

UN “LABORATORIO VIRTUAL” PARA INSTRUMENTACIÓN EN ENERGÍAS RENOVABLES

D. Hoyos, N. Salvo

INENCO –Universidad Nacional de Salta

Centro Regional de Investigaciones Científicas y Tecnológicas – CRICYT C.C.131 C.P. 5500 – Mendoza

Tel. 0387-4255424 e-mail: hoyosd@unsa.edu.ar –nahuel@unsa.edu.ar

RESUMEN: El presente trabajo tiene por objeto presentar “un laboratorio virtual” que permite realizar toma de datos sobre distintas variables físicas, controlarlas, analizarlas y visualizarlas en un entorno de tres dimensiones, guardarlas en un archivo, procesar la información, realizar la simulación de la misma a tiempo real y también se puede ejecutar este proceso en distintas maquinas a través de una red TCP/IP.

Palabras clave: energía solar, instrumentación, control, transferencia, laboratorio.

INTRODUCCION

Cuando se desea ensayar un sistema físico determinado, por lo general se cuenta con diversas herramientas, como ser un programa de diseño, instrumentos de medida, programas que realizan la simulación del prototipo y algún post-procesador, etc. Integrar todos estos elementos, significaría contar con un “laboratorio virtual”.

Desde hace algunos años para analizar un prototipo de un sistema se utiliza como instrumento de medida computadoras con placas de toma de datos, que permiten medir distintas variables físicas. Luego sobre otra computadora se realiza el procesamiento de la información y con estos datos se realiza la simulación del sistema.

Puede ocurrir que el prototipo se encuentre a muchos kilómetros de distancia del laboratorio y este proceso de toma de datos se vuelve muy lento y poco operativo. Además algunas de las variables medidas pueden necesitar de un procesamiento adicional más complicado, lo que retarda más el proceso. En función de todos estos problemas se desarrollaron los denominados Instrumentos Virtuales, que se encargan de realizar las acciones descriptas más arriba en una forma integrada. Estos “Laboratorios Virtuales” son una herramienta muy cara actualmente, lo que imposibilita su adquisición. En este sentido en el presente trabajo se presenta una metodología que permite desarrollar estos instrumentos virtuales utilizando como sistema operativo de base el LINUX y programas existentes.

INSTRUMENTOS VIRTUALES

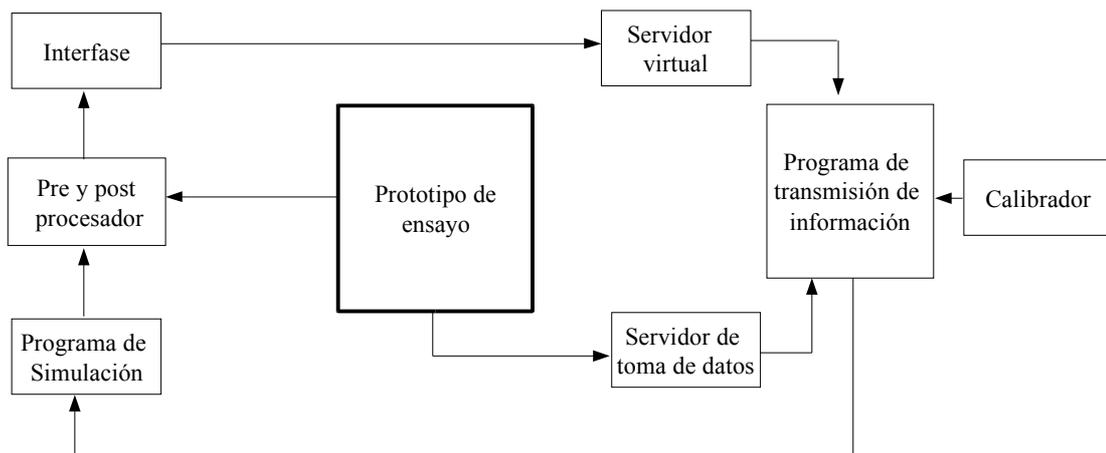


Figura 1: Esquema general del Laboratorio.

Un instrumento virtual esta compuesto básicamente de una computadora, una placa de toma de datos, los controladores de esa placa, un programa que procese esta información, la modifique, guarde y la presente en pantalla. La arquitectura básica del laboratorio será cliente/servidor o sea que existirán diversos servidores encargados de prestar servicios a los diferentes clientes. Esta estructura permite ejecutar distintos procesos del laboratorio virtual en forma paralela y también en distintas computadoras. En la figura 1 se puede observar un esquema del sistema completo.

Los diversos componentes que se muestran en la figura 1, una vez conectados permiten el ensayo y simulación del prototipo elegido. Cabe aclarar que no todos estos componentes deben estar en una misma computadora, pudiendo inclusive estar en regiones físicas muy alejadas entre si.

LOS DIFERENTES COMPONENTES

Placas de toma de datos: Las placas de toma de datos son instrumentos de medida conectados a la computadora. En general disponen de varios canales de medida de tensión (entradas analógicas), salidas de tensión (análogas), salidas digitales, entradas digitales, y medición de tiempos. Se pueden conectar a la computadora utilizando distintas tecnologías *BUS ISA*, *BUS PCI* y *puerto serie* (Tompkins et. al.).

Las placas de toma de datos requieren, en general, que se les configuren algunas características, como ser:

- Puertos de conexión con la PC
- Canales de IRQ
- Medición Simple o Diferencial
- Referencia del convertor A/D

Las mismas habitualmente se configuran sobre la placa abriendo o cerrando pequeñas llaves. La selección de los puertos de comunicación con la PC es el tema mas delicado de esta operación.

Se estudiaron las placas ADQ12, KEITHLEY –DAS1602, y PCLAB812, todas estas se conectaron alternativamente a una computadora a través del Bus ISA, teniendo acceso a ellas a través de los puertos de la PC. El esquema de las mismas se muestra en la figura 3.

Controlador de la Placa: El programa controlador se encarga de comunicar la PC con la placa de toma de datos. Cada placa necesita un programa adaptado a sus características. Estos programas deben ser realizados con rutinas de bajo nivel y como estamos trabajando en Linux (Saikkonen, 1997), por razones de seguridad, un usuario particular no puede acceder a usar los puertos sin un permiso especial. Estos programas controladores se deben programar en lenguaje C y sus salidas y entradas deben ser normalizadas de forma que los datos obtenidos puedan ser tratados sin problemas por el resto del sistema. Se construyeron tres tipos de programas:

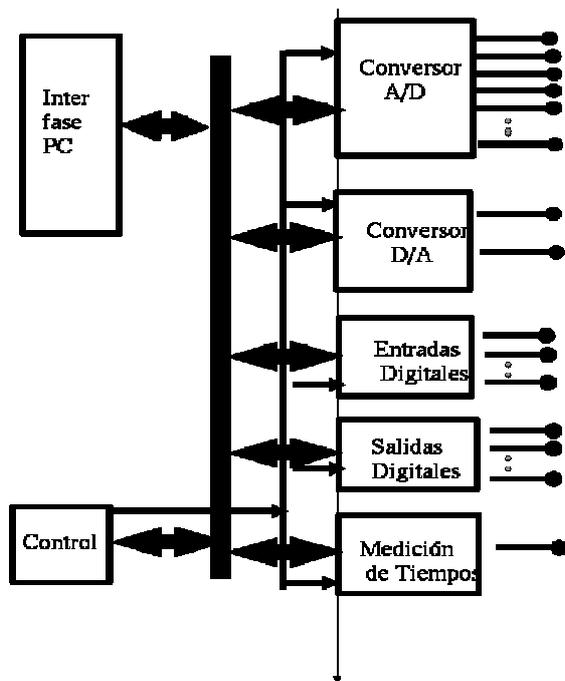


Figura 2: Esquema placa de toma de datos.

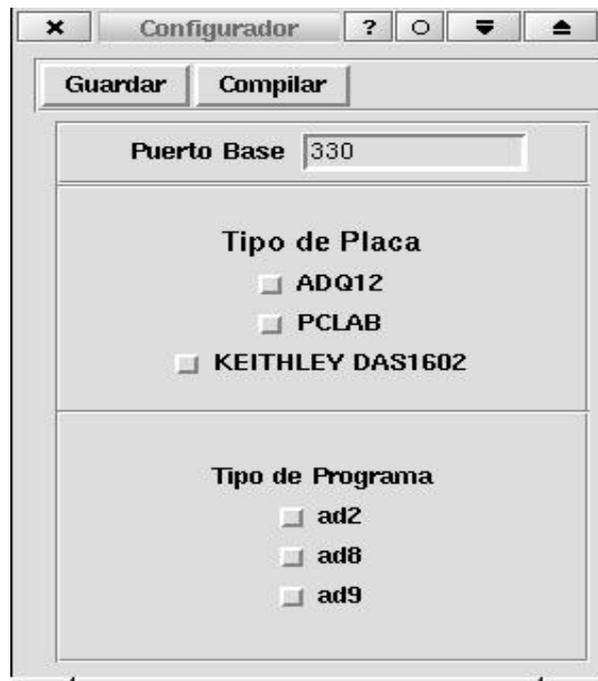


Figura 3: Creador de controladores

- El programa “ad2” toma un dato por vez, como entrada tiene el canal que se desea medir y la amplificación que recibe el mismo. Entregando un número entero en la salida standard.
- El programa “ad8” toma un conjunto de datos, uno después de otro separados entre si por un intervalo de tiempo. Como entrada tiene el canal que se desea medir, la amplificación, el número de datos a medir y el retardo de tiempo en milisegundos. Como salida devuelve al terminar de ejecutarse todos los datos adquiridos en la salida standard.
- El programa “ad9” toma un conjunto de datos sobre dos canales, separados entre si por un intervalo de tiempo. Las entradas a este programa son el primer canal que se desea medir, el segundo canal, la amplificación, el número de datos y el retardo de tiempo. Como salida devuelve al terminar de ejecutarse todos los datos adquiridos en la salida standard.

Se diseño una interfase gráfica en TCL/TK para que crea los programas nombrados anteriormente teniendo en cuenta las

características de la placa. Se debe destacar que estos programas (ad2,ad8,ad9) se los debe crear desde la cuenta root de linux, guardar en el directorio /bin, y cambiar la opción de ejecución UID.

Servidor de datos (p_socket.tcl): Como se desea utilizar este sistema para que tome datos en una computadora aislada o entre dos computadoras en una red TCP/IP se utiliza una arquitectura cliente/servidor. El servidor se encarga de esperar un pedido y el cliente realiza la toma de datos(Brent, 1997).

Para construir el servidor de datos se utilizó la tecnología de programación de *socket*. Se eligió el canal TCP/IP 2540 para que el servidor espere las ordenes que vienen del o los clientes. Estas ordenes son ejecutar “ad2”, “ad8” o “ad9” con sus entradas normalizadas. Las salidas normalizadas son enviadas a través de la red al cliente. Si no existiera red externa se utiliza la red *loopback (interna)* que esta instalada en cualquier maquina con sistema operativo *linux*. Se debe destacar que este programa que fue realizado en TCL/TK debe arrancar cuando se enciende la maquina y se ejecuta siempre. También cuando hay más de un cliente, que pueda estar accediendo a los datos a la vez. Esto nos permite compartir información.

Pre y Post procesador:En este bloque se incluye algún programa que pueda dibujar la geometría del prototipo a ensayar. En particular se utilizó a GID (Ribo, 2000), el cual es un programa al estilo de los CAD (Computer Aided Design) pero con algunas diferencias. La mas importante es que la geometría es construida teniendo en cuenta que una determinada entidad (por ejemplo, líneas), son construidas a partir de entidades de menor nivel (puntos).

GID es un programa gráfico, por lo tanto es una importante herramienta de diseño. Además de esto puede realizar en conjunto con otros programas simulaciones numéricas, tanto para problemas académicos como industriales. Este programa fue desarrollado por el Centro Internacional de Métodos Numéricos en Ingeniería (CIMNE), Barcelona España.

Con el uso de GID, se puede definir una geometría y preparar los datos provenientes de una simulación numérica para visualizar los resultados realizando un post–proceso gráfico. Este programa también permite definir en la geometría el tipo de material que esta compuesta, introduciendo parámetros, como por ejemplo su conductividad térmica o su capacidad específica.

Cuando uno utiliza a GID, como un generador gráfico de diseño, tiene además la oportunidad de realizar el mallado en elementos finitos de la misma. Para tal fin, el programa cuenta con una amplia gama de elementos, lo que proporciona una gran versatilidad de opciones. Las salidas de GID, pueden ser gráficas y ademas obtener en código ascii, los valores de las coordenadas de los nodos del mallado de elementos finitos. Estos valores de coordenadas luego pueden ser utilizados por el “entorno virtual” para generar la geometría necesaria.

Interfase: Este bloque realiza la conexión entre el dibujo generado por GID y el entorno virtual. Se desarrollo con este fin

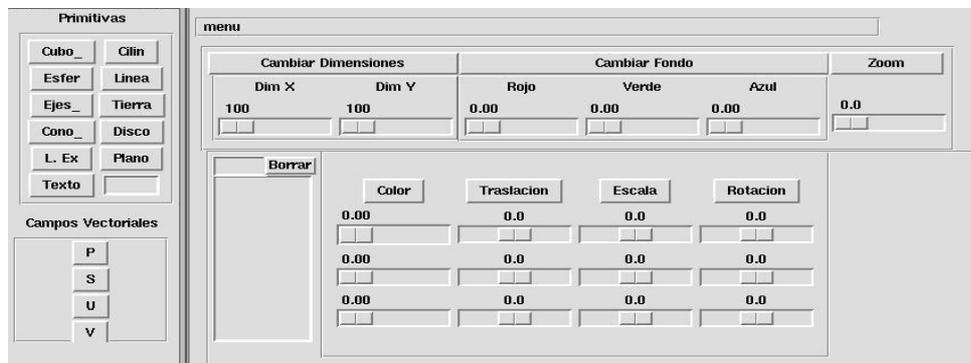


Figura 4: Interfase y generador de entorno virtual

rutinas que traducen la salida de GID, que puede ser un mallado no estructurado, al archivo de datos del lenguaje vtk de visualización en tres dimensiones.

Esta interfase además puede ser usada como editor para dibujar objetos de tres dimensiones, por ejemplo agregar termómetros, radiómetros y los correspondientes letreros a la imagen original, utilizando una combinación de primitivas que pueden ser esferas, cubos, cilindros, discos, planos etc. También empleando el ratón, el usuario puede desplazarse a través de la escena.

Servidor entorno virtual: Este programa funciona como un servidor que modifica su salida cada vez que el o los clientes le envían una orden. Puede ser utilizado como una alarma del sistema, en el sentido de que esta monitoriando en forma gráfica los valores medidos.

Una vez realizado el dibujo del escenario se debe agregar al programa un grupo de rutinas que se encargan de abrir el puerto de comunicación, esperar que alguien se comunique y modificar la imagen en función de los nuevos datos ingresados. En la figura 5 se puede apreciar una imagen del entorno virtual. (Schroeder, 1997)

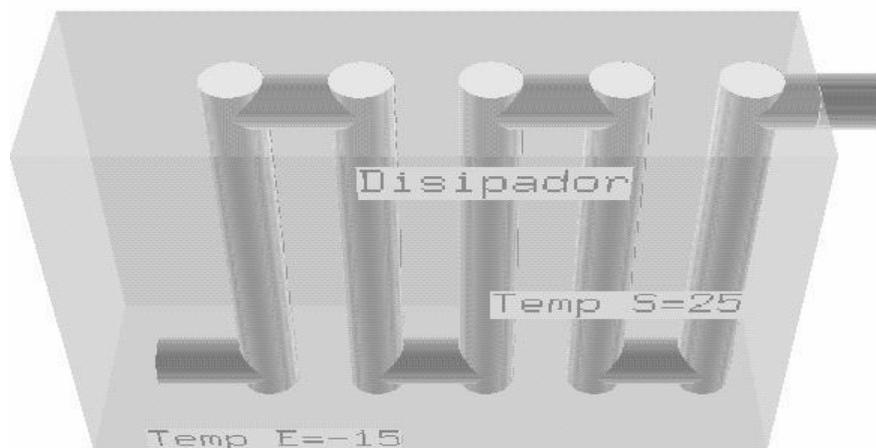


Figura 5: Vista del entorno virtual

Programas de simulación: En esta sección debe ser incluido algún programa que simule el fenómeno que se está estudiando. El bloque de simulación trabaja como cliente/servidor, pudiendo existir dentro del bloque uno o varios programas de simulación.

Se eligió para esto a PHANTOM (Flow Analysis Numerical Techniques Organized Moduli) el cual es un código en elementos finitos para realizar simulaciones numéricas en flujos de fluidos. En particular pueden simularse flujos con materiales no lineales, flujos térmicamente acoplados en regímenes turbulentos, además de considerar la posibilidad de la existencia de superficie libre que separa al fluido analizado y aire.

En lo que respecta a las técnicas numéricas que utiliza, lo más importante de destacar es que permite la posibilidad de considerar gradientes de presión continuos o discontinuos en el mallado de elementos finitos, utilizando técnicas de estabilización. La integración en el tiempo puede ser realizada por varios esquemas de diferencias finitas.

Procesador de señales: Se utiliza como procesador de señales las librerías existentes para tal fin del lenguaje Octave y de un grupo de librerías propias encargadas de calibrar los distintos sensores.

Programa de transmisión de información: Este bloque es el más importante del “laboratorio virtual”, ya que es el encargado de coordinar el flujo de la información a través del sistema. La programación del mismo debe ser realizada para cada experiencia que se desee ensayar, por lo tanto se busca un lenguaje simple y muy poderoso que permita cumplir con este objetivo. Octave cumple con esta característica y agregándole al mismo librerías encargadas de comunicarse con los diferentes servidores se puede coordinar los distintos procesos.

Para cada servidor se utiliza un programa cliente encargado de transferir la información de un bloque a otro. Estos programas clientes se realizaron en TCL/TK. La información se transmite a través de una “pipeline” (tubería) desde el programa cliente hasta la librería de Octave. (Marquéz, 1993)

En el caso de pretender realizar una experiencia en la que se mida una variable física como la temperatura de un determinado prototipo, se debería realizar un programa que corra en Octave que llame al servidor de toma de datos a través de su rutina correspondiente. El dato obtenido es un número entero, como ya se explicó anteriormente, debe ser transformado según una determinada ecuación para poder ser presentado en el “servidor de entorno virtual”.

En el caso de querer simular el funcionamiento del sistema, los datos obtenidos por el servidor de datos, deben ser enviados por el programa de transmisión de la información al servidor de simulación correspondiente. La respuesta de este servidor puede ser enviada al “servidor de entorno virtual”. De esta forma se cuenta con datos provenientes de la simulación y los datos medidos en un elegante medio de visualización en tres dimensiones.

CONCLUSIONES

Una primera conclusión de este trabajo es que el sistema desarrollado permite utilizar tres programas integrados en un mismo ambiente, para diseñar instrumentos virtuales. Por lo tanto se cuenta con una herramienta de trabajo muy flexible.

Se debe destacar que tenemos al menos tres servidores esperando para recibir o enviar información. Estos servidores no necesariamente deben estar en la misma computadora, pudiendo estar en distintas máquinas, separadas entre sí una distancia muy grande siempre y cuando estén conectadas a INTERNET. Este último aspecto se torna importante cuando el sistema de toma de datos se encuentra localizado físicamente en otra localidad. También el sistema permite utilizar varios servidores de datos, simulación y entorno virtual, lo que hace esta metodología de trabajo muy flexible a la hora de diseñar una determinada experiencia.

Esta en desarrollo diseñar los controladores de salidas analógicas y digitales para los instrumentos virtuales posibilitando el control de sistemas a distancia. También se pretende diseñar controladores para otras placas de toma de datos como por

ejemplo el NUDAM.

Se puede adaptar al sistema sin perder performance para utilizar solo una parte, por ejemplo medir una variable física sin simularla.

REFERENCIAS

Saikkonen Riku (1997) Linux I/O port programming mini HOWTO. INTERNET.
SchroederWill, Ken Martin, Bill Lorensen. (1997). The Visualization Toolkit, Prentice Hall.
Ramon Ribo (2000). GID Reference Manual CIMNE
Brent B. Welch Practical Programming in TCL and TK (1997) 2. Edición Prentice Hall.
Márquez García, Francisco Daniel (1993) Unix Programación Avanzada. Ra-ma.
Tompkins, W- Webster, J. Interfacing Sensors to the IBM -PC. Edición Prentice Hall.

ABSTRACT

The present paper propose a virtual laboratory who can get adquisition data from physical variables 3D visualization digital signal proccesing and simulation. This laboratory can work in a TCP/IP network.