP.

El usuario podrá especificar otro tipo de salida; por el momento sólo se contemplarán: tablas ASCII y el formato de Drawspec. También se podrán ver los dump de los datos que se adquirieron en binario y sin formato; ésto se prevee sólo para la depuración del programa.

Las salidas de los **mapas** serán únicamente en FITS, PostScript o impresas.

4 GUI

Tiene las mayores dificultades de todo el software; constará de tres partes fundamentales como se muestra en la figura 2:

- **File de observación:** Es sólo la interface para crear los files de observación. Será construida en C y con XForms como librerías gráficas, tomando como base del formato lo que ya se encuentra construido en el desarrollo del SETI.

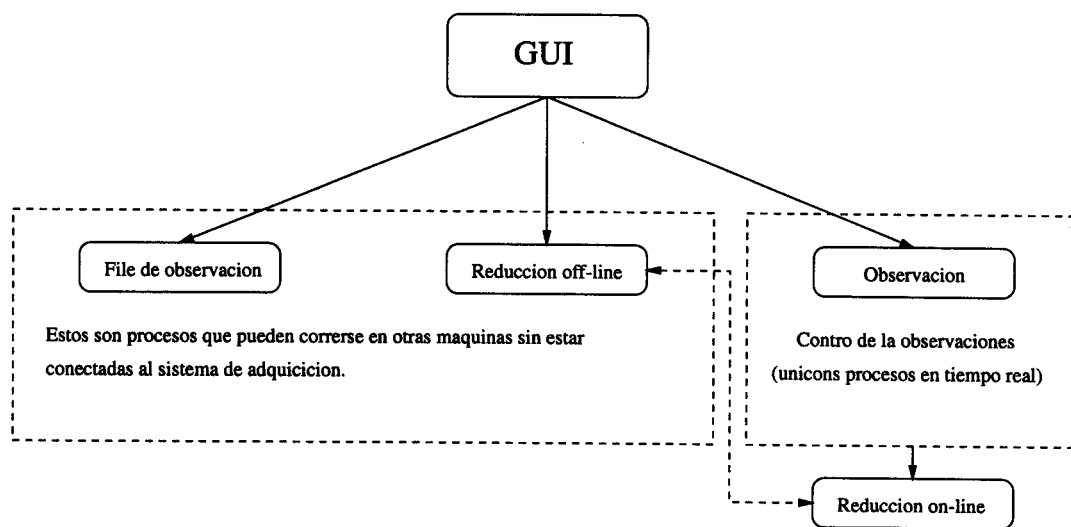


Figura 2: Esquema de la GUI

La opción más interesante para construir esta parte sería realizarlo en JAVA, con lo cual el programa pasaría a ser multiplataforma y desde cualquier lugar del mundo podría crearse un file de observación para el IAR, ya sea desde una PC o una Work Station y con cualquier sistema operativo. Esta última posibilidad no se descarta, pero implica el estudio de un *language* que aún puede sufrir grandes cambios.

- **Reducción off-on-line:** Los datos observados se reducirán off-line; pero estas facilidades estarán disponibles en el momento de la observación; por ejemplo: al hacer pointing es necesario trazar la línea de base y calcular el máximo. Aunque en este caso los tramos de la línea de base estarán previamente definidos en la base de datos de las calibradoras, la reducción se hará en forma automática y sin la intervención del astrónomo, por defecto.
- **Observación:** Consta de la adquisición de datos en sí y del control del equipo (figura 3). Es la parte del programa que tendrá comunicación con el hardware a través de las bibliotecas y los módulos.

5 Adquisición de datos

Sea lo que fuere que se quiera observar, se realiza en forma de **grilla** o **cruz**. Estas dos formas se aplicarán a cualquier situación, siendo el programa el encargado de servir de *front-end* entre el usuario y estas dos rutinas.

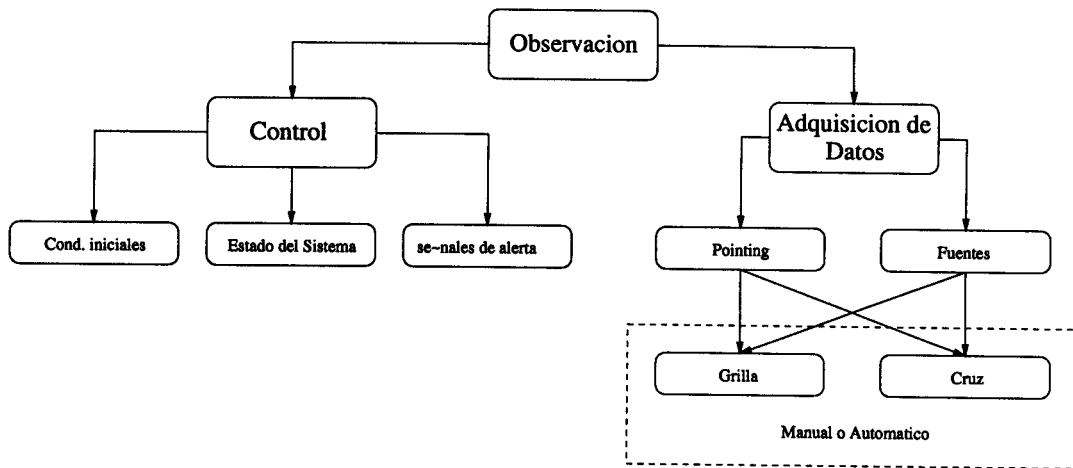


Figura 3: Esquema del sistema de observación

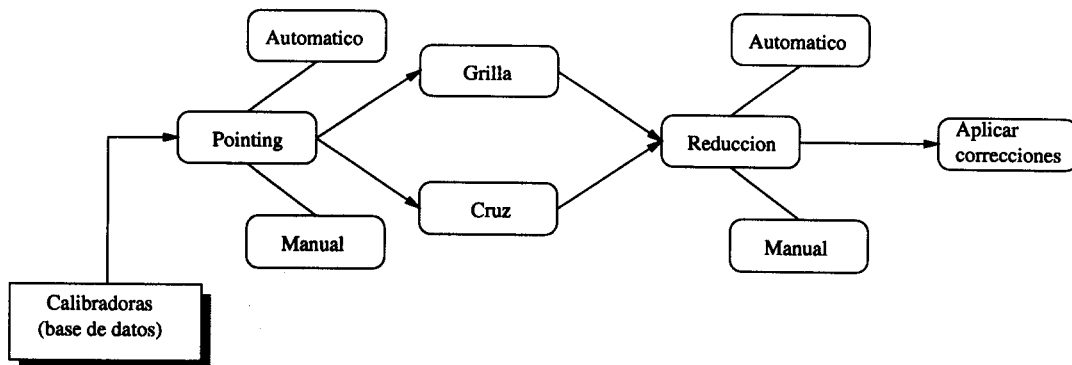


Figura 4: Esquema del pointing

5.1 Pointing

Los objetos sobre los que se hará pointing, pertenecerán a una base de datos. El objeto podrá seleccionarse durante la observación (margen mínimo 20 minutos) o podrá formar parte del file de observación. Las correcciones obtenidas se aplicarán por defecto, a menos que se especifique lo contrario (ver figura 4).

La base de datos con las calibradoras contendrá también los segmentos (o trazos) en los cuales trazar la línea de base para la reducción automática.

5.2 Fuentes

La observación de fuentes puntuales (ver sección 2.1) podrá realizarse: en cruz o grilla por medio del file de observación, y las fuentes extendidas (ver sección 2.2), con una grilla en la misma forma (figura 3).

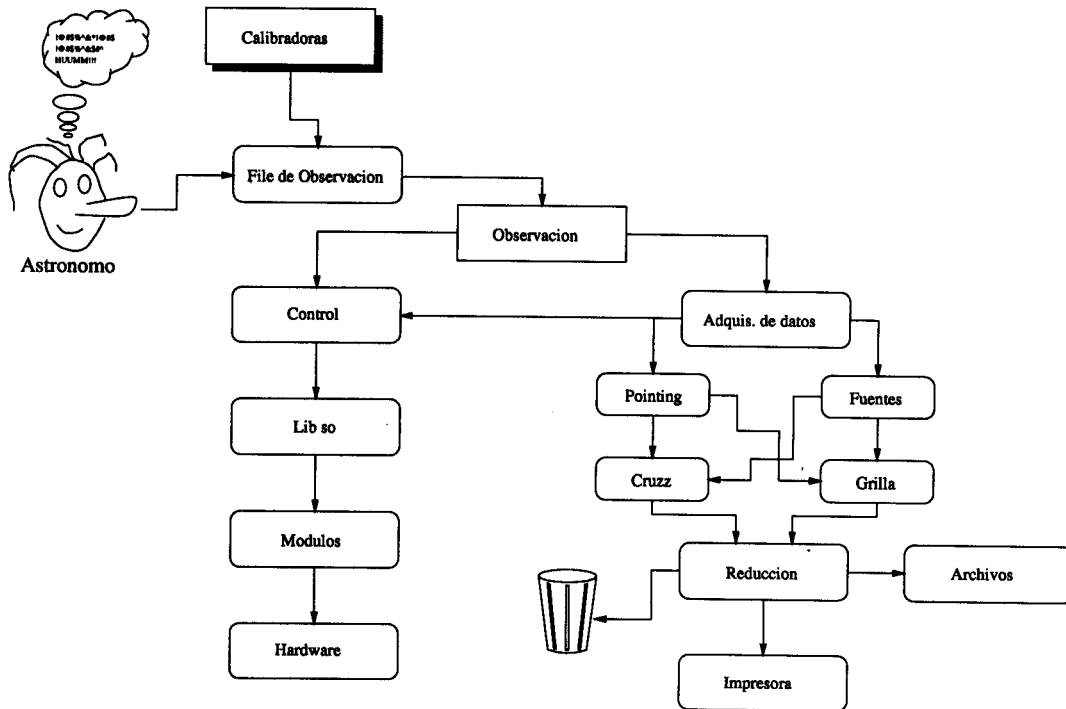


Figura 5: Esquema del Sistema