

Laboratorio Virtual Remoto para Educación en Control: Emulación de una Planta Real de Control de Niveles

Loyarte, Ariel Sebastián – Blas, María Julia – Vega, Jorge Ruben

*Laboratorio de Sistemas de Control (LSC)
Facultad Regional Santa Fe, Universidad Tecnológica Nacional (FRSF-UTN)*

Abstract

Este trabajo describe el desarrollo de un Laboratorio Virtual Remoto (LVR), concebido como un sistema Cliente / Servidor apto para emular la operación controlada de una planta de tanques interconectados, disponible en el Laboratorio de Sistemas de Control de la FRSF-UTN. El LVR le permite a un alumno proponer y evaluar estrategias de control (a lazo abierto y cerrado), y sintonizar un controlador, previo a su implementación en la planta real. El desarrollo del LVR involucró 2 etapas independientes: 1) el modelado matemático de la planta, y 2) el desarrollo de la plataforma de soporte a los LV. Esta plataforma se implementó con tecnología Java. Su arquitectura posibilita el acceso concurrente de múltiples usuarios. El Servidor posee un conjunto de módulos destinados a interactuar con el simulador de la planta, implementado en Matlab/Simulink; y es lo suficientemente versátil como para admitir en un futuro la inclusión de otras plantas. El Cliente consiste en una interfaz gráfica de manejo sencillo para el usuario, la cual facilita la configuración de los modelos de simulación. En todas las etapas de diseño e implementación del LVR participaron alumnos de las carreras de Ingeniería Eléctrica e Ingeniería en Sistemas de Información de la FRSF-UTN, ámbito natural de la explotación futura del producto desarrollado.

Palabras Clave

Laboratorio Virtual – Educación en Control – Modelado Matemático – Cliente/Servidor – Matlab/Simulink – Java

Introducción

En las carreras universitarias de Ingeniería, es indudable que los trabajos prácticos de laboratorio cumplen un rol preponderante en la formación experimental de los alumnos [1]. En el caso particular del departamento de Ingeniería en Sistemas de Información (ISI) de la FRSF-UTN, no se dispone de un laboratorio de control propio; pero los alumnos de la asignatura Teoría de

Control (del cuarto nivel de la carrera), tienen la posibilidad de desarrollar una actividad práctica en el Laboratorio de Sistemas de Control (LSC) del departamento de Ingeniería Eléctrica. El espacio reducido y la escasa disponibilidad horaria del LSC, la extensión de un práctico de laboratorio típico, y el elevado número de alumnos por curso, restringen la posibilidad de implementar prácticos con una profundidad acorde a lo esperado.

A efectos de mitigar el problema arriba descrito, y en el marco de un proyecto de investigación [2], se planteó el desarrollo de un Laboratorio Virtual (LV). De acuerdo a lo establecido en una reunión de expertos en laboratorios virtuales [3], un LV se puede definir como “*un espacio electrónico de trabajo concebido para la colaboración y la experimentación a distancia con objeto de investigar o realizar otras actividades creativas, y elaborar y difundir resultados mediante tecnologías difundidas de información y comunicación*”. Si bien se asume que todo LV admite un acceso remoto por parte de múltiples usuarios (ya sea en forma secuencial o concurrente), aún así continúa siendo usual utilizar la terminología LV remoto (LVR).

El LVR que se presenta en este trabajo fue planeado con un objetivo amplio, y que postuló otorgar a los alumnos y docentes de diferentes asignaturas y carreras de la FRSF la oportunidad de generar y/o explotar simuladores de plantas de interés integrados al LVR [2]. El usuario podrá tener acceso al LVR tanto desde de la red local de la facultad como por Internet.

En un proceso educativo, un LV aporta reconocidas ventajas, tales como permitir a los alumnos ‘experimentar’ sobre una planta, analizar su funcionamiento, y configurar y ajustar un lazo de control con antelación a su implementación definitiva en la planta real. Además, un LV permite reducir los tiempos de experimentación, eliminar los riesgos por uso indebido de equipos reales, y minimizar los costos por ensayo. Debe tenerse en cuenta que un LVR no pretende reemplazar a un laboratorio real, sino complementarlo, haciendo que el alumno se sienta más preparado y con mayor conocimiento sobre la física del problema, a la hora de abordar el trabajo práctico de laboratorio sobre la planta real. Varios LV se han presentado en la literatura (por ej., [4,5]). Existen diversos entornos que facilitan el desarrollo de LVR, como por ejemplo Easy Java Simulations (EJS) en combinación con Java Internet Matlab (JIM) [5-7]. Sin embargo, estas herramientas sólo permiten definir un único LV sobre un modelo y entorno de simulación específico; y por lo tanto no cubren las expectativas del aplicativo buscado.

Este trabajo resume las etapas desarrolladas en el marco del proyecto [2], para disponer un LVR apto para simular una planta real de control de niveles disponible en el LSC de la FRSF-UTN. La plataforma del LVR es general, en el sentido que fue diseñada para incorporar en el futuro otras plantas de experimentación. En las secciones siguientes se describe la planta real, su modelo matemático y el simulador, y se comenta la plataforma desarrollada que permite implantar múltiples LV.

La Planta Real: Descripción y Modelado

La Fig. 1 muestra una fotografía de la parte superior de la planta de control de niveles existente en el LSC; y la Fig. 2 esquematiza los componentes principales del subsistema de flujo: 3 tanques interconectados (TK1, TK2, TK3), un tanque de reserva (TK4), y una bomba centrífuga para la alimentación de agua. La planta está instrumentada con

sensores de nivel en los 4 tanques, y posee un subsistema electrónico de control encargado de conectar el instrumental de la planta con una PC remota que aloja la interfaz gráfica de comando y los algoritmos de control. El caudal de agua se regula con un variador de frecuencia (V.F.) que se utiliza para modificar la velocidad del motor de accionamiento de la bomba.



Figura 1. Fotografía de la parte superior de la planta real, ubicada en el LSC (FRSF-UTN). Se observan los 3 tanques interconectados, y la parte frontal del sistema electrónico de control. (La altura total de cada tanque es de 100 cm.)

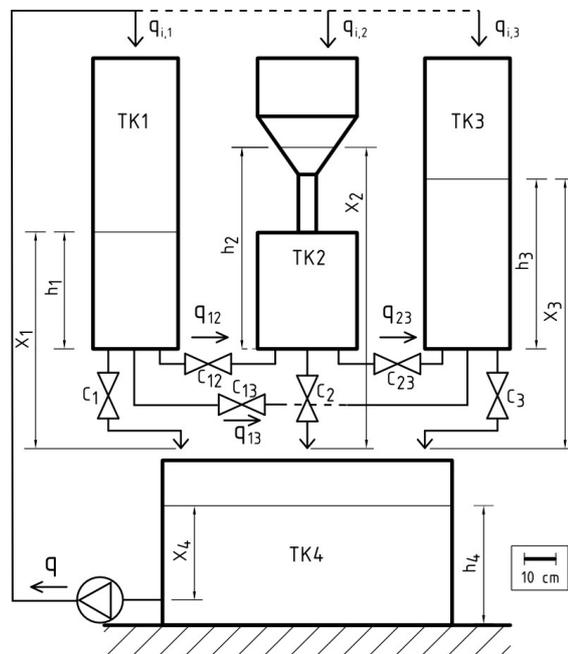


Figura 2. Esquema de la planta real: tanques, restricciones y subsistema de alimentación de agua. La planta es configurable con la apertura o cierre de las válvulas, y la selección del tanque a alimentar.

La planta se utiliza para realizar prácticas de laboratorio en asignaturas de control e instrumentación. El usuario, a través de una interfaz gráfica desarrollada en LabVIEW (*National Instruments*), opera el sistema desde una PC conectada (en forma inalámbrica) con un controlador lógico programable (PLC). El PLC recibe las señales de medición de niveles y las transmite a la PC. A través de un algoritmo de control, la PC calcula el nuevo valor requerido del caudal, lo transforma en un valor equivalente de frecuencia, y envía una señal al PLC para que actúe sobre el V.F. y consecuentemente modifique el caudal de alimentación de agua. La planta puede ser operada tanto en lazo abierto como cerrado. Los tanques TK1, TK2 y TK3 están interconectados por sus bases, y cada uno puede descargar hacia TK4. Todas las válvulas de salida pueden ser abiertas o cerradas, siendo posible entonces definir diferentes configuraciones operativas de la planta. Si bien hoy sólo es posible alimentar agua a un tanque, se están actualmente instalando válvulas digitales de 8 y 10 bits, que permitirán regular el caudal de alimentación a los 3 tanques. La modalidad de regulación del caudal será entonces por apertura y cierre de bits en cada válvula, en lugar de recurrir a la variación de frecuencia que se utiliza actualmente. El tanque TK2 tiene la particularidad de su sección transversal variable con la altura. Este tanque cuenta con 2 zonas cilíndricas de áreas diferentes, y con una zona cónica de acoplamiento. Las dificultades de control de nivel en el TK2 ya han sido exploradas mediante una estrategia de control basadas en agentes inteligentes [8]. El caso de mayor complejidad a resolver es la transición suave del nivel de agua en la zona donde la sección transversal cambia bruscamente (es decir, la zona de conexión de los sectores cilíndricos de secciones transversales diferentes).

Modelo matemático del sistema de tanques

Inicialmente se consideró el modelado de los 3 tanques interconectados. Un modelo

matemático de primeros principios se obtuvo a partir de las ecuaciones de balance de materia. El planteo detallado y su desarrollo fueron expuestos en un trabajo previo [9], aunque inicialmente TK3 no fue incluido. Extendiendo los resultados al caso de 3 tanques interconectados, se deriva el siguiente modelo matemático:

$$A_1 \times \frac{dX_1}{dt} = q_{i,1} - C_1 \times \sqrt{X_1} + C_{12} \times \text{Sgn}(X_1 - X_2 + 3,3[\text{cm}]) \times \sqrt{|X_1 - X_2 + 3,3[\text{cm}]|} + C_{13} \times \text{Sgn}(X_1 - X_3) \times \sqrt{|X_1 - X_3|} \quad (1)$$

$$A_{2i} \times \frac{dX_2}{dt} = q_{i,2} - C_2 \times \sqrt{X_2} + C_{23} \times \text{Sgn}(X_2 - X_3 + 3,3[\text{cm}]) \times \sqrt{|X_2 - X_3 + 3,3[\text{cm}]|} + C_{12} \times \text{Sgn}(X_1 - X_2 + 3,3[\text{cm}]) \times \sqrt{|X_1 - X_2 + 3,3[\text{cm}]|} \quad (2)$$

$$A_3 \times \frac{dX_3}{dt} = q_{i,3} - C_3 \times \sqrt{X_3} + C_{23} \times \text{Sgn}(X_2 - X_3 + 3,3[\text{cm}]) \times \sqrt{|X_2 - X_3 + 3,3[\text{cm}]|} + C_{13} \times \text{Sgn}(X_1 - X_3) \times \sqrt{|X_1 - X_3|} \quad (3)$$

con las siguientes equivalencias:

$$h_1 = X_1 - D_1 \quad (4)$$

$$h_2 = X_2 - D_2 \quad (5)$$

$$h_3 = X_3 - D_3 \quad (6)$$

donde $q_{i,j}$ es el caudal de entrada; A_j es el área transversal; X_j es el nivel medido desde la superficie libre del líquido hasta el punto de descarga (a presión atmosférica); h_j es el nivel medido desde la superficie libre del líquido hasta la base del tanque; D_j es la distancia desde la base del tanque hasta su punto de descarga; C_j es el parámetro de descarga; y C_{jk} es el parámetro de interconexión. El término "Sgn" representa la función signo. Los subíndices j, k ($= 1, 2, 3$), hacen referencia al tanque j, k .

En definitiva, el modelo matemático queda representado por 3 ecuaciones diferenciales ordinarias (EDO), no lineales y acopladas. La Ec. (2) presenta la dificultad adicional

de tener el parámetro A_{2i} variable con el nivel de agua (indicado por el subíndice i), debido a la no-uniformidad de la sección transversal del TK2.

Modelo del subsistema de alimentación

El subsistema que define el caudal de alimentación de agua está conformado por tres componentes: la bomba centrífuga, el variador de frecuencia, y el tanque de reserva TK4. La frecuencia permite variar proporcionalmente la velocidad síncrona del campo magnético giratorio auto-inducido del motor de accionamiento de la bomba. Sin embargo, por tratarse de un motor asincrónico, la velocidad resulta levemente inferior a la obtenida por esa proporcionalidad. Además, se comprobó que la presión en la admisión de la bomba (que es proporcional al nivel de agua en el TK4) influye sobre el caudal resultante, principalmente para frecuencias reducidas. En definitiva, se vio conveniente efectuar una “calibración” del subsistema mediante ensayos diseñados para determinar la relación real existente entre la frecuencia de alimentación y el caudal efectivo. Estos ensayos consistieron básicamente en medir el caudal impulsado por la bomba cuando se modifican tanto la frecuencia del variador como el nivel del TK4. La Fig. 3 muestra la “calibración” obtenida [10].

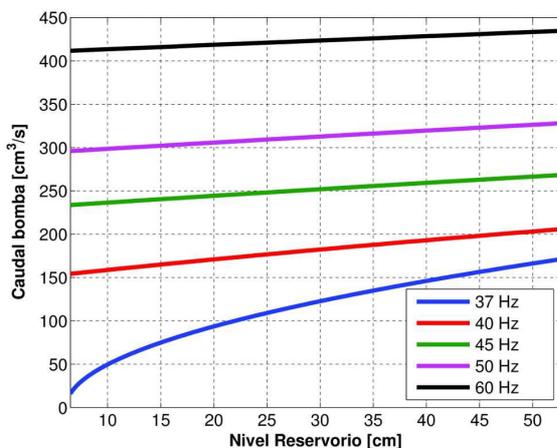


Figura 3. Calibración del caudal de la bomba en función del nivel de agua en el tanque de reserva (TK4), para 5 frecuencias del variador.

Simulador de la planta a lazo abierto

El modelo de la planta (a lazo abierto) se obtiene combinando los modelos de los subsistemas de tanques y de alimentación. Ambos subsistemas se implementaron en el ambiente *Matlab*, utilizando la interfaz gráfica *Simulink*. La Fig. 4 muestra un esquema del simulador, para la planta funcionando en lazo abierto. Por simplicidad, sólo se muestran los bloques de mayor jerarquía. Los bloques indicados como “TK1”, “TK2” y “TK3” modelan las Ecs. (1), (2) y (3), respectivamente, apoyados por los bloques de cálculo de los caudales de interacción (“q12”, “q23” y “q13”). El bloque “Lookup Table” modela la calibración del sistema de alimentación de agua, y calcula el caudal (q) por medio de una tabla de doble entrada, que utiliza la frecuencia del variador (f) y el nivel de TK4 (N), para interpolar linealmente las curvas de la Fig. 3.

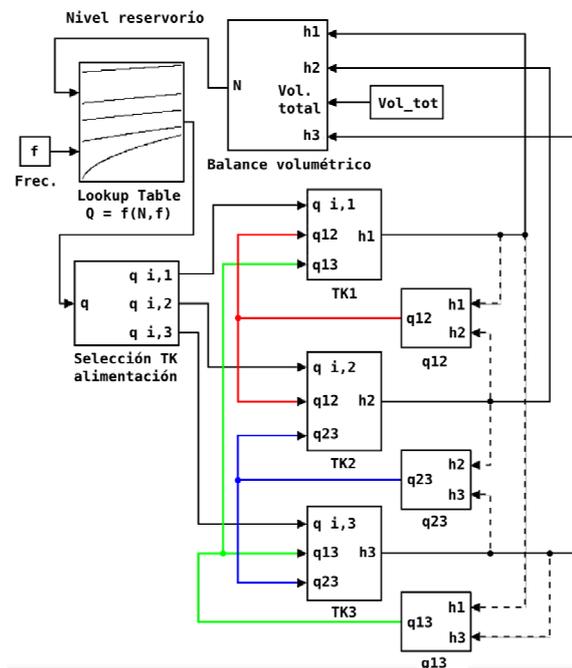


Figura 4. Esquema en bloques del simulador de la planta operando a lazo abierto. Los datos externos son dos: la frecuencia (f) del variador y el volumen total de agua disponible en el sistema (Vol_tot).

Las frecuencia, f , posee valores operativos máximo y mínimo, que limitan el rango de caudales permitidos. El bloque “Selección

TK alimentación” permite elegir el tanque al cual se alimenta agua. Por último, el bloque de “Balance volumétrico” se utiliza para calcular el nivel de TK4, en función del volumen total de agua (“Vol_tot”) disponible en el sistema.

Debe observarse que los parámetros asociados a características geométricas (A_j y D_j) pueden medirse directamente; mientras que las resistencias hidráulicas deben estimarse a partir de alguna metodología específica, como la desarrollada en [10, 11]. En esos trabajos, se utilizó un algoritmo evolutivo de optimización por enjambre de partículas (PSO), para estimar los parámetros asociados a las válvulas de descarga y de interconexión. El algoritmo basado en PSO utiliza el simulador de la Fig. 4, y ajusta en forma iterativa los parámetros del modelo hasta lograr que se reproduzcan los niveles de agua medidos en los experimentos. La Tabla 1 resume los parámetros de la planta.

Tabla 1. Parámetros del modelo matemático del subsistema de tanques [Ecs. (1-3)].

Parámetros medidos	
$D_1 = 18,0 \text{ cm}$	$A_1 = 660,52 \text{ cm}^2$
$D_2 = 21,3 \text{ cm}$	$A_2 = (*)$
$D_3 = 18,0 \text{ cm}$	$A_3 = 660,52 \text{ cm}^2$
Parámetros estimados	
$C_1 = 24,5 \text{ cm}^{2,5}/\text{s}$	$C_{12} = 49,9 \text{ cm}^{2,5}/\text{s}$
$C_2 = 15,5 \text{ cm}^{2,5}/\text{s}$	$C_{23} = 50,0 \text{ cm}^{2,5}/\text{s}$
$C_3 = 20,0 \text{ cm}^{2,5}/\text{s}$	$C_{13} = 50,0 \text{ cm}^{2,5}/\text{s}$

(*) Variable según la altura vertical (ver Ref. [9]).

Cabe destacar que el modelo definitivo implementado en el LVR posee los parámetros de la Tabla 1 como valores recomendados. Sin embargo, se consideró interesante brindar al usuario la posibilidad de alterarlos, permitiendo realizar análisis más generales del sistema. Por caso, a lo largo del tiempo, es probable que los parámetros vayan alterándose como consecuencia del uso, del desgaste y del ensuciamiento progresivo de los conductos. Como caso límite, un valor nulo asignado a alguno de esos parámetros anula el caudal

que fluye por el correspondiente conducto, simulando así el cierre de la válvula.

Simulador de la planta a lazo cerrado

El modelo descrito se complementa con mediciones de los niveles de agua en los tanques, un controlador para determinar el caudal de alimentación, y la asignación de una entrada de consigna (o “set-point”). De esta manera, se cierra un lazo de control de variable única como el esquematizado en la Fig. 5, donde cada bloque requiere de una parametrización específica.

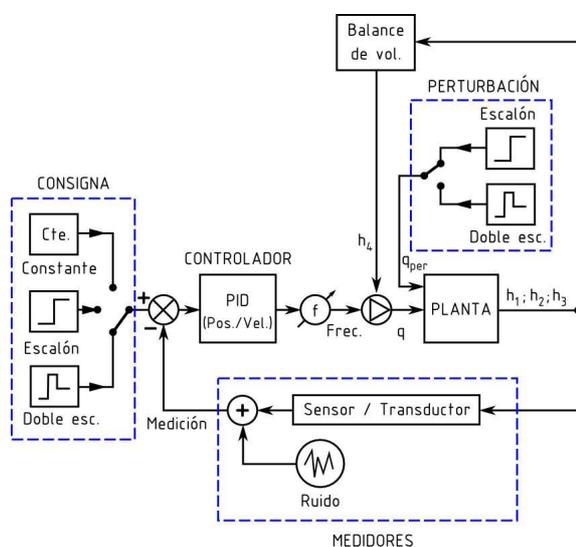


Figura 5. Esquema en bloques de la planta operando a lazo cerrado. El bloque denominado “PLANTA” incluye una modificación menor del esquema representado en la Fig. 4.

El bloque controlador utiliza un algoritmo PID, siendo posible seleccionar la forma de implementación del algoritmo (*posicional* o de *velocidad*), y también sintonizar sus parámetros.

El bloque de consigna puede ser elegido como una “constante”, un “escalón” retrasado en el tiempo, o un “doble escalón”. En cualquier caso, se posibilita la elección de los valores de la magnitud y de los tiempos asociados a los cambios.

El bloque del medidor se modeló en base a las especificaciones técnicas características de los sensores de nivel por presión disponibles en la planta. Estos sensores

operan linealmente en el rango 0-0,1 bar, emitiendo una señal eléctrica de corriente en el rango estandarizado de 4-20 mA. Para los niveles de cada tanque, la simulación permite calcular las corrientes emitidas por cada sensor, tal como resulta durante la operación del sistema real. Adicionalmente, se incluyó la posibilidad de simular ruido de medición aditivo, modelado a partir de una distribución normal (Gaussiana) con valores seleccionables tanto de la media como de la desviación estándar.

Para brindar una mayor versatilidad al modelo, se incorporó un bloque de perturbación, que posibilita la adición de un caudal de agua a un tanque dado. La configuración del bloque consiste en la elección tanto del tanque a perturbar como del tipo de perturbación a simular (una función escalón o un doble escalón). El usuario puede optar por incorporar una de esas funciones o por simular el modelo libre de perturbaciones.

Cabe destacar que en el simulador final se incluyó un algoritmo capaz de detectar posibles inconsistencias en la configuración del LVR establecida por el usuario, y emitir los alertas que posibiliten su corrección. Por ejemplo, el algoritmo actúa cuando a un parámetro se le asigna un valor ubicado fuera de su rango operativo. Asimismo, el algoritmo actúa con señales de alarma y/o detención de la simulación, ante el rebalse de un tanque o cuando se alcanza la condición de funcionamiento en vacío de la bomba, debido a un nivel bajo en el TK4. En cualquiera de los casos, se almacena en una variable interna el código de falla correspondiente. Posteriormente, en la implementación de la aplicación Cliente del LVR (ver la sección siguiente del trabajo), esta variable permitirá notificar al usuario de tales contingencias.

El Laboratorio Virtual Remoto

Se desarrolló una plataforma de software que permite a un usuario interactuar con diferentes simuladores de plantas. La operación del sistema requiere de una conexión a Internet, de la instalación de una

aplicación Cliente en la máquina local, y de la instalación de un aplicativo Servidor en una máquina dedicada. Cabe aclarar que una plataforma es un sistema informático que sirve como base para hacer funcionar determinados módulos de hardware o de software, y que debe facilitar la manipulación de modelos desarrollados en diferentes herramientas, posibilitando su configuración en un entorno local y su ejecución en un entorno remoto.

Arquitectura y Tecnología de la Plataforma

La arquitectura de la plataforma se definió en base a un esquema Cliente/Servidor, el cual se enmarca en el estilo arquitectónico de procesos comunicantes [12,13]. Aunque idealmente ambos componentes deben tener una carga de procesamiento similar, dadas las características de los LVR se optó por trabajar con cargas asimétricas. El Servidor se encarga de gestionar los modelos de simulación y su ejecución, mientras que el Cliente se responsabiliza de gestionar la configuración y los resultados obtenidos. Esta elección aporta numerosas ventajas. No sólo brinda escalabilidad, centralización del control y facilidad de mantenimiento, sino que además contribuye a reducir los requerimientos computacionales de la máquina Cliente, ya que las herramientas encargadas de la simulación deben estar instaladas únicamente en la máquina Servidora. Así, el entorno en el cual se ejecuta el modelo resulta transparente al usuario. La Fig. 6 muestra la arquitectura diseñada en términos de módulos y mecanismos de comunicación, donde se especifican en particular los módulos relativos a la planta de control de niveles. Estos últimos son los encargados de asignar los parámetros al modelo Simulink, ejecutar la simulación en Matlab, y empaquetar el resultado de la misma en un formato entendible por la plataforma.

El nivel de generalización obtenido con este esquema facilita la interacción con múltiples simuladores. Como puede observarse para el caso del laboratorio

asociado al sistema de control de tanques, cada LVR particular lleva un módulo dedicado que gestiona su accionar. De esta manera, la incorporación de un nuevo laboratorio a la plataforma conlleva únicamente una modificación en el aplicativo servidor.

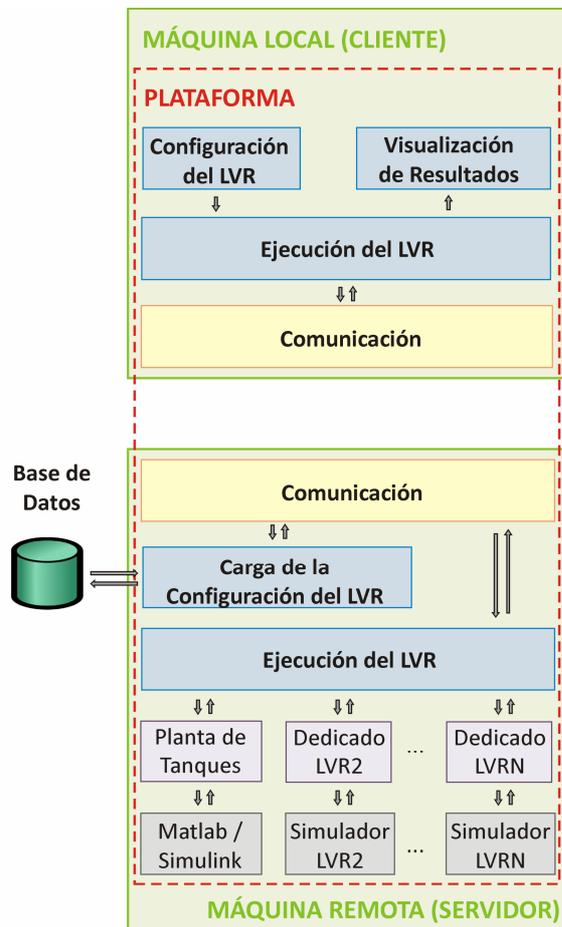


Figura 6. Arquitectura del LVR. El diseño permite ejecutar los simuladores de varias plantas, donde la primera implantada fue la del sistema de tanques (módulos de la parte inferior izquierda).

Un análisis de las características de los lenguajes de programación más populares en relación a los requerimientos de la plataforma, llevó a la elección de la tecnología Java como herramienta de desarrollo. Los lenguajes evaluados fueron Java, C++ y C#. La elección de Java se fundamentó en el hecho de que permite la interacción con múltiples entornos de simulación por medio de librerías especiales. Esta característica le provee una buena ventaja, puesto que la plataforma

requiere de un fuerte intercambio de datos e información con este tipo de componentes. Además, Java brinda un alto nivel de portabilidad, lo que fomenta la usabilidad y contribuye a la realización de un desarrollo independiente del entorno de ejecución. Teniendo en cuenta que el conjunto de datos a resguardar en la plataforma no es significativo ni complejo, puesto que sólo referencia a las características generales de los modelos de simulación, se decidió utilizar MySQL como gestor de base de datos [14,15].

Funcionamiento de un LVR genérico sobre la Plataforma

La Tabla 2 esquematiza el intercambio de información entre los entornos planteados como parte de la plataforma, junto con un detalle de los pasos llevados a cabo para ejecutar un LVR particular. De esta forma se evidencia la interacción existente entre la máquina que contiene la aplicación Cliente y la máquina que contiene la aplicación Servidora.

Tabla 2: Secuencia de tareas Cliente y Servidor durante una ejecución de un LVR.

MÁQUINA	TAREA
Cliente	1. Consultar modelos disponibles
Servidor	2. Buscar modelos disponibles
Servidor	3. Devolver lista de modelos: {Mod. 1, Mod. 2, ..., Mod. N}
Cliente	4. Presentar modelos disponibles
Cliente	5. Configurar modelo específico
Cliente	6. Solicitar chequeo de configuración elegida
Servidor	7. Validar configuración elegida
Servidor	8. Configuración elegida OK
Cliente	9. Solicitar ejecución de configuración elegida
Servidor	10. Ejecutar configuración elegida
Servidor	11. Devolver resultados
Cliente	12. Mostrar resultados

Implementación

El modelo del sistema de control de niveles detallado en las secciones precedentes se

incorporó a la plataforma como un LVR particular, sirviendo como primera experiencia en la incorporación de plantas de experimentación reales sobre la herramienta. En base a la arquitectura previamente definida, se implementaron en Java los módulos de las aplicaciones Cliente y Servidor. Se detallan a continuación los módulos principales para el funcionamiento de la plataforma.

El módulo central de la máquina Cliente es el de “*Configuración de Modelos/LVR*”. Este módulo se definió en términos de un conjunto de clases genéricas que representan los componentes de una configuración válida. Esas clases permiten al aplicativo Cliente instanciar una configuración específica y posteriormente determinar la validez del conjunto de parámetros seleccionados por el usuario. En el caso del sistema de control de niveles, este módulo visualiza los componentes disponibles en la plataforma para la configuración del laboratorio, brindándole al usuario la posibilidad de definir:

- Tipo de señal de referencia.
- Tipo de actuador.
- Tipo de controlador.
- Tipo de medidor.
- Existencia y tipo de perturbación.

Mediante la utilización de una estrategia “*drag-and-drop*”, el usuario configura el esquema a lazo cerrado según la experimentación que desee realizar, seleccionando el conjunto de componentes que mejor se adapte a sus necesidades. Luego, fija los valores de los parámetros asociados a cada componente. La Fig. 7 muestra la interfaz con la cual se lleva a cabo este proceso. Por ejemplo, para el caso del sistema de tanques, la señal de referencia puede elegirse entre una constante, un escalón o un doble escalón; el tipo de actuador corresponde a la selección del variador de frecuencia; el tipo de controlador es un PID seleccionable entre las formas de posición o de velocidad; el

tipo de medidor puede ser ideal (instantáneo y exacto), o puede elegirse la opción de agregarse un ruido de medición aditivo; y por último puede optarse por agregar (o no) una perturbación de caudal, que puede ser modelada con un escalón simple o con un doble escalón.

El módulo “*Ejecución de LVR*” de la máquina Servidora persigue el objetivo de atender las solicitudes de simulación, para lo cual crea un hilo de ejecución por cada LVR a ejecutar. Dicho hilo es el encargado de invocar al módulo responsable de la simulación, habiéndose definido su accionar en función del contenido de la instancia de la clase *Configuracion.java* proveniente del aplicativo Cliente. Teniendo en cuenta que dicha instancia contiene la información de los componentes y parámetros del LVR a ejecutar, se redirecciona esa información al módulo dedicado que manipula el modelo de simulación correspondiente. Además, existe una clase abstracta denominada *ModuloSimulador.java* de la cual deben heredar los módulos dedicados para garantizar consistencia con la plataforma.

La estructura de hilos planteada permite atender múltiples pedidos de forma concurrente. Si la herramienta de simulación permite trabajar con ejecuciones en paralelo, podrán invocarse múltiples motores desde diferentes hilos. Esto permite atender solicitudes de distintos clientes en un mismo momento. En caso contrario, el módulo dedicado será quien administre (por medio de la utilización de una cola u otro tipo de estructuras de datos) las solicitudes pendientes, invocando al motor de ejecución según el turno de cada petición.

Finalizada la ejecución de la simulación, la aplicación Servidora devuelve (a la aplicación Cliente) en la instancia de *Configuracion.java* los resultados de cada una de las variables medidas durante la simulación (conjunto de instancias de *Resultado.java*).

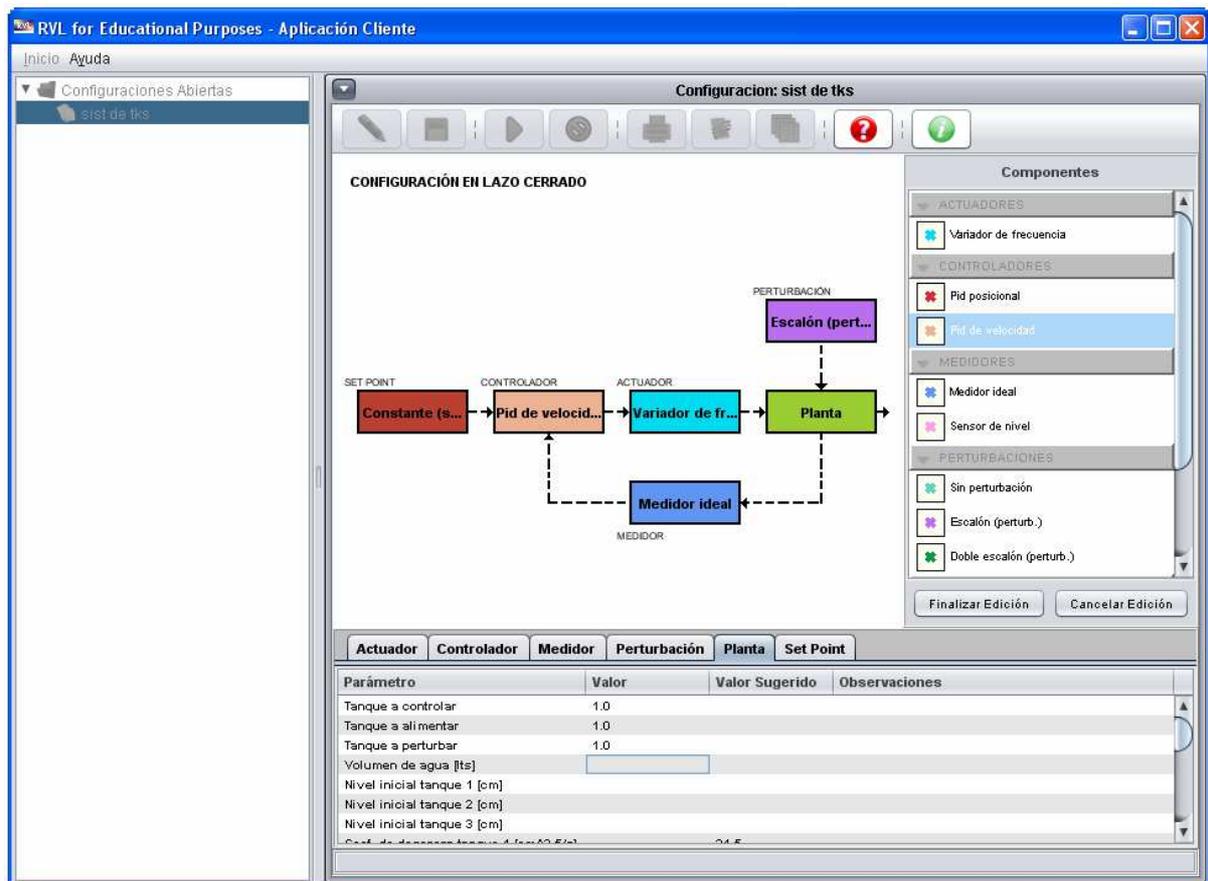


Figura 7. Configuración del LVR: Interfaz del usuario (aplicativo Cliente). El esquema de bloques indica la estructura general de un sistema de control a lazo cerrado. El usuario configura la simulación en base a la disponibilidad de componentes (lista de la derecha); y luego fija los parámetros de cada componente (solapas indicadas en la parte inferior de la interfaz).

El módulo “*Planta de Tanques*” está dedicado en el aplicativo Servidor al LVR específico del ejemplo del sistema de tanques. Este módulo es el responsable de garantizar, previo a la realización de la simulación, la validez de los valores fijados en cada uno de los parámetros. Dicho módulo interactúa con la herramienta Matlab/Simulink al momento de ejecutar la simulación. Para ello, utiliza la librería JMatLink (<http://jmatlink.sourceforge.net/>), la cual provee el conjunto de métodos necesarios para la inicialización de variables, la ejecución de la simulación, y el retorno de los resultados. De esta forma, se simplifica la comunicación entre la plataforma y la herramienta de simulación requerida. Finalizada la simulación, se empaqueta el conjunto de valores retornados en instancias de la clase

Resultado.java; y posteriormente se envía esta información al cliente para su visualización y eventual almacenamiento de los resultados.

El módulo “*Visualización de Resultados*” define el formato de presentación de los resultados obtenidos a lo largo del proceso de simulación. Permite visualizar dicha información bajo dos esquemas: tablas y gráficos. No existe un conjunto de gráficos predeterminados, sino que el usuario debe definir sus propios gráficos en términos de las variables de abscisas y ordenadas. De esta manera se flexibiliza el estudio de los resultados, puesto que el operador tiene la capacidad de definir el conjunto de gráficos a estudiar en función de las variables que le sean de interés. En forma tabulada el módulo también permite presentar el conjunto de datos asociados al gráfico.

Además, brinda la posibilidad de visualizar dos gráficos en una misma pantalla, facilitando la comparación de la evolución temporal de diferentes variables. Este módulo permite también obtener una única tabla que resume los principales resultados de la simulación.

La Fig. 8 muestra la apariencia de la interfaz gráfica de visualización de resultados, y (superpuesta parcialmente) un ejemplo de la tabla con sus datos numéricos asociados. En el caso particular del LVR del sistema de control de niveles, el usuario tiene la posibilidad de definir gráficos en base a: tiempo de simulación, nivel de cada uno de los tanques principales, nivel del tanque de reserva, corriente de los sensores de medición, frecuencia de la bomba, etc.

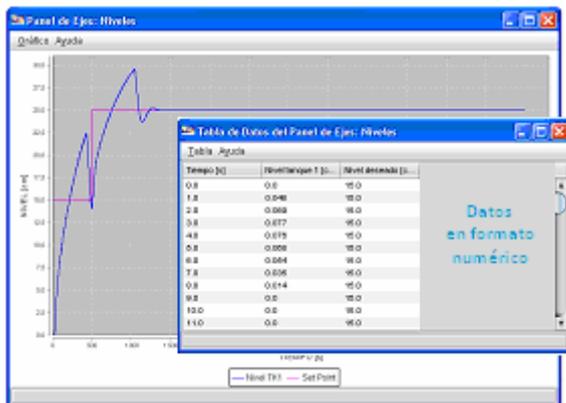


Figura 8. Visualización de resultados del LVR: Interfaz del usuario (aplicativo Cliente). Pantalla gráfica con la evolución temporal de 2 curvas. Parcialmente superpuesta se indica también la tabla con los datos numéricos de las variables asociadas.

El módulo “Carga de Configuración del LVR” es el encargado de recuperar la información asociada a los LVR disponibles. Mediante un conjunto de consultas SQL interactúa con la base de datos en 3 oportunidades, solicitando:

- Nombre y descripción de los LVR disponibles.
- Conjunto de componentes asociados a un LVR particular (valores de referencia, actuadores, medidores, controladores y perturbaciones).

- Datos, variables de entrada y variables de salida del modelo de simulación asociado a un LVR particular.

El módulo “Comunicación” se encuentra desdoblado en los dos aplicativos. Su implementación se realizó a nivel de sockets, gestionando la conexión en ambos extremos. Del lado Servidor, se espera por un mensaje en un puerto bien conocido. La recepción de este mensaje indica que un cliente ha enviado una solicitud. Esta solicitud será atendida por un nuevo hilo de ejecución, el cual existirá mientras se procese la respuesta. El lado Cliente crea una conexión con el Servidor en base a su dirección IP y al puerto en el cual éste está escuchando; y sobre esta conexión realiza el envío de las solicitudes. Con el objetivo de mejorar la interpretación de dichas solicitudes, se definió una estructura para cada uno de los mensajes intervinientes en el accionar de la plataforma, lográndose un intercambio de mensajes limpio y robusto.

Funcionalidades Adicionales

En forma complementaria a los módulos principales, se incorporó a la plataforma un conjunto de funcionalidades que sirven como soporte a la experimentación realizada. En el aplicativo Cliente se adicionó la posibilidad de almacenar configuraciones, generar reportes sobre los resultados de la simulación, exportar las gráficas diseñadas en formato PNG y guardar las tablas de resultados en formato TXT. Por otra parte, en el aplicativo Servidor se posibilitó el almacenamiento de un archivo de log en formato TXT y la generación de estadísticas referentes a tiempos de transacción y cantidad de clientes conectados.

Discusión Final

El desarrollo del LV ha sido llevado a cabo, básicamente, por 2 alumnos de la FRSF-UTN: un alumno de la carrera Ingeniería Eléctrica y otro de Ingeniería en Sistemas de Información. Por sus formaciones

respectivas, cada uno pudo desempeñarse con mayor eficiencia en la generación del modelo matemático de la planta (alumno de Ing. Eléctrica) o en el desarrollo de la plataforma informática (alumno de ISI). Sin embargo, en diversas oportunidades han trabajado en forma conjunta y coordinada, fomentándose así un esquema de trabajo interdisciplinario que fue ejecutado con un elevado grado de éxito, y que contribuye a sus respectivas formaciones profesionales.

El aplicativo Cliente (Fig. 7) muestra un esquema típico de lazo cerrado de una planta genérica. Sin embargo, debe tenerse en cuenta que la herramienta también permite simular la operación en lazo abierto. Tal operatoria puede lograrse a través de una parametrización específica, configurando un medidor ideal de ganancia nula (para eliminar la realimentación de la medición), y simultáneamente adoptando un controlador con sólo efecto proporcional (P), de ganancia unitaria. Para evitar esta configuración singular, se prevé incorporar en la interfaz una opción para que el mismo sistema Cliente fije automáticamente estas características de configuración, cuando el usuario manifieste la necesidad de simular en lazo abierto. Si bien la cantidad de señales de entrada posibles es aún limitada, también se tiene previsto incorporar nuevas funciones en un futuro.

El simulador del LVR fue desarrollado en Matlab/Simulink, dado que la FRSF-UTN cuenta con una licencia de este software. Sin embargo, también se están explorando alternativas de uso de software libre y multiplataforma. Por ejemplo, esta misma planta de control de niveles ya ha sido desarrollada en el software Python, utilizando los paquetes matemáticos NumPy y SciPy para la resolución numérica del sistema de EDOs. La interfaz gráfica se construyó utilizando el toolkit GTK+ y el binding Python correspondiente. El paquete Matplotlib se utilizó para dotar al nuevo sistema de posibilidades gráficas. Más detalles pueden consultarse en [16].

Los aplicativos desarrollados no admiten una incorporación directa como módulos

dentro de la difundida plataforma Moodle (moodle.org). La diferencia entre ambos entornos radica en las tecnologías utilizadas. En efecto, Moodle es una aplicación web, mientras que la plataforma desarrollada es una aplicación distribuida que utiliza conexiones TCP/IP con un equipo remoto. Sin embargo, es factible reformular el aplicativo Cliente para que funcione mediante una interfaz web (en lugar de una interfaz de escritorio), y brindar así un módulo apropiado para su inclusión en Moodle. De esta manera, el aplicativo Servidor actual podría utilizarse sin alteraciones, brindando servicios a un aplicativo web y a un aplicativo distribuido.

Conclusiones

La plataforma desarrollada para soporte y gestión de LVR ha sido diseñada con un sentido amplio y general, como para habilitar la incorporación de otras plantas, y contribuir, de esta manera, a engrosar el número de LVR gestionados por el sistema. Para ello, sólo será necesario trabajar sobre el aplicativo Servidor, a efectos de incorporar los módulos dedicados del nuevo sistema (ver Fig. 6). El desarrollador del simulador de una nueva planta deberá conocer los detalles que permitan su comunicación con otros módulos. Para ello, actualmente se está redactando la documentación técnica específica asociada a la plataforma, en vistas a asegurar su continuidad de uso en el futuro. Más detalles sobre los aspectos relacionados al análisis, desarrollo e implementación de la plataforma, pueden consultarse en [17].

La herramienta desarrollada puede ser explotada por usuarios (por ejemplo, alumnos), que deseen experimentar sobre el ajuste de algoritmos de control a lazo cerrado, aun sin tener un conocimiento acabado sobre la física de la planta a controlar, y sin haber intervenido en su modelado ni en el desarrollo del correspondiente simulador. Un usuario estándar del LVR, sólo requerirá disponer de la aplicación Cliente. Este aplicativo es reducido, ejecutable en una máquina

estándar, con la única condición de poseer acceso a Internet y los controladores Java.

Como herramienta didáctica, el LVR puede ser utilizado para la enseñanza universitaria en cursos de control, automatización, y sistemas dinámicos, entre otros. La posibilidad de su explotación en un aula se limita sólo por la disponibilidad de acceso a la red.

El objetivo último del presente proyecto es poder instalar el aplicativo Servidor en una máquina de la FRSF-UTN, y promover su uso en la comunidad académica. A tales efectos, deben aún sortearse aspectos legales, reglamentarios y técnicos, que aseguren un correcto uso y permanente mantenimiento del sistema en condiciones adecuadas.

Agradecimientos

A la UTN, por el soporte económico brindado a través del proyecto de investigación PID 25/O113 (ver detalles en la referencia [2]).

Referencias

- [1] Feisel, L., Rosa, A. (2005). The Role of the Laboratory in Undergraduate Engineering Education. *J. of Eng. Educ.*, 94, 121-130.
- [2] Proyecto 25/O113 (FRSF-UTN), "Laboratorios Virtuales y Reales con Acceso Local y Remoto Destinados a la Enseñanza y la Investigación en Automatización y Control". Dir. J. Vega. 2009-13.
- [3] Vary, J. (2000). "Informe de la Reunión de Expertos sobre Laboratorios Virtuales", Instituto Internacional de Física Teórica y Aplicada (IITAP), Ames, Iowa – UNESCO, París.
- [4] Guzmán, J., Vargas, H., Sánchez, J., Berenguel, M., Dormido, S., Rodríguez, F. (2007). *Education Research in Engineering Studies: Interactivity, Virtual and Remote Labs*. Distance Education Research Trends, Nova Science Publisher.
- [5] Farias, G., Dormido, S., Esquembre, F., Santos, M., Dormido-Canto, S. (2008). Laboratorio Virtual de Reconocimiento de Patrones usando Easy Java Simulations y Matlab. XXIX Jornadas de Automática, Tarragona, España.
- [6] Farias, G., Esquembre, F., Sanchez, J., Dormido, S., Vargas, H., Dormido-Canto, S., Canto, M. (2006). Desarrollo de Laboratorios Virtuales, Interactivos y Remotos utilizando Easy Java Simulations y Modelos Simulink. 12 Congreso Latinoamericano de Control Automático, Salvador de Bahía, Brasil.
- [7] Farias, G., Esquembre, F., Sanchez, J., Dormido, S., Vargas, H., Dormido-Canto, S., Duro, N. (2006). Laboratorios Virtuales Remotos Usando Easy Java Simulations y Simulink. XXVII Jornadas de Automática, Almería, 926-933.
- [8] Campo, D., Gutiérrez, M., Vega, J., Roa, J., Stegmayer, G., Loyarte, A. (2011). "Operación de una Planta de Nivel. Evaluación de una Estrategia de Control Basada en Agentes Inteligentes". 40 Jornadas Argentinas de Informática (JAIIO), 12 Argentine Symposium on Technology (AST), pp. 180-192. Córdoba, 29 de agosto al 02 de septiembre de 2011. ISSN: 1850-2806.
- [9] Loyarte, A., Díaz, G., Rosa, J. (2010). Sistema de Control de Niveles con Tanques de Sección No Uniforme. Modelado Matemático y Ajuste de Parámetros. XXII AADECA, 31 de agosto de 2010, Buenos Aires. Trabajo AF-1783.
- [10] Loyarte, A., Blas, M. (2012). Laboratorio Virtual Remoto: Una Herramienta para La Enseñanza de Control Automático. XXIII AADECA, 03-05 de oct. de 2012, Buenos Aires.
- [11] Loyarte, A. (2011). Sistema de Control de Niveles con Cuatro Tanques Interconectados: Modelado Matemático y Estimación de Parámetros. XIV RPIC, Oro Verde, 16-18 de nov. de 2011. Trabajo: 167, pp. 965-970.
- [12] Shaw, M., Garlan, D. (1996). *Software Architecture, Perspectives on an Emerging Discipline*. Prentice-Hall.
- [13] Bass, L., Clements, P., Kazman, R. (1998). *Software Architecture in Practice*. Addison-Wesley.
- [14] Boullón Garzón, R., Rodríguez Laredo, P. (2006). Comparativas SQL. Proyecto Final de Carrera, Ing. Téc. en Telecomunicaciones, Esp. en Sistemas Electrónicos, Universidad Politécnica de Catalunya, España.
- [15] Rosenthal, A., Martínez, R., Morillo, D., Miranda, D. (2010). PostgreSQL y SQLServer: Una Comparativa de Rendimiento. Ministerio del Poder Popular para la Educación Universitaria, Colegio Universitario de Los Teques Cecilio Acosta, Venezuela, Vol. 15, pp. 13.
- [16] Loyarte, A. (2013). "Plataforma de simulación de un sistema real de control de niveles basada en software libre". Trabajo enviado a CoNaIISI 2013.
- [17] Blas, M. (2013). "Implementación de una Herramienta Cliente/Servidor para la Experimentación con Laboratorios Virtuales y Remotos". Trabajo enviado a CoNaIISI 2013.

Datos de Contacto

Prof. Jorge Ruben Vega. Facultad Regional Santa Fe (FRSF), Universidad Tecnológica Nacional (UTN). Lavaisse 610 (3000) Santa Fe (Argentina). E-mail: jvega@santafe-conicet.gov.ar.