

Operación de una Planta de Niveles. Evaluación de una Estrategia de Control basada en Agentes Inteligentes

Campo, David; Gutiérrez, María; Vega, Jorge; Roa, Jorge; Stegmayer, Georgina
Loyarte, Ariel;

Universidad Tecnológica Nacional (UTN), Facultad Regional Santa Fe (FRSF), Lavaise 610
– Santa Fe – Argentina

{mmgutier; jroa}@frsf.utn.edu.ar; {jvega; gstegmayer}@santafe-conicet.gov.ar;
dncampo@gmail.com; ariel.loyarte@gmail.com

Abstract. En carreras de grado, las asignaturas de control usualmente utilizan como caso de estudio el modelado y la operación de sistemas de control de niveles, en base al diseño y sintonía de controladores lineales clásicos. En algunas plantas no lineales, tales controladores pueden exhibir un desempeño pobre y conducir a errores sistemáticos o a operatorias indeseadas de los accionamientos. Este trabajo propone utilizar el concepto de agente inteligente para definir una estrategia de control de una planta no lineal, compuesta por dos tanques interconectados (uno de ellos, con sección transversal no uniforme con la altura). Para reducir la brecha entre el desarrollo teórico del agente y su implementación práctica, se utiliza una herramienta que permite el diseño desde una perspectiva conceptual, y la generación automática del código del agente de software correspondiente. Se muestra su utilización con un ejemplo de un agente lógico que simula la manipulación del caudal de alimentación. La planta real está disponible en el Laboratorio de Sistemas de Control, de la FRSF-UTN.

Palabras claves: Agente, sistemas de control, planta no lineal, controladores.

1. Introducción

En los cursos para la enseñanza de control, quizá uno de los ejemplos más típicos se basa en los sistemas de control de nivel de líquidos en tanques; y a pesar de su amplia divulgación, estos sistemas continúan todavía siendo vigentes. En los últimos años, se ha trabajado en el desarrollo de laboratorios (reales o virtuales) con acceso local o remoto, pensados como herramientas para dar soporte a cursos de grado en el área de automatización y control. Por ejemplo, García y col. (2005) presentaron un laboratorio remoto para estudiar un sistema multivariable no lineal de 4 tanques. Más recientemente, Dormido y col. (2008) publicaron un trabajo sobre un laboratorio de control basado en Internet, para aplicaciones de control no lineal multivariable, sobre la base de un sistema dinámico compuesto por tres tanques interconectados. Como interfaz para el soporte de la programación se utilizó una herramienta basada en Java;

y la integración fue implementada en un ambiente de experimentación colaborativo basado en Web (Gillet y col., 2005).

En las plantas de control de nivel usadas en los cursos, generalmente se utilizan sistemas de tanques de sección transversal uniforme, interconectados, con entradas de líquido independientes, y con salidas de líquido (desde ambos tanques) limitadas por sendas restricciones hidráulicas o válvulas. El procedimiento típico de estudio consiste inicialmente en desarrollar un modelo matemático del sistema, estimar sus parámetros en base a experimentos (reales o supuestos), e implementar el modelo en un simulador para estudiar la evolución dinámica de las principales variables del sistema, ante distintos tipos de entradas típicas. En una segunda etapa, se encara el desarrollo de un sistema de control a lazo cerrado, con medición de alguno de los niveles, y selección / ajuste de un controlador clásico (típicamente, con efectos proporcional e integral: P+I). El desempeño del controlador P+I es muy bueno en sistemas de tanques uniformes. Sin embargo, tal desempeño puede verse afectado cuando los tanques tienen secciones transversales irregulares.

Loyarte y col. (2010) desarrollaron un modelo matemático no lineal para el sistema de control de niveles que se detalla en la Figura 1, y que corresponde a una planta piloto existente en el Laboratorio de Sistemas de Control (Departamento de Ingeniería Eléctrica, FRSF-UTN). La complejidad del sistema reside en la sección transversal no uniforme del tanque 2. Los parámetros del modelo desarrollado (C_1 , C_2 , C_{int}) fueron ajustados en base a experimentos de estado estacionario; y posteriormente validados con experimentos dinámicos (Loyarte y col., 2010). Desde el punto de vista del control de niveles en este sistema, la mayor complejidad ocurre en la zona cercana al cambio de sección brusca observada en el tanque 2 (TK2) (a los 40 cm medidos desde el fondo de ese tanque). Al utilizarse un controlador clásico, es muy probable que el desempeño del sistema dinámico se vea deteriorado en la zona próxima a los 40 cm, en el sentido de producirse sobre-exigencias a los accionamientos de la planta (bomba impulsora del agua), así como también en cuanto a la reproducción de la consigna de control previamente seleccionada.

En este trabajo, se propone analizar la operación de la planta en lazo cerrado, utilizando inicialmente la estrategia de control más simple: un controlador proporcional (P). En una segunda instancia, se pretende diseñar y evaluar un esquema de control alternativo y más específico al problema, utilizando un agente de software (Russell y Norvig, 2003) que desarrolle las tareas típicas de un controlador. Para implementar el agente de software se utilizó una herramienta para el diseño de agentes inteligentes: IDEM-IA (Santana, 2010). A los efectos de la toma de decisiones por parte del agente se utilizaron reglas lógicas. El trabajo se organiza de la siguiente manera. En la sección siguiente se presenta el problema estudiado y la respuesta obtenida con el uso de un controlador P. Luego el agente de software propuesto se describe usando IDEM-IA y se muestran las soluciones obtenidas. Finalmente se presentan las conclusiones del trabajo.

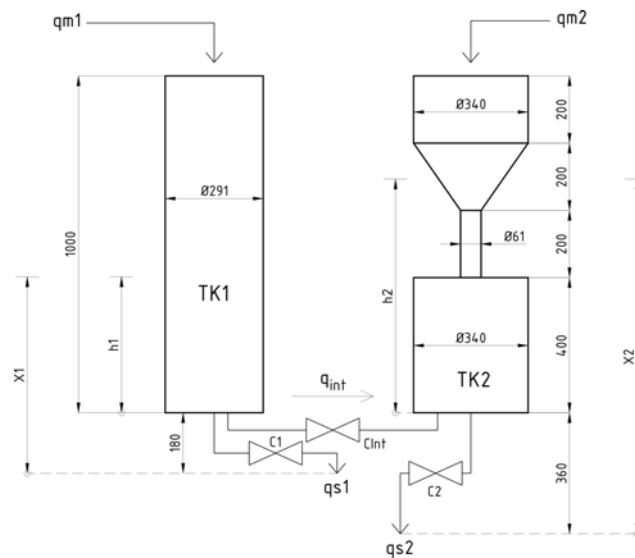


Fig. 1. Sistemas de tanques interconectados (dimensiones reales indicadas en milímetros)

2. Ejemplo Propuesto

Se propone estudiar por simulación dinámica el control a lazo cerrado del sistema de tanques interconectados de la Figura 1. Se pretende controlar el nivel de agua en el TK2 (h_2), manipulando sólo el caudal de ingreso de agua al TK1 (q_{m1}). Se admitirá que la medición del nivel h_2 es ideal, en el sentido que el medidor hipotético de nivel no tendrá error, retardo, ni ruido. El esquema de control en bloques, se describe en la Figura 2.

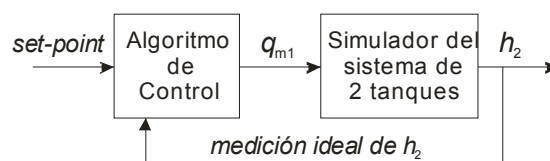


Fig. 2. Esquema de bloque del sistema de control

El simulador del sistema de dos tanques consiste en una aplicación ya disponible desarrollada en el ambiente de simulación Matlab (MathWorks, Inc.). Esta aplicación está parametrizada de manera tal que puede ser ejecutada con diferentes valores de caudales de entrada, en pasos discretos de tiempo de período T [seg]. Durante cada período T , el simulador observa una entrada constante de magnitud q_{m1} , y efectúa la simulación del sistema de tanques de manera continua. Al cabo de T seg, el simulador retorna la información de los niveles de cada tanque (h_1 y h_2). El valor de h_2 , después

de T segundos, es utilizado por el algoritmo de control para actualizar el caudal de alimentación a utilizarse en el periodo siguiente, en función del valor de consigna seleccionado o *set-point*.

2.1 Algoritmo de control clásico

Como aplicación básica del sistema de control, se seleccionó un controlador clásico proporcional (P), de ganancia constante de valor K_p . La ley de control, se calcula de la siguiente manera:

$$q_{m1}[(k+1) T] = Q_{REF} + K_p [h_2(k T) - h_2^d(k T)] \quad (1)$$

donde el índice ' k ' ($k = 0, 1, \dots$) indica la secuencia temporal discreta; y h_2^d representa la entrada de consigna o *set-point*. El valor constante Q_{REF} representa un caudal de referencia fijo.

La planta de tanques considerada es de orden 2 (dado que incluye 2 tanques interconectados de orden 1). El control a lazo cerrado con controlador proporcional y medidor ideal no altera el orden original del sistema, y por tal motivo el lazo cerrado será también de orden 2. En estas condiciones, el sistema de lazo cerrado tendrá necesariamente error de estado estacionario (*off-set*), por lo que no podrá alcanzar el nivel h_2 final prefijado en la consigna.

Como ejemplo de aplicación, se selecciona una consigna tipo rampa saturada, definida como:

$$h_2^d(k T) = \begin{cases} h_2^d(0) + \frac{50 - h_2^d(0)}{1500}(k T); & \text{si } k T < 1500 \\ h_{2,\text{final}} & ; \text{ si } k T \geq 1500 \end{cases} \quad (2)$$

Es decir, un incremento lineal de h_2 , desde $h_2(0)$ hasta $h_{2,\text{final}}$, alcanzados en un tiempo de 1500 seg; y luego se mantiene el valor de $h_{2,\text{final}}$ hasta el tiempo final de simulación. Los tiempos característicos de la dinámica de la planta son del orden de los 2000 seg. A efectos de la aplicación se seleccionó un período de actualización de las variables de entrada / salida de $T = 5$ seg.

En la Tabla 1 se fijan los principales parámetros de simulación del sistema. Dado que el sistema de tanques bajo estudio presenta una discontinuidad a los 40 cm del TK2, se realizaron dos simulaciones donde se controla al sistema usando el algoritmo de control clásico, tomando como valores de $h_{2,\text{final}} = 50$ cm y 70 cm, respectivamente. La primera altura tomada está en la zona conflictiva mientras que la segunda altura supera esta zona. Estas pruebas permiten analizar cómo se comporta el algoritmo clásico en sistemas que presentan irregularidades.

En las Figuras 3(a, b), se muestra el funcionamiento del algoritmo clásico para un valor de consigna final $h_{2,\text{final}} = 50$ cm, cuando la constante del controlador se fija en $K_p = 40$. Se observa que el sistema de lazo cerrado presenta las siguientes deficiencias: (a) no es posible alcanzar el valor final de referencia de 50 cm; (b) la

evolución dinámica de h_2 permanece siempre por debajo del nivel deseado (h_2^d); y (c) la bomba tiene un accionamiento brusco (rápida disminución del caudal) alrededor de los 1300 seg., cuando el nivel h_2 ingresa en el sector más angosto del TK2 (zona de 40 cm de nivel, indicado por una horizontal en la figura).

En las Figuras 4(a) y 4(b), se representa el funcionamiento del lazo cerrado para $h_{2,final} = 70$ cm. Apreciaciones similares a las anteriores vuelven a repetirse en este caso, pero la reducción brusca se presenta alrededor de los 850 seg.

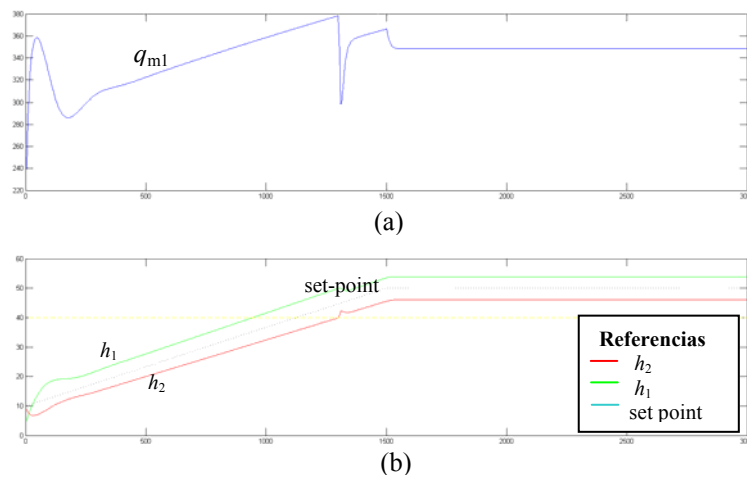


Fig. 3. Control de la planta usando control clásico para *set-point* 50cm

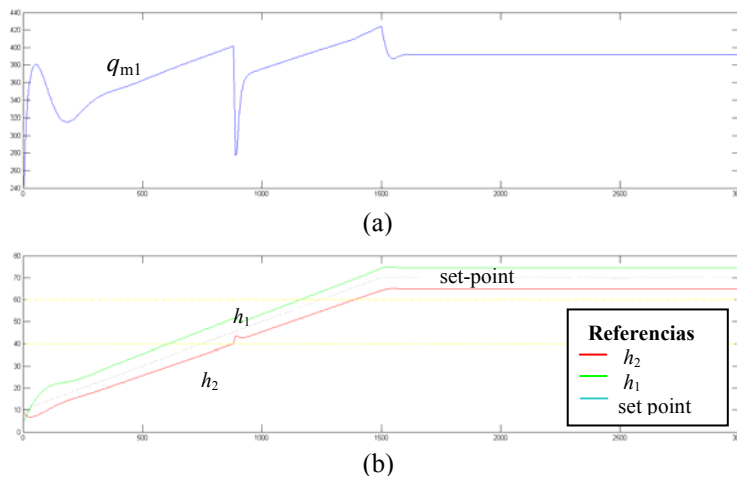


Fig. 4. Control de la planta usando control clásico para *set-point* 70cm

A continuación, se plantea un algoritmo de control inteligente como una alternativa válida para este tipo de sistemas.

3. Desarrollo del Agente

Para el diseño del agente que controla la planta descrita en el apartado anterior, se definió un agente reactivo utilizando la herramienta IDEM-IA. El código generado por IDEM-IA resuelve la conexión de Java con Matlab, proveyendo las librerías y el código necesario. Gracias a estas características del código auto-generado fue posible desarrollar el agente que interactúa con un ambiente (o “planta”) desarrollado en Matlab con mínimo esfuerzo de programación y concentrando la atención en la solución del problema de control y diseño del agente. Estas características de IDEM-IA son importantes para relacionar a través de la solución de un problema práctico, conceptos incluidos en asignaturas de grado de la FRSE, tales como “Teoría de Control” e “Inteligencia Artificial”. El código generado por la herramienta utiliza como base el Framework FAIA¹ (Roa y col., 2010; Gutierrez y col., 2010) el cual presenta un diseño orientado a objetos de un agente de software.

Desde el punto de vista de Inteligencia Artificial, es necesario definir el agente y el ambiente con el cual interactúa. La definición del agente implica identificar metas, acciones, percepciones, estado y estrategia de toma de decisión. La definición del ambiente implica definir su estado y su simulador.

La Figura 5 muestra la definición del agente llamado *WaterTankAgent* y de su ambiente llamado *WaterTankEnvironment*.

En cuanto al agente, se definió una acción *ChangeQm1Flow* que cambia el caudal de entrada al tanque 1 mediante la manipulación del caudal impulsado por la bomba.

La percepción definida es la altura del tanque 2, la cual se encuentra en la figura como *WaterTankPerception*.

El estado del agente está formado por: la altura del TK2 (h_2), el caudal de entrada a TK1 (q_{m1}), La altura de TK2 donde comienza la sección estrecha (H21), la altura del TK2 donde termina la sección estrecha (H22), y la función bc que describe la relación entre el *set-point* y el caudal de entrada en estado estacionario para dicho *set-point* y la constante multiplicativa. Todos estos valores son del tipo *double*.

La meta del agente es el *set-point* a lograr, el cual puede ser un valor o bien una función. La meta en la figura aparece representada con la variable h_2^d .

Para la toma de decisión, se adoptó un agente reactivo que utilice reglas lógicas que le permitan cambiar el caudal de entrada de manera de obtener la mejor performance.

¹ <http://code.google.com/p/faia/>

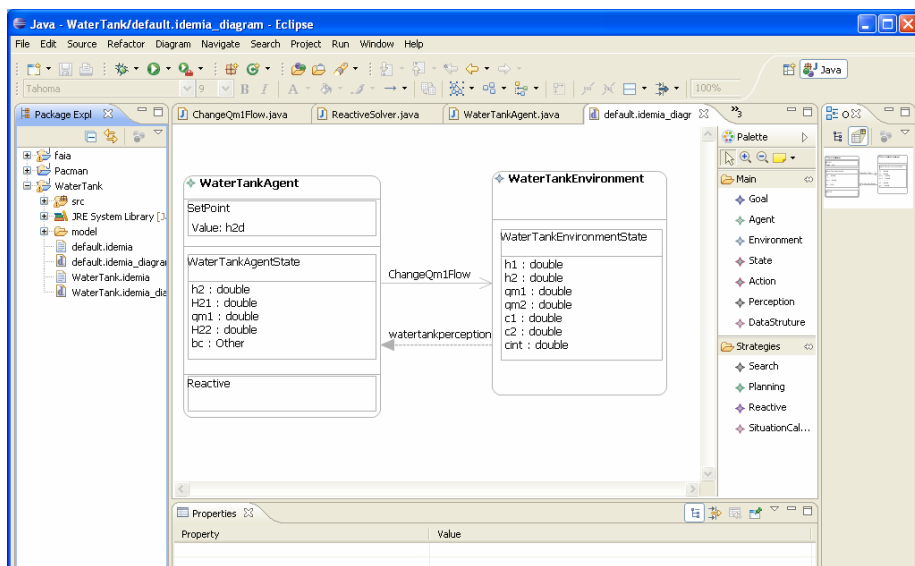


Fig. 5. Descripción conceptual del Agente usando IDEM-IA

Con respecto al ambiente se definió como estado los parámetros que definen al sistema de tanques; es decir: q_{m1} , q_{m2} para los caudales de entrada a los tanques 1 y 2, respectivamente; y h_1 y h_2 para identificar las correspondientes alturas. Por otra parte, C_1 y C_2 son constantes de resistencia hidráulica del tanque 1 y 2, y C_{int} es la constante de resistencia de interacción entre ambos tanques.

A partir de esta definición conceptual del agente y del ambiente, es posible generar en forma automática el código, obteniendo como resultado un conjunto de clases las cuales deben ser modificadas por el diseñador. El proyecto Java generado puede verse en la Figura 6. Sobre el lado izquierdo se observan las clases que fueron generadas a partir del diagrama. Luego, es necesario agregar el código correspondiente a la problemática tratada. Por ejemplo, la clase que se encuentra resaltada *ReactiveSolver* aparece sobre el lado derecho mostrando que el método *solve* debe ser completado (leyenda *TODO: complete method*). En este método es donde se escribirá el código correspondiente a la estrategia que usa el agente para modificar el caudal de entrada.

También se puede ver en la Figura 7 que la clase *WaterTankAgent* está completamente definida y ningún método debe ser completado.

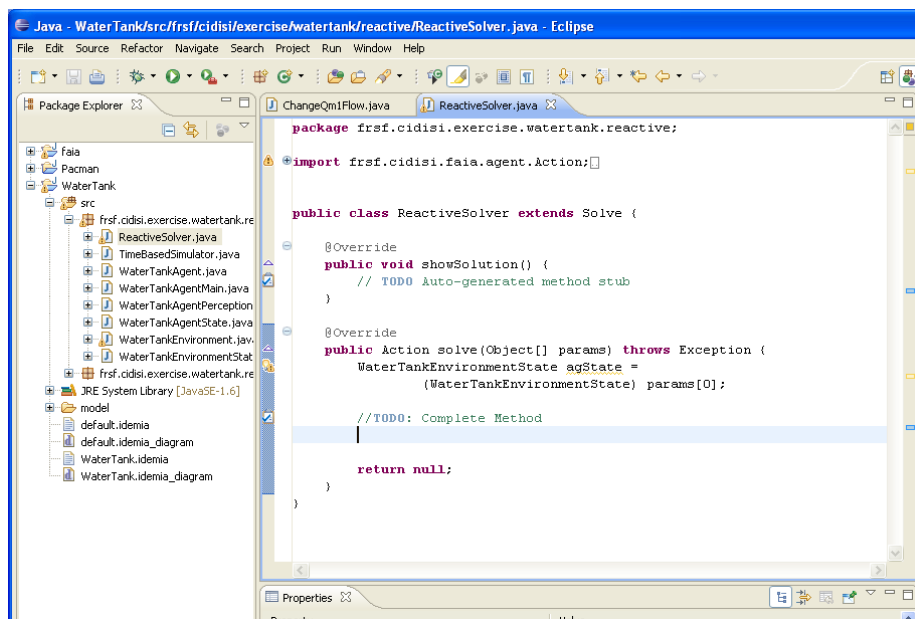


Fig. 6. Código generado a partir del diagrama IDEM-IA

3.1 Estrategia

Se utilizan reglas *if-then* para la definición de la estrategia. Estas reglas permiten seleccionar la mejor acción basado en el objetivo del agente (*set point*), el error calculado entre el objetivo y el valor percibido y la base de conocimiento.

Para codificar esta estrategia se tuvo en cuenta la información de la arquitectura de la planta a controlar. Por ejemplo, las variables de estado H_{21} y H_{22} definen las alturas límites del TK2 donde comienza y termina la sección transversal no uniforme. De esta forma, es posible en principio plantear una estrategia adecuada para este tipo de sistemas, dado que el agente posee conocimiento *a priori* que utilizará para lograr un desempeño mejorado.


```

public class WaterTankAgent extends ReactiveBasedAgent {
    public WaterTankAgent() {
        this.setAgentState(new WaterTankAgentState());
    }

    @Override
    public void see(Perception p) {
        this.getAgentState().updateState(p);
    }

    @Override
    public WaterTankAgentState getAgentState() {
        return (WaterTankAgentState) super.getAgentState();
    }

    @Override
    public Action selectAction() {
        ReactiveSolver solver = new ReactiveSolver();

        this.setSolver(solver);

        // It gets the best action from the solver
        Action selectedAction = null;
        try {
            selectedAction =
                this.getSolver().solve(new Object[] { this.getAgentState() });
        } catch (Exception ex) {
            Logger.getLogger(WaterTankAgent.class.getName()).log(Level.SEVERE,
                null, ex);
        }
    }
}

```

Fig. 7. Código de la clase generada *WaterTankAgent*

También se dotó al agente de conocimiento relacionado al caudal de entrada en estado estacionario para los distintos niveles de h_2 . Esta información es utilizada para decidir cuál es el caudal apropiado en cada iteración. Para ello, el agente fue entrenado utilizando simulación de la planta a lazo abierto, obteniéndose para distintos niveles de h_2 caudales que permiten mantener el sistema en estado estacionario. Estos datos se materializaron en la función *bc(objetivo)*. Esta función retorna un par de la forma *(constante, caudal)* que identifican respectivamente la constante por la cual se multiplica el error y el caudal que mantiene al sistema en estado estacionario para el argumento *objetivo*.

Para evitar el salto brusco del caudal de entrada que se genera con el controlador P clásico, en la estrategia planteada del agente, se estableció que los cambios de caudal no pueden superar el 10% del valor anterior. Se conoce que estos cambios se generan en las zonas cercanas a la sección transversal no uniforme del TK2 identificados con H_{21} y H_{22} . Por tal motivo las reglas que se definen tienen en cuenta estos valores.

Concretamente, para decidir el valor del caudal en la próxima iteración se utilizaron 3 tipos de reglas que se corresponden con cada una de las secciones del TK2. Es decir, se utiliza una regla cuando el objetivo está por debajo de H_{21} , otra cuando el objetivo está entre H_{21} y H_{22} , y una tercera para objetivos mayores a H_{22} .

Para el cálculo del caudal, en las reglas se utiliza la siguiente expresión:

$$q_{m1} = A + B * \text{set point} + C (h_2 - \text{set point})^2 \quad (3)$$

donde A, B y C son constantes que se obtienen de la base de conocimiento; y la diferencia $(h_2 - \text{set point})$ representa el error de reproducción de la consigna.

La Figura 8 muestra el código de toma de decisión implementado por el agente en la clase *solve*, la cual representa la estrategia de toma de decisión.

```
private Action controlInteligente(WaterTankAgentState
tankState) {

    double meta = tankState.getH2d();

    double error = meta -tankState.getH2();

    double constant;

    double flow = 0.0;

    double sign = error / Math.abs(error);

    double mul = 10;

    // regla 1: set-point por debajo de 40cm3
    if (meta < tankState.H21) {

        constant = tankState.bc(meta).c() - (meta *0.01755);

        flow = tankState.bc(meta).q()+ constant*(meta -1.0)
+ sign*Math.pow(mul*error, 2.0);}

    // regla 2: set point entre 40cm3 y 60cm3
    if ((meta >= tankState.H21) && (meta <= tankState.H22))
    {

        constant = tankState.bc(meta).c();

        flow = tankState.bc(meta).q() + constant*(meta -
40.0) + sign*Math.pow(mul/2*error, 2.0);}

    // regla 3: set-point mayor a 60cm3
    if (meta >= tankState.H22) {

        constant = tankState.bc(meta).c();

        flow = tankState.bc(meta).q()+ (0.1)*(meta -60.0) +
constant*(meta - 60.0) + sign*Math.pow(mul/2*error,
2.0);}

    Action ac = new ChangeQm1Flow(normalize flow(tankState,
```

```
flow, error));  
  
return ac; }
```

Fig. 8. Algoritmo de control del agente.

La función *normalize_flow(state, error)* controla que el flujo no supere, por defecto o por exceso, el 10 % del valor anterior, si esto ocurre, aumenta o disminuye solo en 10%. También controla no superar las cotas establecidas para la bomba.

3.2 Resultados de Simulación

Para poder comparar el control inteligente implementado en el agente presentado anteriormente, se simularon los mismos ejemplos del control clásico. Los resultados de estos experimentos se muestran en las Figuras 9 y 10. Con la estrategia de control diseñada, no se observan cambios bruscos en el caudal de alimentación, sino que sólo resultaron aumentos o disminuciones graduales. Este comportamiento se debe a que cuando existe una disminución/aumento del caudal que supera el 10% del valor utilizado en la iteración anterior, el agente reajusta el caudal independientemente del valor que haya resultado de la aplicación de las correspondientes expresiones de cálculo. Con respecto a los niveles de los tanques h_1 y h_2 , se observa que con el control inteligente, las alturas se mantienen cercanas a los valores de *set-point* durante todo el intervalo de simulación, alcanzándose en el período estacionario, el valor de h_{2final} con un error prácticamente nulo.

4. Conclusiones

Se estudió por simulación numérica una planta de control de niveles compuesta por dos tanques interconectados. La planta original es no lineal, y como consecuencia su operación en lazo cerrado utilizando un controlador clásico tipo proporcional, presenta algunas desventajas esperables, tales como: i) la falta de capacidad para reproducir trayectorias de consigna, y ii) la exigencia inadecuada sobre algunos dispositivos de accionamiento (la bomba de agua, en este ejemplo).

Para mitigar las deficiencias observadas, se propuso una estrategia de control basada en agentes de software, la cual hace uso de información sobre la geometría de la planta para optimizar el desempeño dinámico de lazo cerrado. Los resultados obtenidos en esta instancia son sólo indicativos de la potencialidad del método; pero no fueron optimizados para lograr un mejor desempeño. Los desarrollos efectuados y los resultados obtenidos pueden ser utilizados como ejemplos conceptuales para el dictado de asignaturas tales como “Teoría de Control” e “Inteligencia Artificial”, en asignaturas de grado de la FRSF-UTN.

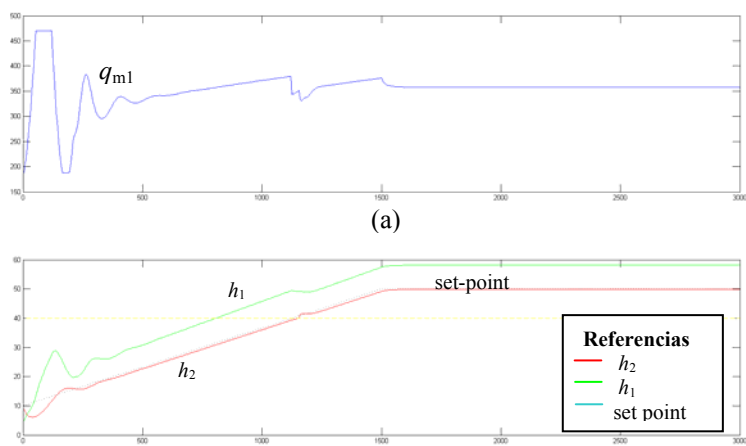


Fig. 9. Control de la planta usando Agente para un *set-point* 50cm

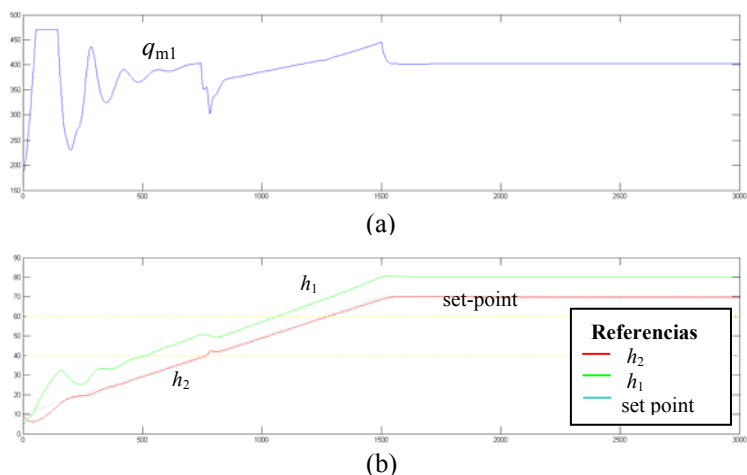


Fig. 10. Control de la planta usando Agente para *set-point* 70 cm

Referencias

- Dormido, R; Vargas, H; Duro, N; Sánchez, J; Dormido-Canto, S; Farías, G; Esquembre, F; Dormido, S (2008). "Development of a Web-Based Control Laboratory for Automation Technicians: The Three-Tank System". IEEE Trans. on Educ., 51, 1, 44.
- García, AA; Berenguel, JL; Dormido, S; Domínguez, M (2005). "Laboratorio Remoto para el Proceso de los Cuatro Tanques y su Aplicación en Docencia de Control Multivariable". EIWISA 2005 – IV Jornadas de Enseñanza a través de Internet/Web de la Ingeniería de Sistemas y Automática (CEA-IFAC), pp. 3-8. ISBN: 84-9732-451-X. Congreso Español de Informática, Granada (España), 14-16 sept.

Operación de una Planta de Niveles. Evaluación de una Estrategia de Control basada en Agentes Inteligentes 13

- Gillet, D; Nguyen, V; Rekik, Y (2005). "Collaborative Web-based experimentation in Flexible Engineering Education", IEEE Trans Educ., 48, 4, 696-704.
- Loyarte, A., Diaz, G., Rosa, J. (2010). Sistema de control de niveles con tanques de sección no uniforme. Modelado matemático y ajuste de parámetros. XXII Congreso Argentino de Control Automático – AADECA 2010, Buenos Aires, 31 de agosto al 02 de septiembre de 2010.
- Roa, J., Pividori, M., Gutiérrez, M., Stegmayer, G. (2010). Chapter 6: How to develop intelligent agents in an easy way with FAIA. Quality and Communicability for Interactive Hypermedia Systems: Concepts and Practices for Design. Editorial IGI global. Editores Francisco V. Cipolla Ficara , AINCI (International Association of Communication Interactive) and ALAIPO (Latin Association of HCI). ISBN: 978-1-61520-763-3. Pag: 120-140.
- Gutiérrez, M., Roa, J., Pividori, M., Stegmayer, G., Vega, J. (2010). Capítulo: FAIA: un Framework para el desarrollo de agentes inteligentes. La tecnología educativa al servicio de la educación tecnológica. Ed. eduTecNe. Editores Uriel Cukierman y José M. Virgili. ISBN: 978-987-25855-9-4. Pag. 459-485.
- Russell, S. and Norvig, P. (2010). Artificial Intelligence: A modern Approach. Third edition. Prentice Hall.
- Santana, W. (2010). IDEM-IA: un ambiente de desarrollo integrado para el modelado e implementación de Agentes de software. Jornada de Jóvenes Investigadores. Santa Fe-UNL.